

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

**Escuela Técnica Superior en Ingeniería Agronómica y del
Medio Natural**



Escuela Técnica Superior
de Ingeniería Agronómica y del Medio Natural

**SISTEMA DE DETECCIÓN Y ALERTA DE
INUNDACIONES MEDIANTE HARDWARE Y
SOFTWARE CON LICENCIA DE CÓDIGO
ABIERTO**

**TRABAJO FINAL DE GRADO EN INGENIERÍA AGROALIMENTARIA
Y DEL MEDIO NATURAL, CURSO ACADÉMICO 2019/2020**

ALUMNO: *José Moscardó Sáez*

TUTOR: *Dr. Pablo González Altozano*

COTUTORA: *Dra. María Gasque Albalade*

Valencia, 3 de julio de 2020

TÍTULO: SISTEMA DE DETECCIÓN Y ALERTA DE INUNDACIONES MEDIANTE HARDWARE Y SOFTWARE CON LICENCIA DE CÓDIGO ABIERTO

RESUMEN

Las inundaciones son el fenómeno natural con mayor impacto en la población mundial. Los efectos desfavorables de las crecidas conllevan importantes consecuencias socioeconómicas y medioambientales que pueden prolongarse durante años. Las previsiones a medio largo plazo no mejoran este escenario, pues la tendencia es que la población residente en las zonas inundables aumente, y los procesos de cambio climático acentúen la frecuencia e intensidad de las crecidas de agua.

El análisis de los datos meteorológicos es una herramienta fundamental en la previsión de las crecidas, pero resulta insuficiente para actuar como una red de alerta de crecidas. Esta medida debe ir acompañada del seguimiento de los caudales en las zonas con riesgo de inundación.

En el presente trabajo se ha desarrollado, mediante hardware y software con licencia de código abierto, un sistema de monitoreo, detección y alerta de inundaciones aplicable a torrentes, ramblas y ríos, que mediante redes inalámbricas y plataformas IoT, facilita la difusión automatizada de avisos de alerta en páginas web, mensajes SMS, aplicaciones para teléfonos móviles y mensajes en redes sociales.

Palabra clave: Arduino, código abierto, prevención de inundaciones, vigilancia, alarma.

- **Autor del TFG:** José Moscardó Sáez.
- **Tutor:** Dr. Pablo González Altozano.
- **Cotutor:** Dra. María Gasque Albalate.
- **Localidad y fecha:** Valencia, 3 de julio de 2020.

TITLE: FLOOD DETECTION AND WARNING SYSTEM USING OPEN SOURCE HARDWARE AND SOFTWARE

ABSTRACT

Flooding is the natural phenomenon with the greatest impact on the world's population. The adverse effects of floods have significant socio-economic and environmental consequences that can last for years. The impact can be even worst considering that the population living in flood areas will continue to increase in the coming decades. Furthermore, the processes of climate change will accentuate the frequency and intensity of water floods.

The analysis of weather data is a necessary tool in flood forecasting, but it is insufficient to act as a flood alert network. It must be accompanied by monitoring of water flows in areas at risk of flooding.

In this work, using open source licensed hardware and software, a flood monitoring and alert system has been developed. The device is applicable to torrents, ephemeral streams and rivers. Through the use of wireless networks and IoT platforms, it facilitates the automated publication of alert notices on web page, social networks, SMS messages and mobile applications.

Key words: Arduino, open source, flood prevention, surveillance, warning.

- **Author:** José Moscardó Sáez.
- **Supervisor:** Dr. Pablo González Altozano.
- **Co-supervisor:** Dra. María Gasque Albalate.
- **Date and location:** Valencia, 3 July 2020.

ÍNDICE

ACRÓNIMOS	9
1. INTRODUCCIÓN	10
1.1. CRECIDAS E INUNDACIONES.....	10
1.1.1. Definiciones	10
1.1.2. Tipología de las crecidas	11
1.2. EFECTOS ECONÓMICOS Y SOCIALES DE LAS INUNDACIONES	12
1.2.1. Contexto mundial	12
1.2.2. Las inundaciones en España	15
1.1.5. Perspectivas ante el cambio climático.....	16
1.2. GESTIÓN DEL RIESGO DE INUNDACIONES.....	17
1.2.1. La gestión del riesgo de inundaciones en España.....	17
1.2.2. Previsión de crecidas	18
1.2.2.1. Sistema Automático de Información Hidrológica	19
1.2.3. Comunicación del riesgo.....	22
1.3. HARDWARE Y SOFTWARE DE CÓDIGO ABIERTO EN EL CONTEXTO RURAL	23
1.3.1. Tecnología DiDIY en el contexto rural.....	24
1.3.2. Arduino	26
1.3.2.1. Arduino Software	28
1.3.3. Sensores y módulos de código abierto.....	29
1.4. DETECCIÓN DEL NIVEL Y CAUDAL DE AGUA MEDIANTE ULTRASONIDOS.....	30
1.4.1. Dispositivos de medición del nivel de agua basados en el hardware de código abierto.....	31
2. OBJETIVOS	32
3. MATERIAL, METODOLOGÍA Y PUESTA A PUNTO	33
3.1. DESCRIPCIÓN DEL DISPOSITIVO	33
3.1.1. Hardware	35
3.1.1.1. Microcontrolador Arduino UNO Rev3.....	35
3.1.1.2. Sensor de ultrasonidos JSN R04Tv2	38
3.1.1.3. Módulo de comunicación GSM/GPRS SIM900.....	41
3.1.1.4. Pantalla LCD 20x4 I2c.....	43
3.1.1.5. Alimentación.....	45
3.1.1.6. Montaje del prototipo	46

3.1.1.7. Alojamiento del dispositivo.....	48
3.1.2. Software.....	49
3.1.2.1. Programación del prototipo en Arduino IDE.....	49
3.1.2.2. ThinkSpeak.....	59
3.1.2.3. Aplicación móvil con MIT App Inventor	60
3.1.2.4. Publicación en página Web.....	62
3.1.2.5. Publicación Redes Sociales.....	63
3.1.5. Coste material del prototipo.....	64
3.2. ENSAYOS DE FUNCIONAMIENTO Y PRECISIÓN	65
3.2.1. Ensayo previo de precisión	65
3.2.2. Comportamiento del dispositivo en mediciones de altura de lámina de agua	68
3.3. SIMULACIÓN DE MONITOREO Y ALERTA DE INUNDACIÓN EN UN TORRENTE	69
3.3.1. Descripción de la simulación realizada	69
3.2.2. Ubicación del dispositivo	70
3.2.3. Determinación de la distancia del sensor al lecho del cauce e incorporación de datos al programa	73
3.2.4. Determinación del nivel de alerta e incorporación de datos al programa	74
3.2.5. Simulación de crecida del torrente con parámetros controlados	75
4. RESULTADOS Y COMPROBACIÓN DEL FUNCIONAMIENTO	77
4.1. PRECISIÓN DEL DISPOSTIVO	77
4.2. ENVÍO DE DATOS A LA PLATAFORMA THINGSPEAK	78
4.3. PUBLICACIÓN AUTOMÁTICA EN PÁGINA WEB	81
4.4. PUBLICACIÓN AUTOMÁTICA EN APP PARA ANDROID	82
4.5. EMISIÓN DE MENSAJES SMS DE ALERTA	83
4.6. PUBLICACIÓN DE MENSAJES DE ALERTA EN LA RED SOCIAL TWITTER	85
5. CONCLUSIONES.....	87
6. BIBLIOGRAFÍA	88
ANEXOS	94
ANEJO I. Datos de la simulación.....	95
ANEJO II. Esquema de montaje.....	97

ÍNDICE DE FIGURAS

Figura 1. Desastres naturales. Daños económicos. Años 1998-2017.....	14
Figura 2. Número de víctimas mortales en España por desastres naturales, 1995-2017.	15
Figura 3. Víctimas mortales por inundaciones y avenidas en España, 1990-2018.	16
Figura 4. Estación de aforos, Zagra. Río Turia.	20
Figura 5. Estación de Aforos. Los Frailes, río Júcar.	21
Figura 6. Sensores de nivel, sin contacto con el agua.	21
Figura 7. Sensor Piezorrestivo instalado en acequia	22
Figura 8. Principales placas Arduino.	27
Figura 9. Ejemplo de un sketch de Arduino IDE	29
Figura 10. Funcionamiento de los sensores de distancia basados en ultrasonidos	30
Figura 11. Diagrama de bloques	34
Figura 12. Placa Arduino UNO Rev3.	36
Figura 13. Conexiones de la placa Arduino UNO Rev3.	38
Figura 14. Sensor JSN R04Tv2.....	39
Figura 15. Diagrama de tiempos del sensor JSN R04Tv2	39
Figura 16. Placa del módulo JSN R04Tv2	40
Figura 17. Placa del módulo SIM 900.....	42
Figura 18. Módulo SIM 900.....	43
Figura 19. Pantalla LCD 20x4 I2C.....	44
Figura 20. Módulo de conexiones I2C	45
Figura 21. Componentes del prototipo, previa instalación.....	46
Figura 22. Conexiones de los componentes	47
Figura 23. Dispositivo montado	47
Figura 24. Esquema de montaje	48
Figura 25. Alojamiento del dispositivo en una caja estanca.	49
Figura 26. Esquema de interpretación de variables del programa	51
Figura 27. Diagrama de flujo del programa	53
Figura 28. Canal de ThingSpeak del presente proyecto.....	59
Figura 29. API Keys del canal de ThingSpeak	60
Figura 30. Logo de MIT App Inventor	61
Figura 31. Diseño de la pantalla de inicio de la App con MIT App Inventor.....	61
Figura 32. Bloques de instrucciones de MIT App Inventor	62
Figura 33. Página Web en la que se publican los datos	63
Figura 34. Programación de reacciones con ThingTwitter de ThingSpeak	64
Figura 35. Croquis explicativo del ensayo de precisión.....	66

Figura 36. Ensayo de precisión	66
Figura 37. Comprobaciones de funcionamiento con lámina de agua.....	68
Figura 38. Ubicación del dispositivo para realizar la simulación	70
Figura 39. Torrent Sarrià, a su paso por el Parc Bit, Palma. Illes Balears.	71
Figura 40. Anclaje del dispositivo.....	72
Figura 41. Dispositivo anclado sobre el torrente Sarrià	72
Figura 42. Mediciones del lecho del cauce del torrente	73
Figura 43. Croquis explicativo de la simulación.....	75
Figura 44. Resultados del ensayo de precisión	77
Figura 45. Canal de ThingSpeak donde se agregan los datos enviados por el prototipo	78
Figura 46. Detalle de la gráfica generada automáticamente con ThingSpeak	79
Figura 47. Datos de la Simulación II agregados a la plataforma ThnigSpeak	80
Figura 48. Publicación de datos en la página Web del proyecto.....	82
Figura 49. Aplicación para sistemas Android, creada en el presente proyecto.....	83
Figura 50. Envío de SMS de alerta	85
Figura 51. Publicación automática de mensajes en la red social Twitter.....	86

ÍNDICE DE TABLAS

Tabla 1. Población afectada por desastres naturales entre los años 2000-2017.....	13
Tabla 2. Principales variables analizadas por las SAIH.....	20
Tabla 3. Características de la placa Arduino UNO Rev3.....	37
Tabla 4. Características del módulo JSN R04Tv2	40
Tabla 5. Descripción de las conexiones del sensor JSN R04Tv2	41
Tabla 6. Características del módulo SIM900	42
Tabla 7. Descripción de las conexiones del módulo SIM900	43
Tabla 8. Características técnicas de la pantalla LCD 20x4 I2C	44
Tabla 9. Descripción de los pines de conexión del módulo I2C	45
Tabla 10. Principales variables del programa	50
Tabla 11. Coste material del prototipo	65
Tabla 12. Resultados del ensayo de precisión del dispositivo	67
Tabla 13. Resultados del ensayo de precisión de mediciones de altura de lámina de agua.....	69
Tabla 14. Datos recogidos por la plataforma ThingSpeak	79
Tabla 15. Datos de la Simulación II agregados a la plataforma ThnigSpeak	81
Tabla 16. Secuencia de envío y recepción de SMS.....	84
Tabla 17. Datos de la simulación	95

ACRÓNIMOS

ARPSI. Áreas de Riesgo Potencial Significativo de Inundación

DGPC. Dirección General de Protección Civil y Emergencias.

DiDIY. Hazlo tú mismo digital. Del inglés, Digital-Do-It-Yourself.

DIY. Hazlo tú mismo. Del inglés, Do-It-Yourself.

GPRS. Servicio General de Paquetes Vía Radio. Del inglés, General Packet Radio Service.

GSM. Sistema Global Para las Comunicaciones Móviles. Del inglés, Global System for Mobile.

IDE. Integrated Development Environment.

IoT. El internet de las cosas. Del inglés, Internet of Things.

LCD. Pantalla de cristal líquido. Del inglés, Liquid Cristal Display.

M2M. Máquina a Máquina. Del inglés, Machine To Machine.

MIT. Instituto de Tecnología de Massachusetts. Del inglés, Massachusetts Institute of Technology.

MITECO. Ministerio para la Transición Ecológica y el Reto Demográfico.

OMM. Organización Mundial de Meteorología

OSH. Hardware de código abierto. Del inglés, Open Source Hardware.

SAIH. Sistemas Automáticos de Información Hidrológica

SIM. Módulo de identificación de abonado. Del inglés, Subscriber Identity Module.

SMS. Servicio de Mensajes Cortos. Del inglés, Short Message Service.

SNCZI. Sistema Nacional de Cartografía de Zonas Inundables

1. INTRODUCCIÓN

1.1. CRECIDAS E INUNDACIONES

1.1.1. Definiciones

A la hora de abordar la problemática relacionada con la gestión de los riesgos y efectos de las inundaciones, es necesario conocer las interpretaciones tanto técnicas como legales de la terminología básica. Por uniformidad, se recomienda (OMM, 2011) el uso del Glosario Hidrológico Internacional, de la Organización Meteorológica Mundial (1992), en el cual se define crecida (sinónimo de avenida) como:

Elevación, generalmente rápida, del nivel del agua en un curso hasta un máximo, a partir del cual dicho nivel desciende a una velocidad menor; Flujo relativamente alto medido como nivel o caudal; Marea ascendente (OMM, 1992, p.124).

Refiriéndose a los efectos de las crecidas, el mismo glosario define inundación:

Desbordamiento del agua fuera de los confines normales de un río o cualquier masa de agua, o acumulación de agua procedente de drenajes, en zonas que normalmente no se encuentran anegadas (OMM, 1992, p.127).

La Directiva 2007/60/CE, relativa a la evaluación y gestión de los riesgos de inundación, contextualiza la definición en el marco de la realidad europea. Hace referencia a los torrentes de montaña y corrientes de agua intermitentes del Mediterráneo, como las ramblas.

Anegamiento temporal de terrenos que no están normalmente cubiertos por agua. Incluye las inundaciones ocasionadas por ríos, torrentes de montaña, corrientes de agua intermitentes del Mediterráneo y las inundaciones causadas por el mar en las zonas costeras, y puede excluir las inundaciones de las redes de alcantarillado (Parlamento Europeo, 2007, p.29).

La transposición de la citada Directiva en el Real Decreto 903/2010, de 9 de julio, de evaluación riesgos de inundación, amplía la definición de la Directiva:

Anegamiento temporal de terrenos que no están normalmente cubiertos de agua ocasionadas por desbordamiento de ríos, torrentes de montaña y demás corrientes de agua continuas o intermitentes, así como las inundaciones causadas por el mar en las zonas costeras y las producidas por la acción conjunta de ríos y mar en las zonas de transición (España, 2010, p.5).

1.1.2. Tipología de las crecidas

Como se ha comentado anteriormente, las inundaciones son consecuencia de las crecidas de agua. Se pueden establecer diferentes tipos de clasificación de inundaciones atendiendo a su origen, lugar geográfico, tipo de cauce, precipitación o masa de agua causante, e incluso el momento del año en el que ocurren (Diez-Herrero *et al.*, 2009). Este nivel de concreción es necesario a la hora de establecer el nivel de riesgo existente, y las estrategias para minimizar los daños.

El Programa de Gestión de Inundaciones, asociado a la Organización Mundial de Meteorología (OMM, 2011), clasifica las crecidas en nueve tipos:

- Crecidas repentinas. Asociadas a tormentas violentas y de corta duración, que afectan a un área pequeña. Se producen generalmente en valles angostos. La rapidez con la que ocurren las hace especialmente peligrosa para la vida humana.
- Crecidas fluviales. Las planicies de inundación o terrenos de arrastre son el escenario de las crecidas fluviales. Cuando la masa de agua excede la capacidad de los canales naturales, estos se desbordan, extendiendo la superficie del cauce.
- Crecidas de un solo evento. Es el tipo más común de inundaciones. Son consecuencia de lluvias intensas que se prolongan en el tiempo desde varias horas hasta días.
- Crecidas de eventos múltiples. Se producen a partir de precipitaciones pluviales intensas, asociadas a perturbaciones atmosféricas que se suceden una tras otra en corto tiempo.

- Crecidas estacionales. Corresponden a crecidas que, en general tienen lugar con regularidad, como resultado de precipitaciones pluviales estacionales intensas. Es el caso de las lluvias monzónicas.
- Crecidas costeras. La concurrencia de mareas altas, con oleajes de temporal y fuertes vientos, son la causa más frecuente de este tipo de inundaciones.
- Crecidas estuarinas. La marea costera se encuentra con un caudal concentrado de agua dulce de río que fluye hacia el mar. La interacción entre ambos flujos puede causar una acumulación de agua alta y rápida que se mueve hacia el interior del estuario.
- Inundaciones urbanas. Las lluvias intensas sobre pueblos y ciudades pueden ocasionar una escorrentía rápida en las zonas pavimentadas, que excede la capacidad de los sistemas de drenaje de aguas pluviales.
- Crecidas por deshielo. La acumulación de masas de hielo en terrenos elevados y en zonas de alta latitud, pueden ocasionar escorrentías si la primavera es especialmente calurosa.

Tal y como indica la OMM, son las crecidas repentinas las que, por su carácter imprevisible, las hacen más peligrosas.

1.2. EFECTOS ECONÓMICOS Y SOCIALES DE LAS INUNDACIONES

1.2.1. Contexto mundial

Los efectos desfavorables de las crecidas de agua a menudo conllevan importantes consecuencias socioeconómicas y medioambientales, entre las que figuran pérdidas de vidas y bienes; migraciones humanas y de fauna y ganadería; degradación del medio ambiente; y escasez de alimentos, energía, agua y otras necesidades básicas (OMM, 2009).

Las inundaciones han afectado a más personas que cualquier otro tipo de desastre natural en el siglo XXI. (CRED, 2020). Tal y como se recoge en la Tabla 1, desde el año 2000 hasta el 2017, las inundaciones han afectado a más de 86 millones de personas.

*Tabla 1. Población afectada por desastres naturales entre los años 2000-2017.
Fuente CRED-IDD (2018)*

	Personas afectadas	% respecto del total
Inundaciones	86.696.923	44,89%
Sequías	58.734.128	30,41%
Tormentas	34.083.106	17,65%
Terremotos	6.783.729	3,51%
Temperaturas extremas	6.368.470	3,30%
Desplazamientos de tierra	264.117	0,14%
Actividad volcánica	169.308	0,09%
Incendios	19.243	0,01%
TOTAL:	193.119.024	

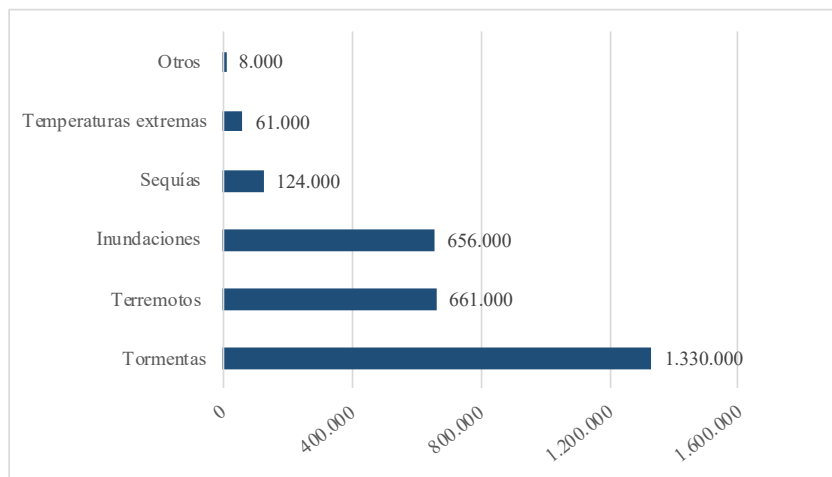
En el año 2019, se contabilizaron 396 desastres naturales en todo el mundo, de los cuales el 49% fueron inundaciones, y fueron causantes del 43% del total de muertes ocurridas a consecuencia de desastres naturales a lo largo del citado año (CRED, 2020)

Los efectos de las inundaciones presentan claras desigualdades entre países pobres y ricos. Los países con ingresos bajos presentan tasas de mortalidad más altas en términos absolutos, y también en número de muertes por millón de habitantes (CRED, 2018). Los países en desarrollo son muy vulnerables a ese tipo de peligros naturales, pues la necesidad tiende a obligar a los pobres a ocupar los territorios más expuestos (IPCC, 2015). Desde el año 1970 al 2010, la población mundial residente en zonas inundables se incrementó en un 100%, y la previsión para el 2050 es que se siga en la tendencia de (Jongman y Ward, 2012), incrementando por lo tanto la población en riesgo.

Además, en los países con menos recursos, los daños en estructuras y sistemas productivos derivan en graves déficits de nutrición, especialmente en la población infantil (Rodríguez-Llanes *et al*, 2016), y un incremento notable de las muertes provocadas por diarreas y problemas respiratorios (Alderman *et al*, 2012).

En cuanto a los países desarrollados, su vulnerabilidad aumenta con el crecimiento económico y la acumulación de bienes en las zonas inundables y en los asentamientos altamente urbanizados (OMM, 2009).

En el ámbito económico, entre los años 1998 y 2017 se produjeron pérdidas por un valor de 656.000 millones de USD\$. Situándose, tal y como se refleja en la Figura 1, tras las tormentas y los terremotos, en cuanto pérdidas económicas ocasionadas.



*Figura 1. Desastres naturales. Daños económicos. Años 1998-2017
Millones de USD\$*

Fuente: CRED (2018).

En cuanto a la Figura 1, conviene poner en relieve que Estados Unidos es el país que ha reportado más daños económicos provocados por desastres entre los años 1998-2017.

La estimación de los costos económicos de las inundaciones y otros desastres naturales es una tarea que presenta importantes dificultades (CRED, 2018). En muchos casos, los propios países no son capaces de evaluar con exactitud los costos económicos. En otros, no se transfieren los datos a los organismos internacionales. Por esta razón, se están realizando estimaciones de pérdidas económicas provocadas por inundaciones basadas en la altura de la lámina de agua (Ceballos *et al.*, 2016). Estos sistemas de modelización de daños están resultando especialmente útiles en países con bajos recursos (Bernal *et al.*, 2016).

1.2.2. Las inundaciones en España

En España, las inundaciones constituyen el fenómeno natural con mayor impacto económico y social (DGPCE, 2020). En la Figura 2 se presentan las víctimas mortales por desastre natural en el período comprendido entre los años 1995-2017. El balance total de víctimas es de 1.303, de los cuales las inundaciones con 342 fallecidos, supone el 25,25% de total (MITECO, 2018).

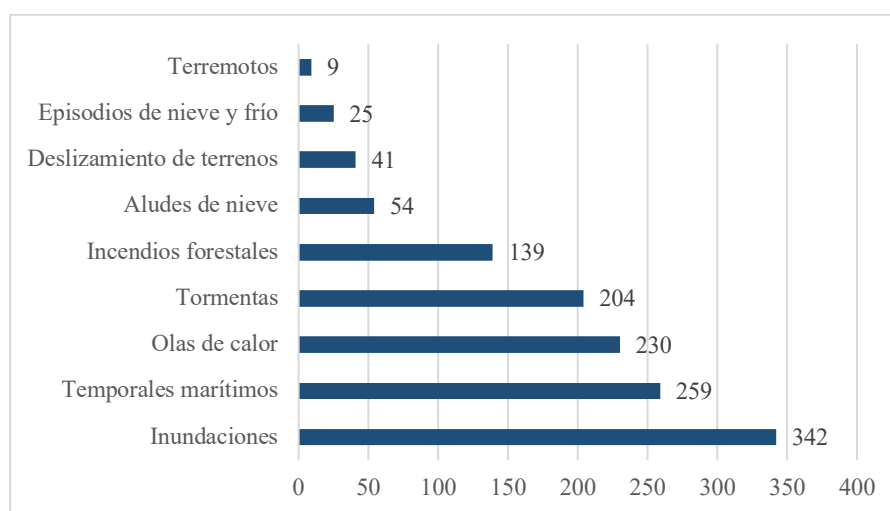


Figura 2. Número de víctimas mortales en España por desastres naturales, 1995-2017.

Fuente: DGPCE

En el análisis de víctimas a consecuencia de las inundaciones, que se presenta en la Figura 3, se observa que en el año 1996 se registraron 110 víctimas, consecuencia de la riada del camping de Biescas, y en el año 1997 se contabilizaron 40 víctimas a causa de la riada de Badajoz.

En España, Según el informe de la ONU sobre la Estrategia para la reducción de los desastres naturales, ocho millones de personas estarían expuestas a los efectos de la sequía, 49 mil se podrían ver afectadas por un terremoto y 28 mil podrían sufrir los efectos de las inundaciones (MITECO, 2018).

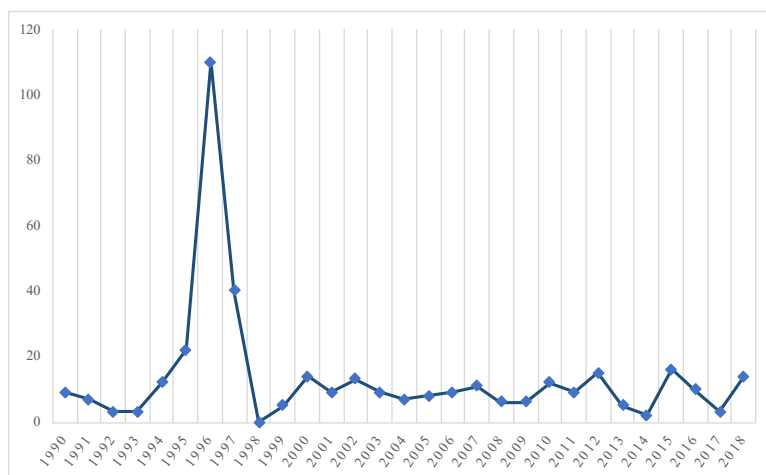


Figura 3. Víctimas mortales por inundaciones y avenidas en España, 1990-2018.

Fuente: DGPE (2018)

En cuanto al impacto económico de las inundaciones, entre los años 2000 y 2017, las inundaciones causaron daños por un valor de 14.400 millones de euros (MITECO, 2018).

1.1.5. Perspectivas ante el cambio climático

En la última década se han realizado diferentes estudios sobre el impacto que el cambio climático pueda tener en los ciclos hidrológicos. Se espera que el cambio climático incremente la frecuencia y destrucción de las inundaciones (Bustillos *et al.*, 2017). Garijo *et al.* (2018), apunta que habrá necesariamente variaciones en cuanto a magnitud y frecuencia, destacando el aumento de inundaciones provocadas por gota fría. Cajigal y Maldonado (2019) llegan a conclusiones similares al confirmar que se está reportando un aumento de la intensidad de los ciclones tropicales, y las inundaciones asociadas a estos fenómenos.

Por lo tanto, es difícil establecer predicciones concretas sobre la evolución futura de los episodios de crecidas e inundaciones, pero todo apunta a que nos dirigimos hacia un escenario en que estos fenómenos podrían ser más frecuentes e intensos.

1.2. GESTIÓN DEL RIESGO DE INUNDACIONES

La Directiva 2007/60 define el riesgo de inundación como “combinación de la probabilidad de que se produzca una inundación y de las posibles consecuencias negativas para la salud humana, el medio ambiente, el patrimonio cultural y la actividad económica, asociadas a una inundación.” Es en la evaluación del riesgo y en la gestión de este, donde la Directiva pretende ser una herramienta que permita legislar a los estados miembros para reducir las consecuencias negativas de las inundaciones.

Legislar el riesgo requiere equilibrio, pues en un exceso de celo, normativas restrictivas podrían tener consecuencias inesperadas, reduciendo las oportunidades de sustento de la población habitante de las planicies de inundación (OMM, 2009).

1.2.1. La gestión del riesgo de inundaciones en España

A partir de la Directiva 2007/60 de evaluación y gestión de los riesgos de inundación, y su transposición a la legislación española mediante el Real Decreto 903/2010 de evaluación y gestión de riesgos de inundación, se establecen las actuaciones necesarias para gestionar el riesgo. Estas actuaciones las podemos dividir en cinco etapas fundamentales:

1. Evaluación preliminar del riesgo. En el año 2011, las 16 Demarcaciones Hidrográficas, delimitaron las Áreas de Riesgo Potencial Significativo de Inundación (ARPSIs), realizada por las autoridades competentes en materia de aguas, costas y protección civil.
2. Elaboración de los mapas de peligrosidad y riesgo de Inundación. A partir de los estudios de evaluación del riesgo de inundación y de la delimitación de los (ARPSIs), se puso en funcionamiento el Sistema Nacional de Cartografía de Zonas Inundables (SNCZI), que recoge los Mapas de Peligrosidad del Riesgo de Inundación
3. Planes de Gestión del Riesgo de Inundación. La Directiva 2007/60 y el Real Decreto 903/2010, establecen que el elemento esencial de los Planes de Gestión del Riesgo Inundación es la identificación de los objetivos. La propia OMM, en

sus recomendaciones sobre la gestión de las inundaciones (OMM, 2009) ya advierte que el elemento esencial del debate entre los actores implicados es la definición clara y consensuada de los objetivos.

Se exponen a continuación los objetivos de los Planes de Gestión del Riesgo de Inundación recogidos en el RD 903/2010:

- a. Incremento de la percepción del riesgo de inundación y de las estrategias de autoprotección en la población, los agentes sociales y económicos.
- b. Mejorar la coordinación administrativa entre todos los actores involucrados en la gestión del riesgo.
- c. Mejorar el conocimiento para la adecuada gestión del riesgo de inundación.
- d. Mejorar la capacidad predictiva ante situaciones de avenida e inundaciones.
- e. Contribuir a mejorar la ordenación del territorio y la gestión de la exposición en las zonas inundables.
- f. Conseguir una reducción, en la medida de lo posible, del riesgo a través de la disminución de la peligrosidad para la salud humana, las actividades económicas, el patrimonio cultural y el medio ambiente en las zonas inundables.
- g. Mejorar la resiliencia y disminuir la vulnerabilidad de los elementos ubicados en las zonas inundables.
- h. Contribuir a la mejora o al mantenimiento del buen estado de las masas de agua a través de la mejora de sus condiciones hidromorfológicas.

1.2.2. Previsión de crecidas

En el Plan de Aplicación de las Decisiones de Cumbre Mundial sobre el Desarrollo Sostenible, Johannesburgo (2002), se puso en relieve la necesidad de establecer sistemas de alerta temprana para mitigar los riesgos derivados de las inundaciones, y de promover el acceso y la transferencia de esta tecnología. Una pieza indispensable de cualquier

sistema de alerta temprana es la previsión de las crecidas en aquellos puntos clasificados de riesgo de inundación. Esta previsión se realiza a partir de datos meteorológicos y de aforo.

El análisis de los datos meteorológicos es una herramienta fundamental en la previsión de las crecidas. En este sentido, la metodología estadística aplicada para la obtención de conclusiones es clave. Garrijo *et al.* (2018) demostraron que los métodos de regionalización estadística han subestimado las precipitaciones extremas. Siendo este efecto determinante en el área mediterránea de España. Por esta razón se debe recurrir a la combinación de datos tanto meteorológicos como de caudal. En este sentido, la Organización Mundial de la Meteorología, en su Manual sobre predicción y avisos de crecidas (2011) pone en relieve que, a excepción de las cuencas fluviales más importantes, la densidad de estaciones meteorológicas y de instalaciones de información de caudal para la gestión de recursos resulta insuficiente en la mayoría de los casos para que ellas puedan operar como parte de la red de previsión y alerta de crecidas.

1.2.2.1. Sistema Automático de Información Hidrológica

En España, las diferentes Confederaciones Hidrográficas, disponen de Sistemas Automáticos de Información Hidrológica (SAIH) que permiten tanto la gestión de los recursos hídricos, como la previsión de las crecidas. Constituyen una red de recogida y tratamiento de datos meteorológicos, de caudal circulante y de recursos hídricos. Las redes SAIH tienen capacidad de comunicar datos de manera telemática.

En la Tabla 2 se presentan las principales variables analizadas por las SAIH.

Tabla 2. Principales variables analizadas por las SAIH

Embalses, presas y azudes	Nivel de embalse
	Caudal de salida
	Posición de compuertas
Ríos, ramblas, canales y conducciones	Nivel y caudal
Estaciones meteorológicas	Temperatura, evaporación
	Precipitación líquida y sólida
	Nivel y densidad de nieve acumulada

Los sistemas de medición de niveles y caudales de los SAIH combinan las estaciones clásicas de aforo, como las representadas en las Figuras 4 y 5, junto con sensores de nivel sin contacto como el representado en la Figura 6, o el sensor piezoresistivo de la Figura 7.



*Figura 4. Estación de aforos, Zagra. Río Turia.
Fuente CHJ (2020)*



*Figura 5. Estación de Aforos. Los Frailes, río Júcar.
Fuente CHJ (2020)*



*Figura 6. Sensores de nivel, sin contacto con el agua.
Río Segura aguas arriba de Orihuela. Fuente CH Segura (2020)*



Figura 7. Sensor Piezorrestivo instalado en acequia
Fuente CH Segura (2020)

A instancias de la Subdirección General de Planificación y Uso Sostenible del Agua, en el año 2017 (Azcorra *et al.*) se evaluaron los sistemas de comunicación de los SAIH. En dicha evaluación se resaltó la heterogeneidad del actual sistema de comunicaciones y transmisión de datos en las diferentes SAIH, y en la necesidad de mudar a tecnologías inalámbricas de madurez y solvencia contrastadas, además de incorporar tecnologías M2M (Machine To Machine) e IOT (Internet of Things).

1.2.3. Comunicación del riesgo

Los sistemas de alerta por si solos no son una solución. Han de ir acompañados de mensajes claros y precisos, y además la población ha de estar concienciada y preparada para actuar. Por lo tanto, un paso previo es la educación y formación de la población objetivo para que, en caso de necesidad de proceder una evacuación, ésta se realice de forma ordenada y efectiva (OMM, 2009).

La comunicación del riesgo de inundación a la población refuerza la conciencia del peligro y motiva a las personas a realizar acciones preventivas (Kellens *et al.*, 2013). Aspectos tan importantes como el grado de conocimiento de Plan de Protección Civil se ven reforzados si la población ha sido previamente informada del riesgo (Bodoque *et al.*, 2019).

Los servicios de alerta de crecidas pueden estar perfectamente organizados a un nivel técnico y administrativo, pero la percepción del riesgo y de la alerta depende de las estructuras y marcos sociales, lo cuales son variables y a menudo imprevisibles. El objetivo, según Manual sobre Predicción y Avisos de Crecidas (OMM, 2011) es que las alertas deben ser recibidas por todas las personas.

1.3. HARDWARE Y SOFTWARE DE CÓDIGO ABIERTO EN EL CONTEXTO RURAL

Se entiende como hardware de código abierto, (OSH por sus siglas en inglés), aquel hardware cuyo diseño se pone a disponibilidad de los usuarios para que pueda ser estudiado, modificado, distribuido, reproducido y vendido (OSHW Association, 2019). La idea del hardware de código abierto no es nueva, se remonta a los principios de la electrónica (Ackerman, 2009) cuando los primeros radioaficionados construían sus propios receptores. Actualmente el mercado del OSH es amplio y creciente. Empresas como Adafruit, Sparkfun, Seed Studios, MakerBot Ind o Arduino han logrado poner a disposición de los usuarios todo un amplio abanico de dispositivos, placas programables, sensores y módulos de comunicación.

En el ámbito de la producción agraria y entorno rural, la plataforma de código abierto Arduino está facilitando a usuarios de distintos países, especialmente India, México y Brasil, el desarrollo de soluciones tecnológicas de bajo coste (Cuauhtemoc *et al.*, 2018). Los proyectos realizados con Arduino abarcan todo el abanico de la producción agraria. Se exponen a continuación las principales temáticas:

- Monitoreo de parámetros climáticos y agronómicos (Hashim *et al.*, 2015).
- Geolocalización (Ruiz, 2016).
- Automatización de tareas como el riego y la fertilización (Yilmaz, 2017), producción en invernaderos (Kouhia, 2016) y ganadería (Henry, 2016).
- Agricultura de precisión (Lowrance, 2012).
- Robótica y drones aplicada a tareas agrarias (Shuvrodeb *et al.*, 2017).
- Producción de energía (Lopez *et al.*, 2016).
- Protección de cultivos (Rashid, 2017).
- IoT, Internet of things (Castillo, 2017).
- Investigación agraria (Tharmatharsanan, *et al.* 2016) (D'Ausilio, 2012).
- Procesado de alimentos e industrias agroalimentarias (Poojary *et al.*, 2016).

En lo que se refiere a la investigación agraria, se han realizado interesantes proyectos en los que el uso de Arduino, junto con sensores y módulos de código abierto, han posibilitado disponer de instrumentos de medida para uso en laboratorio o campo en entornos con bajos recursos (Fisher, 2012) (Moscardó, 2019). Arduino ha demostrado ser una plataforma lo suficiente precisa como para ser utilizada en laboratorio (D'Ausilio, 2011). Además, Arduino cuenta con la ventaja de su bajo precio, alimentación con baterías convencionales de 9V, todo un abanico de sensores compatibles de bajo coste, no son necesarios conocimientos previos de programación y existe un amplio repositorio de programas disponibles en las comunidades de Arduino (Moscardó, 2019).

1.3.1. Tecnología DiDIY en el contexto rural

El movimiento DIY ,Do-It-Yourself (Hazlo tú mismo), surge en la década de los años 20 del pasado s.XX. Se puede definir como la creación, modificación o reparación de objetos tecnológicos sin la ayuda de profesionales remunerados (Kuznetsov *et al.*, 2010). Este movimiento vive, desde principios del 2010, un importante resurgir gracias a la concurrencia de diferentes factores:

- La expiración de patentes en el campo de la fabricación digital y la microelectrónica (Fox, 2014).

- El acceso a Internet como fuente de conocimiento, discusión, co-creación y promoción de ideas (Tabarés, *et al.* 2016).
- El auge de las plataformas de hardware y software de Código Abierto. También conocido como Open Source.

Esta vinculación del movimiento DiY con las actuales tecnologías digitales ha provocado su redefinición como "Digital-Do-It-Yourself" (DiDIY). Este movimiento no es únicamente urbano, el ámbito rural supone un contexto en el que DiDIY está en continuo crecimiento. Podemos resumir las claves de su éxito en el entorno rural:

- El coste de la electrónica siempre ha sido un hándicap para su implantación en entorno rural (Sadler *et al.*, 2016). En el caso de la agricultura de precisión, los pequeños y medianos agricultores no pueden afrontar los costes de esta tecnología (EURACTIV, 2015), por lo que disponer de tecnología de bajo coste, fácil de reproducir y adaptable, supone un aliciente para su implantación.
- Las actividades que se desarrollan en el contexto productivo rural son enormemente variadas. Los productores se encuentran día a día con problemas que no siempre tienen una solución comercializada. DiDIY les permite adaptar directamente la solución a un problema concreto, incorporando nueva tecnología, a la vez que adaptan o modifican la existente (Tabarés, 2018).
- La tecnología, debido tanto a su coste de adquisición, como a las patentes y copyright que la protegen, genera dependencia a los agricultores que no siempre pueden afrontar (Carolan, 2017). Además, las tecnologías con capacidad de recopilar información (Big Data) están generando un fundamentado recelo entre los agricultores sobre el uso que las empresas pueden dar a estos datos. Por esta razón la tecnología DiDIY, y el hardware y software de código abierto están cada vez más presente en el entorno rural.
- El acceso a internet en el entorno rural posibilita la participación en redes sociales y comunidades virtuales colaborativas sobre DiDIY. También, internet facilita un acceso inmediato a webs, tutoriales y manuales bajo licencia pública (Creative Commons, 2020) donde desarrollar un aprendizaje autónomo y encontrar la información necesaria para construir por uno mismo el prototipo diseñado.

- Por otro lado, DiDIY tiene también una importante dimensión social de intercambio y cooperación (DiDIY, 2015), que es muy atractiva en contextos de dispersión y baja densidad de población, como es el entorno rural.

En la práctica, los agricultores y las comunidades rurales en general pueden beneficiarse de DiDIY en (al menos) cuatro áreas principales (Fioretti *et al.*, 2016):

- Construcción y mantenimiento de máquinas agrícolas y otras herramientas, o de sus piezas de repuesto. La tecnología de impresión 3D permite el diseño e impresión dando solución a necesidades fundamentales del entorno rural (Pearce, 2015).
- Monitoreo remoto y automático de la tierra, el ganado, instalaciones y equipos. El uso de sensores de bajo coste aplicados a la actividad agraria eleva a un nivel superior la agricultura tradicional (Meenakshi, 2016).
- Automatización y control remoto de numerosas operaciones agrícolas (Vieira *et al.*, 2003).
- Intercambio de información y conocimientos (DiDIY, 2015).

1.3.2. Arduino

Arduino es una plataforma de hardware y software de código abierto (Arduino, 2020). Se define como plataforma debido a que Arduino cuenta con un amplio abanico de placas hardware programables (fig.8). Las dimensiones y características de cada placa son diferentes, lo cual amplía la versatilidad de Arduino para el desarrollo de proyectos.

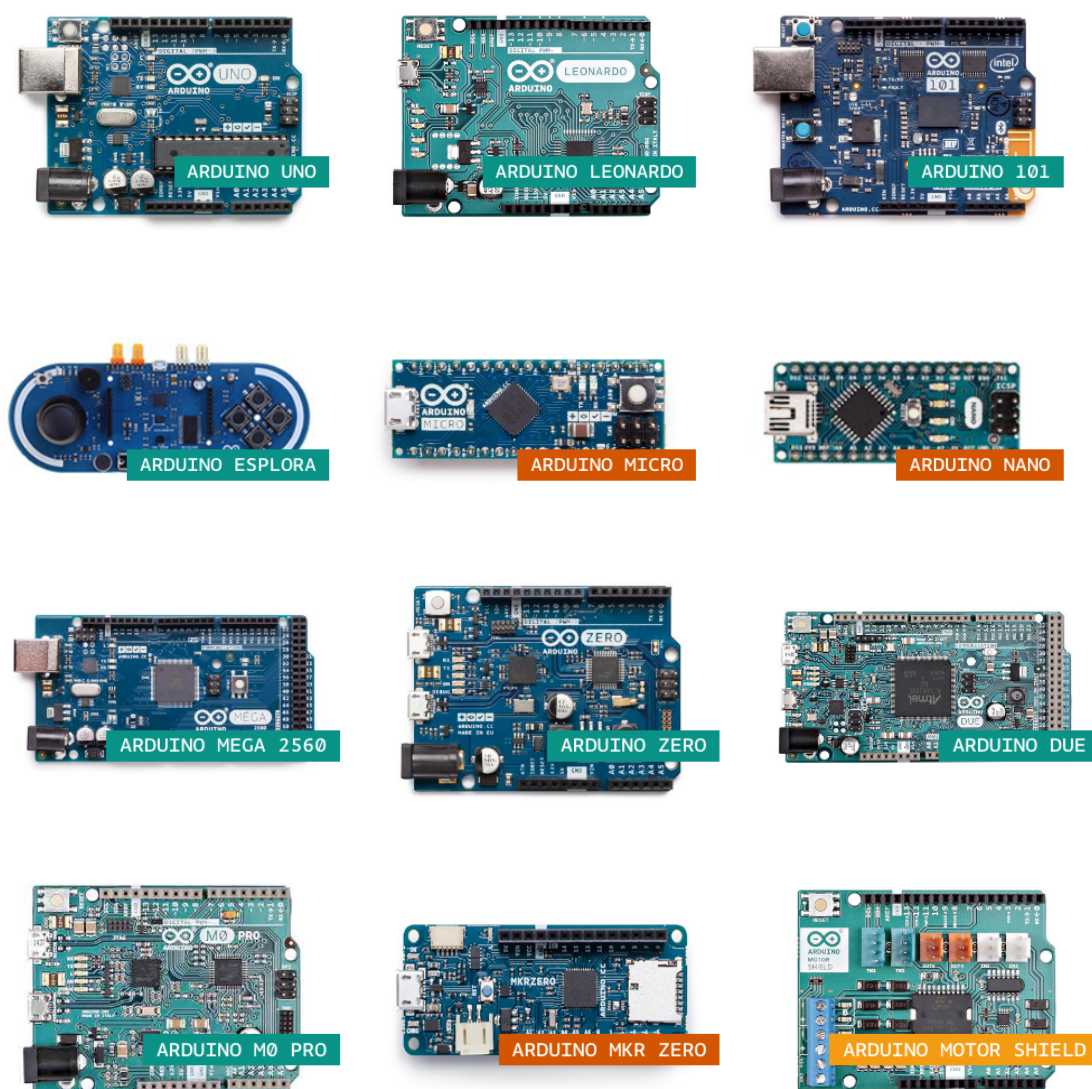


Figura 8. Principales placas Arduino.

Fuente (Arduino, 2020).

Arduino nace en el año 2005, en el Instituto de Diseño Interactivo de Ivrea, Italia. Creado por un equipo formado por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis, su primer objetivo era facilitar a los estudiantes un dispositivo de bajo coste y que funcionase en cualquier sistema operativo. Desde entonces, Arduino ha hecho posible el desarrollo de infinidad de proyectos, desde los más sencillo y doméstico, hasta complejos instrumentos científicos. Las características que han permitido que Arduino sea utilizado por tantos usuarios se pueden resumir en los siguientes puntos (Moscardó, 2019):

- Hardware de bajo coste. Las placas Arduino son relativamente económicas. Además, al ser de código abierto, en el mercado podemos encontrar placas compatibles a precios aun más económicos.
- Multiplataforma: el software Arduino (IDE), de código abierto, es ejecutable en los sistemas operativos Windows, Macintosh OSX y GNU-Linux.
- Facilidad de programación: Arduino IDE no es un entorno de programación complejo.
- Software de código abierto y extensible: el software Arduino es de código abierto, disponibles para la extensión por programadores experimentados. El lenguaje se puede expandir a través de las bibliotecas de C ++.
- Hardware de código abierto y expandible: las placas de Arduino se publican bajo una licencia de Creative Commons. Diseñadores de circuitos experimentados pueden crear su propia versión del módulo, ampliarlo y mejorarlo. Incluso los usuarios relativamente inexpertos pueden construir su propia versión.
- Comunidad de usuarios Arduino: Un importante factor del éxito de Arduino es la comunidad de usuarios. En internet podemos encontrar tutoriales, explicaciones, proyectos publicados y foros para intercambiar ideas, opiniones y trabajar de manera colaborativa.

1.3.2.1. Arduino Software

El software de Arduino, Arduino IDE (Integrated Development Environment) es el programa oficial de Arduino que permite interactuar entre el ordenador y el Hardware Arduino (Banzi, 200). Es un programa de código abierto El código mediante el cual escribimos en nuestra computadora es traducido en Lenguaje C. Posteriormente, el compilador del IDE traduce a código binario el programa, el cual puede ser interpretado por la placa Arduino. El código fuente para el IDE se publica bajo la Licencia Pública General de GNU. Es compatible con los principales sistemas operativos: GNU-Linux, Mac y Windows.

La Figura 9, muestra una imagen de la interfaz de programación de Arduino IDE. Representa un sencillo programa que haría parpadear de manera indefinida una luz LED conectada al pin digital número 13 de una placa Arduino UNO. La imagen se acompaña con comentarios para así poder entender las partes y funciones de un Sketch (programa) básico en Arduino IDE.

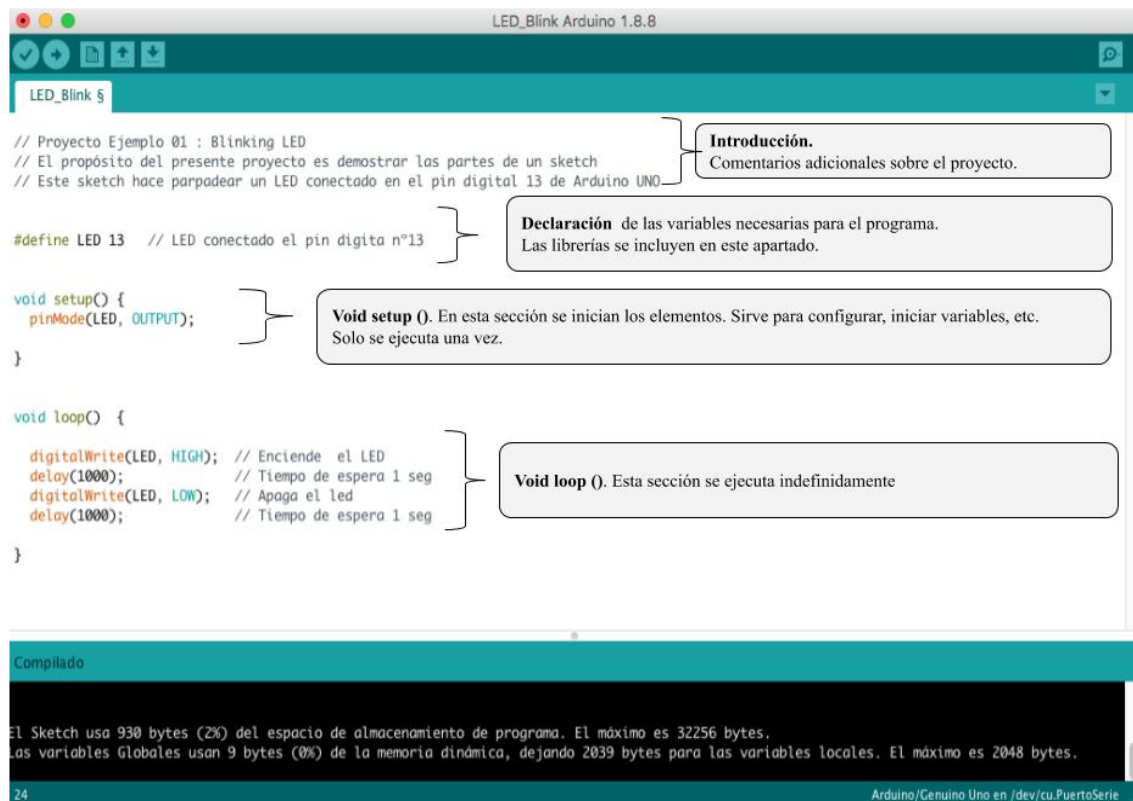


Figura 9. Ejemplo de un sketch de Arduino IDE

1.3.3. Sensores y módulos de código abierto

Un sensor es un transductor que convierte algún fenómeno físico en señales eléctricas. En hardware de código abierto podemos encontrar una amplia gama de sensores y módulos para la comunicación, visualización, geolocalización, actuadores, memoria o expansión. Empresas como Adafruit (2020) o Sparkfun Electronics (2020), además de vender estos productos, cuentan con foros, videos demostrativos y tutoriales para facilitar el aprendizaje autónomo y compartir proyectos.

1.4. DETECCIÓN DEL NIVEL Y CAUDAL DE AGUA MEDIANTE ULTRASONIDOS

Los ultrasonidos son ondas acústicas de frecuencia superior a las detectadas por el oído humano (>20 kHz). Los dispositivos de ultrasonidos que permiten determinar la distancia entre un sensor (emisor y receptor) y un obstáculo funcionan (fig.10) emitiendo una secuencia de pulsos de ultrasonidos, y registrando el tiempo que tarda el eco del pulso en volver al sensor.

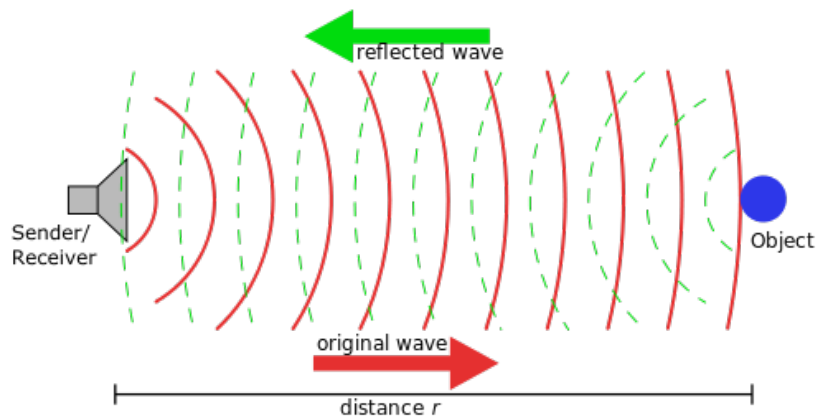


Figura 10. Funcionamiento de los sensores de distancia basados en ultrasonidos

Fuente: Wikimedia Commons

La distancia entre el sensor y el objeto se calcula mediante la ecuación:

$$d = t \cdot V / 2$$

Donde d es la distancia, t es tiempo, y V velocidad del sonido. La velocidad del sonido depende de la temperatura y humedad del aire (Han et al., 2019). No obstante, podemos determinar con bastante precisión la velocidad del sonido mediante la ecuación:

$$V = 331.4 + 0.6T$$

Donde V es velocidad del sonido (m/s), y T es temperatura del aire. Si consideramos una temperatura de 20°C obtendríamos una velocidad de 343,4 m/s.

La Organización Mundial de Meteorología, en su Guía de Prácticas Hidrológicas (2008), considera que los limnígrafos basados en la aplicación de ultrasonidos, son dispositivos útiles para determinar el nivel de agua. En la misma guía recomienda que en las estaciones de medición de registro continuo, las mediciones con intervalos de una hora son suficientes. Para mediciones de corrientes fluviales pequeñas o con crecidas repentinas y cuencas de drenaje urbanas sería necesario registrar la altura de agua con mayor frecuencia, de esta manera se podrán obtener hidrogramas más precisos. Las limitaciones en cuanto a la frecuencia de las mediciones vendrán impuestas por la capacidad de la batería y la memoria de almacenamiento de datos.

1.4.1. Dispositivos de medición del nivel de agua basados en el hardware de código abierto

En los últimos años se han desarrollado diferentes dispositivos de medición de nivel agua y alerta de inundaciones basados en la aplicación de ultrasonidos y plataformas de hardware libre. Podemos destacar el sistema de alerta temprana basado en el sensor HSR04 y el microcontrolador ATMEL AT89S51 (Kuantama *et al.*, 2012); el sistema de monitoreo y alerta mediante el sensor HC-SR04, la plataforma Arduino y módulo de comunicación WIFI ESP8266 para la interconexión IoT (Diriyana *et al.*, 2019); y el uso del sensor LV-EZ3, junto con Arduino y el módulo de comunicación GSM/GPRS SIM900 (Odhli *et al.*, 2016).

El conjunto de prototipos destaca por su sencillez, economía, bajo consumo y versatilidad.

2. OBJETIVOS

Desarrollar, mediante hardware y software con licencia de código abierto, un sistema de monitoreo, detección y alerta de inundaciones aplicable a torrentes, ramblas y ríos, que mediante el uso de redes inalámbricas y plataformas IoT, facilite la difusión de avisos de alerta.

3. MATERIAL, METODOLOGÍA Y PUESTA A PUNTO

3.1. DESCRIPCIÓN DEL DISPOSITIVO

El prototipo diseñado, basado en la programación de una placa Arduino UNO Rev3, determina, mediante el sensor de ultrasonidos JSN SR04v2, la altura y su variación en el tiempo de una lámina de agua. Conforme se realizan las lecturas, estas se proyectan en una pantalla LCD 16x4. El dispositivo cuenta con un módulo de comunicación GSM/GPRS el cual, mediante una tarjeta SIM con servicio de suscripción a internet, agrega los datos en la plataforma de análisis de IoT, ThingSpeak. La información que se incorpora a esta plataforma puede ser rescatada para su análisis con aplicaciones informáticas de hoja de cálculo. Una vez los datos se han añadido a ThingSpeak, estos se publican automáticamente en las siguientes plataformas:

- Aplicación móvil creada con App Inventor. Los usuarios que dispongan de la aplicación tienen acceso en todo momento a las lecturas de nivel de lámina de agua.
- Página web de acceso público, en la que se visualizan las lecturas del prototipo junto con información de servicio para dar a conocer los planes de evacuación y emergencia ante episodios de inundaciones.

El prototipo se programa para enviar mensajes de alerta en el caso de que la lámina de agua supere el nivel establecido como crítico. En ese momento se realizan las siguientes alertas:

- Se envía un SMS a un número de teléfono previamente incorporado en el programa, informando que el nivel de agua ha llegado al nivel señalado como crítico. Posteriormente, el sistema irá enviando sucesivos SMS toda vez que la lámina de agua incremente su altura en un valor determinado.
- Publicación automática de un mensaje digital, Tuit, en la red social Twitter[®]. Los usuarios suscritos al canal reciben un Tuit, alertando de la subida del nivel de agua.

En la Figura 11 se representa el diagrama de bloques del prototipo, en el que se pueden identificar los componentes (Hardware) y el cuerpo de comunicaciones que el dispositivo establece.

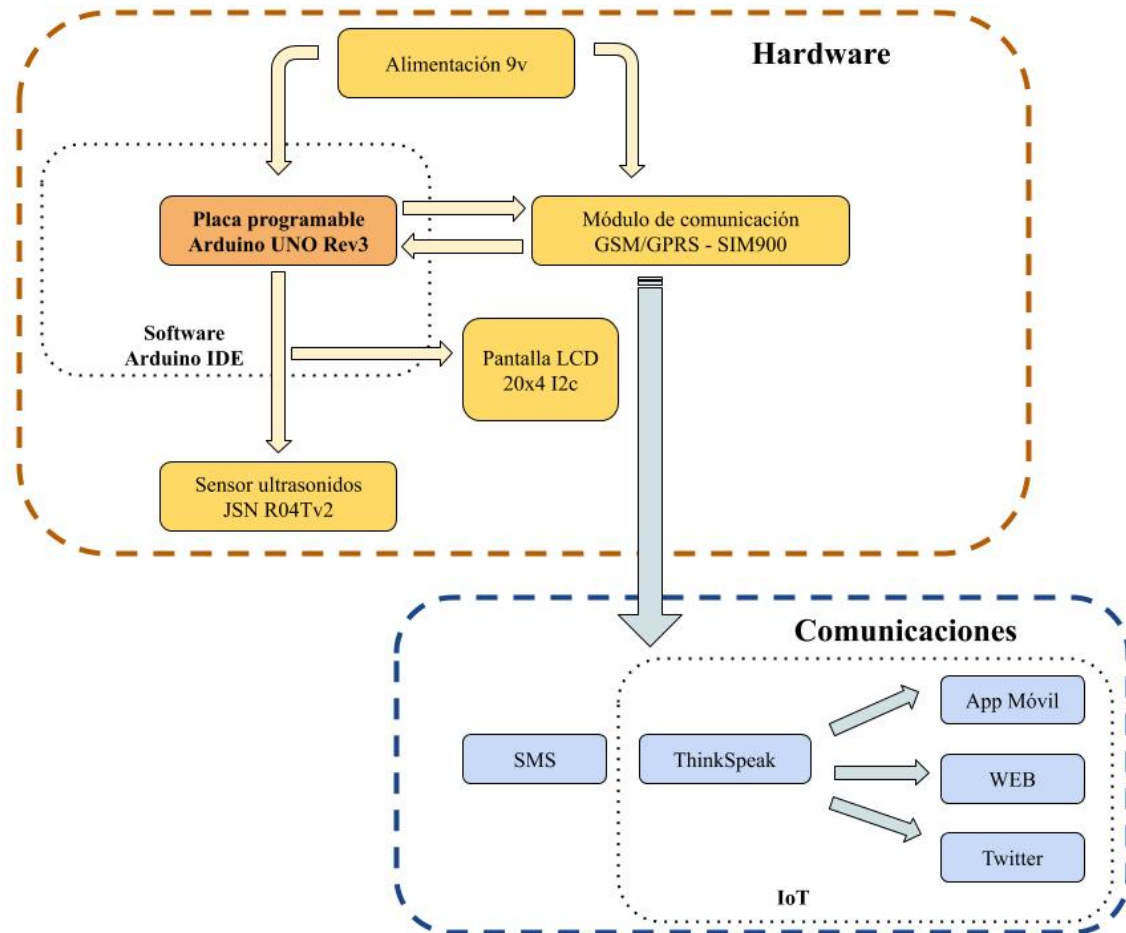


Figura 11. Diagrama de bloques

En lo que respecta a las comunicaciones, y tal como se recoge en la Figura 11, podemos identificar aquellas que se enmarcan en ámbito de IoT, pues están interconectadas y de manera automatizada establecen un diálogo con la información previamente captada con el prototipo.

En lo que respecta al Software, o soporte lógico que permite la interacción del hardware, el grueso del proyecto lo forman el conjunto de instrucciones del programa ejecutable en la placa Arduino UNO Rev3. El entorno de programación es el Arduino IDE que una vez escrito en nuestra computadora, es traducido en Lenguaje C. Posteriormente, el

compilador del IDE traduce a código binario el programa, el cual puede ser interpretado por la placa Arduino.

Para que los datos obtenidos por el prototipo puedan ser agregados de manera inalámbrica a la plataforma de IoT ThingSpeak, es necesario configurar el canal de envío y recepción de datos. En este proyecto también se ha programado el diálogo automatizado entre las diferentes plataformas que componen el entorno IoT del mismo: ThingSpeak, la aplicación móvil diseñada con MIT App Inventor, la red social Twitter y la página Web que recibirá y publicará los datos.

La aplicación móvil creada para facilitar la recepción e interpretación de los datos obtenidos por el prototipo ha sido diseñada mediante MIT App Inventor. Es una aplicación de código abierto de programación visual. Se fundamenta en el sistema de edición mediante bloques.

3.1.1. Hardware

El prototipo cuenta con el siguiente hardware:

- Placa programable Arduino UNO Rev3.
- Sensor de ultrasonidos resistente al agua JSN R04T v2.
- Pantalla LCD 26x4 con módulo de comunicaciones I2c.
- Módulo de comunicaciones GSP/GPRS SIM900.

En los siguientes epígrafes, se detallan cada uno de los componentes.

3.1.1.1. Microcontrolador Arduino UNO Rev3

Entre todas las placas de la plataforma Arduino, destaca Arduino UNO Rev3. Es la placa de iniciación a la programación, heredera de la primera placa creada por el equipo de Massimo Banzi. La Figura 12 muestra la versión más actual de esta placa programable, la Rev3.



Figura 12. Placa Arduino UNO Rev3.

Fuente, Arduino (2020)

En la Tabla 3 se muestran las características principales de Arduino UNO Rev3.

Tabla 3. Características de la placa Arduino UNO Rev3.

Microcontrolador	ATmega328P
Voltaje de trabajo	5V DC
Voltaje de entrada recomendado	7-12V DC
Voltaje de entrada límite	6- 20 V DC
Amperaje por I/O Pin	20 mA
Memoria Flash	32 KB (ATmega328P), de los cuales 0.5 Kb son utilizados para el bootloader.
SRAM	2KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Reloj, velocidad	16 MHz
Largo	68.6 mm
Ancho	53.4 mm
Peso	25 g

La placa Arduino UNO Rev3 cuenta con pines de conexión para entrada y salida de información analógica y digital, así como diferentes conexiones para la alimentación del dispositivo y la transferencia de programas desde el PC a la placa. A continuación, se describen todas estas conexiones, las cuales se detallan en la Figura 13.

- 12 pines Digitales (pines 2–13). Estos pueden ser entradas o salidas de datos, quedan especificados por el programa que se cargará en la placa. En la figura 13 se pueden identificar los pines y su nomenclatura.
- 2 pines de conexión para puertos de serie RX (recepción) y TX (transmisión). Pines 0 y 1.
- 6 pines analógicos (pines 0–5) Estos pines toman valores analógicos (es decir, lecturas de voltaje de un sensor) y los convierten en un número entre 0 y 1023 (8 bits). En la figura 13 aparecen como Analog In/Out.
- Pines de alimentación 5v y 3v. Ambos pines permiten la alimentación de dispositivos conectados a la placa Arduino. Por otro lado, el pin de 5v permite alimentar la placa desde una fuente externa.
- Conexión USB (Universal Serial Bus). Mediante esta conexión se transfieren los programas a la placa, a la vez que se puede alimentar el dispositivo.
- Entrada de alimentación de corriente continua de 9V.

- ICSP (In Chip Serial Programmer) permite el acceso a la memoria de programa del AVR (Flash) directamente desde el PC al microcontrolador sin usar el puerto USB.
- Microcontrolador ATmega328P. Es un microcontrolador CMOS de 8 bits de baja potencia basado en la arquitectura AVR®.

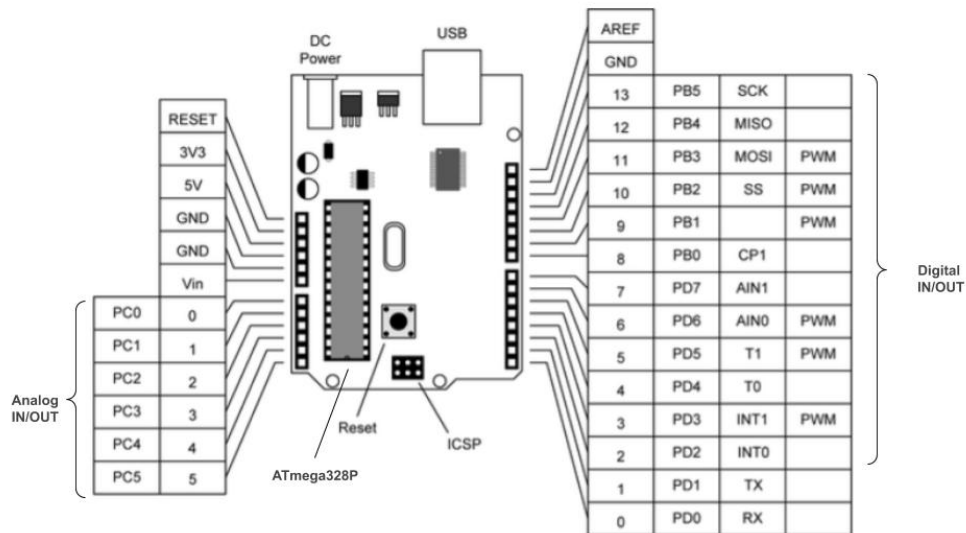


Figura 13. Conexiones de la placa Arduino UNO Rev3.

3.1.1.2. Sensor de ultrasonidos JSN R04Tv2

El sensor de ultrasonidos resistente al agua JSN R04Tv2 (Fig. 14), permite la estimar la distancia entre el sensor y un objeto situado en un rango de 20cm a 600cm (Jahankit, 2020). Como se puede apreciar en la Figura 14 el dispositivo cuenta con una placa de circuito impreso con 4 pines para su conexión, y un emisor-receptor de ultrasonidos.



Figura 14. Sensor JSN R04Tv2

El módulo JSN R04Tv2 permite calcular la distancia entre el dispositivo y un objeto mediante la transmisión y recepción de pulsos ultrasónicos. El sensor funciona enviando una señal ultrasónica de ocho pulsos con una frecuencia de 40 kHz. La señal reflejada es recibida por el pin Echo (Fig. 15). La medida de distancia del objeto se obtiene mediante la ecuación $V=331.4 + 0.6T$, donde T es la temperatura del aire tal y como se ha descrito en el epígrafe 1.4 del presente trabajo. En el desarrollo del presente trabajo se ha considerado una temperatura de 20°C, por lo que obtendríamos una velocidad de 343,4 m/s.

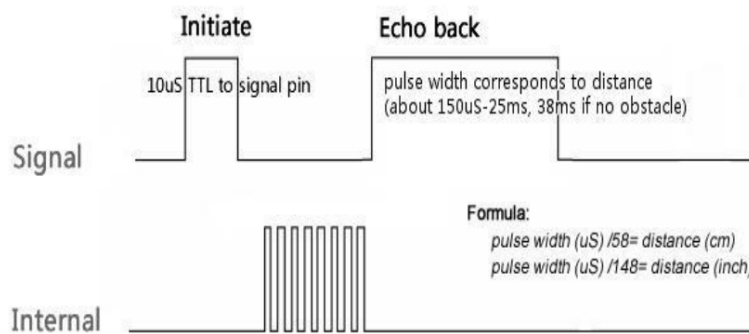


Figura 15. Diagrama de tiempos del sensor JSN R04Tv2

Las características del módulo JSN R04Tv2 se detallan en la Tabla 4. Como se puede apreciar, la alimentación del dispositivo puede realizarse mediante 3.0V o 5.0V. Para lecturas más estables, se recomienda la alimentación mediante 5.0V.

Tabla 4. Características del módulo JSN R04Tv2

Parámetro	Características
Alimentación	DC 3.0, 5.0 V
Intensidad de trabajo	8mA
Frecuencia de trabajo	40KHz
Rango de trabajo	20 cm – 600 cm
Precisión	+ - 1cm
Ángulo de medida	75 grados
Temperatura de trabajo	-20° C, 70°C

En la Figura 16, se pueden apreciar los pines de conexión del módulo JSN R04Tv2. En la Tabla 5, se detallan la función de cada uno de los pines.

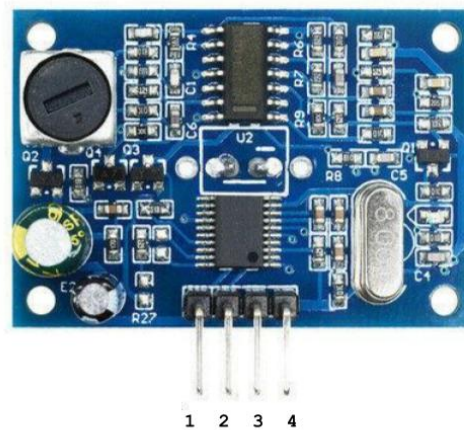


Figura 16. Placa del módulo JSN R04Tv2

Tabla 5. Descripción de las conexiones del sensor JSN R04Tv2

Pin N°	Símbolo	Descripción
1	GND	Ground - Tierra
2	Echo	Echo salida (output) del Sensor (TTL)
3	Trig	Trigger entrada (input) del sensor (TTL)
4	VCC	Alimentación +5V DC (4.5V min – 5.5V max)

3.1.1.3. Módulo de comunicación GSM/GPRS SIM900

GSM, Sistema Global Para las Comunicaciones Móviles (Global System for Mobile) es un sistema de comunicación digital de voz y datos. Nació el año 1987 como estándar internacional de comunicaciones digitales móviles. El éxito del sistema se basa en que la información se comprime, digitaliza y encripta (Sánchez, 2005). De esta manera facilita la comunicación, permitiendo que múltiples llamadas compartan canal, a la vez que asegura la privacidad, integridad y confidencialidad. El sistema permite la transmisión de datos a 9.6 kbt/s, lo cual facilita el Servicio de Mensajes SMS (Short Message Service). Este sistema opera en Europa en la banda de 900 MHz.

GPRS, Servicio General de Paquetes Vía Radio, General Packet Radio Service, en sus siglas en inglés. Fue desarrollado por el Instituto de Telecomunicaciones Estándar Europeo (ETSI) para la comunicación de paquetes de datos a través de la red GSM. Se diseñó para utilizar internet en los dispositivos móviles, además de servicios derivados como la transferencia de ficheros. La velocidad máxima teórica del GPRS es de 171kb/s.

El módulo de conexión GSM/GPRS se basa en el SIM 900 de SIMCOM (SIMCom, 2010) (Fig. 17).



Figura 17. Placa del módulo SIM 900.

Fuente (SIMCom,2010)

El módulo SIM 900 es compatible con Arduino UNO. Se controla mediante comandos Hayes, también conocidos como comandos AT, debido a que todas las instrucciones tienen como inicio las siglas AT de “Attention”.

En la Tabla 6. se detallan las características del módulo SIM900.

Tabla 6. Características del módulo SIM900

Parámetro	Características
Potencia de transmisión	2W 850 / 900 Mhz
Control	Comandos AT
Bandas de trabajo	850/900/1800/1900 MHz
Consumo	15 mA en modo sleep
Alimentación	5 ~ 18V
Temperatura de trabajo	-40°C ~ 85°C
Bandas que soporta	GSM850, EGSM900, DCS1800 y PCS1900
SIM de trabajo	2G SIM
Comunicación	Envío y recepción de llamadas Envío y recepción de SMS Recepción de radio FM

En la Figura 18 se pueden observar las conexiones de la SIM900. La Tabla 7 recoge la descripción de las conexiones de esta tarjeta de comunicaciones.

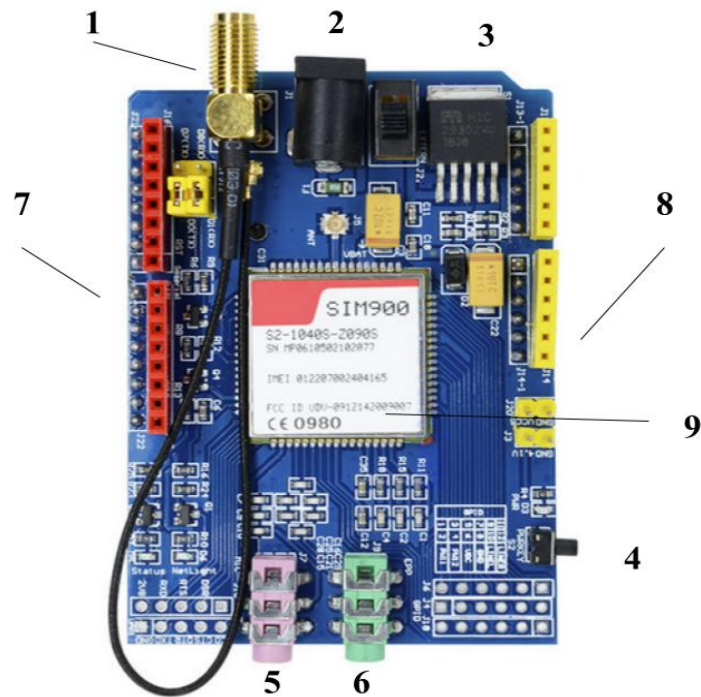


Figura 18. Módulo SIM 900

Tabla 7. Descripción de las conexiones del módulo SIM900

Código	Descripción
1	Antena
2	Alimentación DC
3	Regulador de voltaje
4	Encendido
5	Auricular
6	Micrófono
7	Bus de comunicaciones
8	Bus de conexiones para encendido y apagado remoto

3.1.1.4. Pantalla LCD 20x4 I2c

Las pantallas de cristal líquido LCD (Liquid Cristal Display) o display LCD pueden mostrar caracteres alfanuméricos. La pantalla LCD de 20x4 I2c (Fig. 19) está basada en

el controlador HD44780 de Hitachi, puede mostrar 20 caracteres por línea y cuenta con 4 líneas en la pantalla. Cada carácter se muestra en una matriz de 5x7 píxeles.



Figura 19. Pantalla LCD 20x4 I2C

En la Tabla 8 se recogen las especificaciones de esta pantalla.

Tabla 8. Características técnicas de la pantalla LCD 20x4 I2C

Parámetro	Características
Voltaje de entrada	5 VDC; 1,5 mA a 2,5mA
Temperatura operativa	-20 to 70°C
Dimensiones	680 mm x 36 mm x 20 mm
Peso	45,0g
Interfaz	I2C de 4 bits

En este caso, para reducir las conexiones con Arduino, y liberar entradas y salidas para el resto del hardware, se ha utilizado el módulo de conexión I2c (Fig. 20). El citado módulo logra reducir a tan solo 4 conexiones con la placa Arduino. El módulo cuenta además con un potenciómetro para regular el contraste de la pantalla, facilitando así su visualización

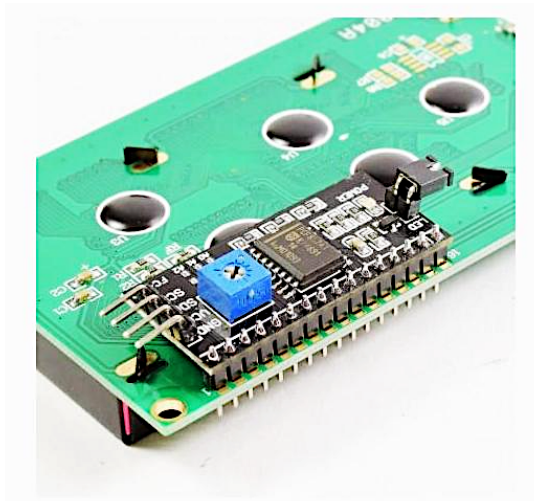


Figura 20. Módulo de conexiones I2C

La Tabla 9 ofrece la interpretación de cada uno de los pines del módulo I2C.

Tabla 9. Descripción de los pines de conexión del módulo I2C

Símbolo	Descripción
GND	Ground - Tierra
VDD	Alimentación +5V DC
SDA	Conexión SDA
SCL	Conexión SCL

3.1.1.5. Alimentación

Si hay acceso a red eléctrica, el sistema puede ser alimentado mediante una fuente de externa de 9V y 1000mA, con un conector de alimentación 5,5x 2,1mm.

En caso de no disponer de toma de corriente eléctrica, el dispositivo puede ser alimentado con dos baterías de 9V con conectores de alimentación 5,5x 2,1mm.

Otras alternativas son también viables, como la instalación de una placa solar de 9V y 3w, junto con una batería recargable.

3.1.1.6. Montaje del prototipo

El montaje de los diferentes componentes que forman el prototipo es sencillo. La pantalla LCD y el sensor JSN R04Tv2 disponen de módulos de conexión que facilitan la tarea. Por otro lado, tanto Arduino UNO Rev3, como el módulo SIM900 disponen de encastrados para cableado puente. En la Figura 21 se presentan los diferentes componentes antes de proceder al montaje en una caja estanca. Tal y como se puede apreciar, el sensor JSN R04Tv2 ha sido ubicado en una de las perforaciones de la caja, lo cual facilita la estabilidad del sensor y las lecturas de distancia.

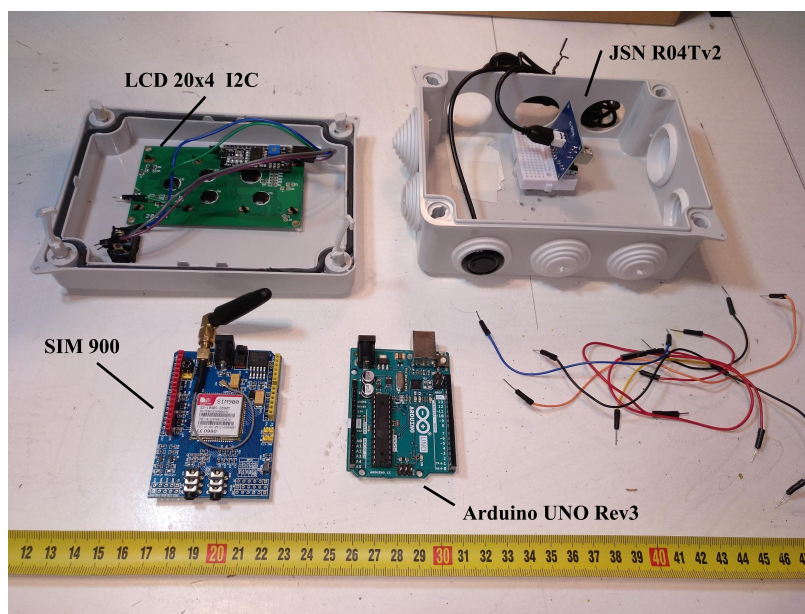


Figura 21. Componentes del prototipo, previa instalación

En la Figura 22, se puede observar el detalle de las conexiones de los diferentes componentes. Como se puede apreciar, se hace uso de una protoboard (placa de pruebas) para facilitar las conexiones.

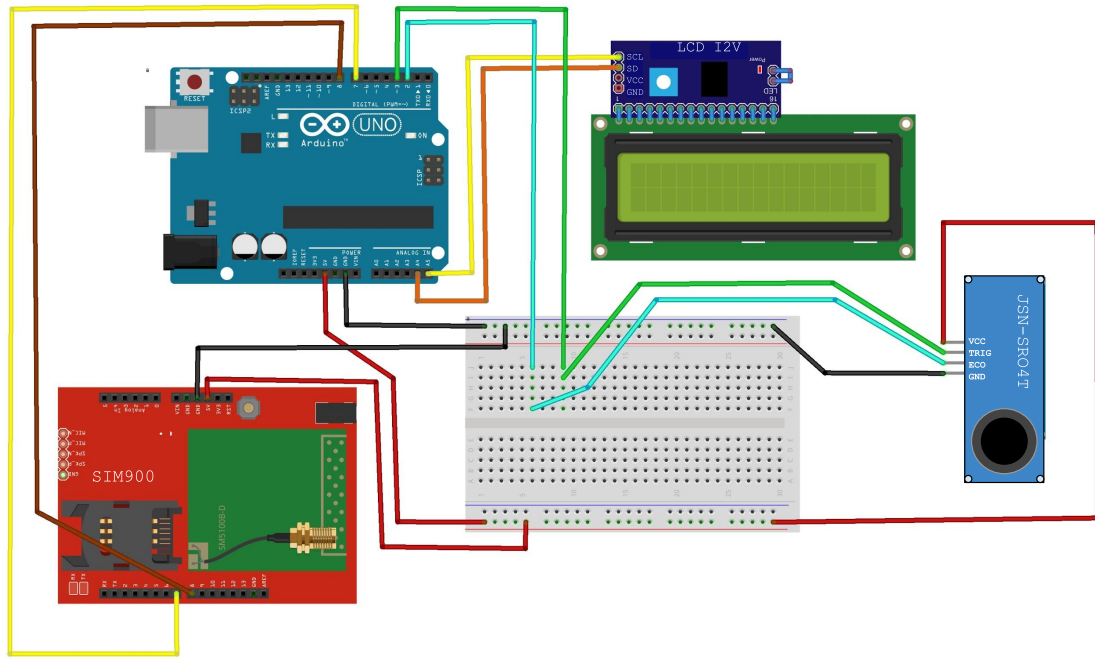


Figura 22. Conexiones de los componentes

En la siguiente Figura 23, se presenta el dispositivo con las conexiones realizadas y empacado en la caja estanca.

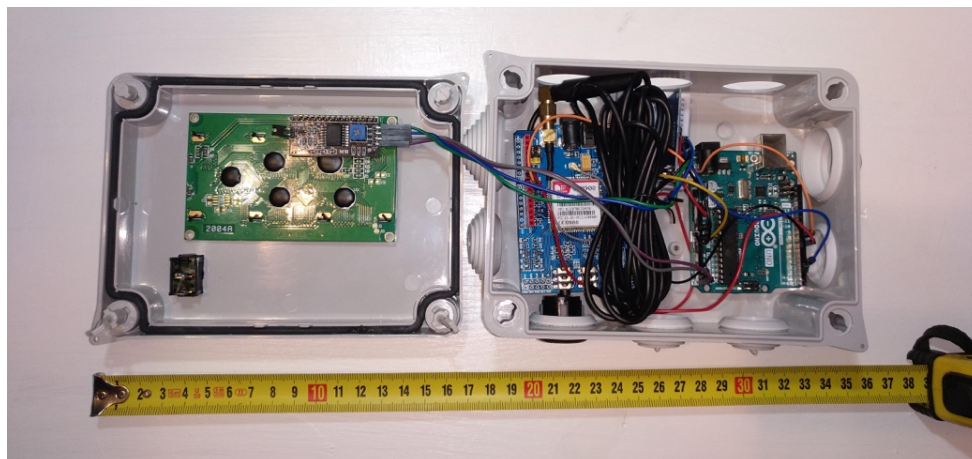


Figura 23. Dispositivo montado

En la siguiente figura (Fig. 24) se puede observar el esquema de montaje del prototipo. Para facilitar su visualización, la misma figura se presenta ampliada en el Anejo II del presente trabajo.

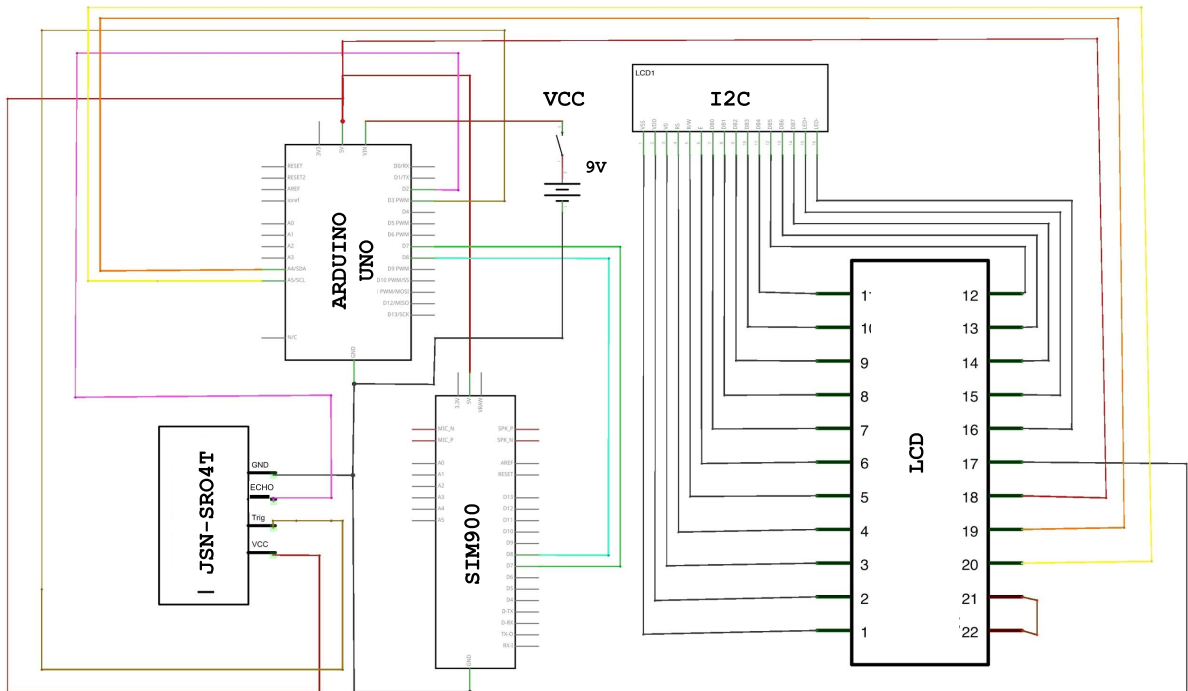


Figura 24. Esquema de montaje

3.1.1.7. Alojamiento del dispositivo

El sistema es empacado en una caja estanca de registro eléctrico de dimensiones 160x115x70mm. En la Figura 25 se muestra el hardware del prototipo empacado en la caja estanca. Sobre la tapa se observa la pantalla LCD y el interruptor de encendido-apagado y en una de las perforaciones laterales de la caja asoma el sensor JSN R04Tv2

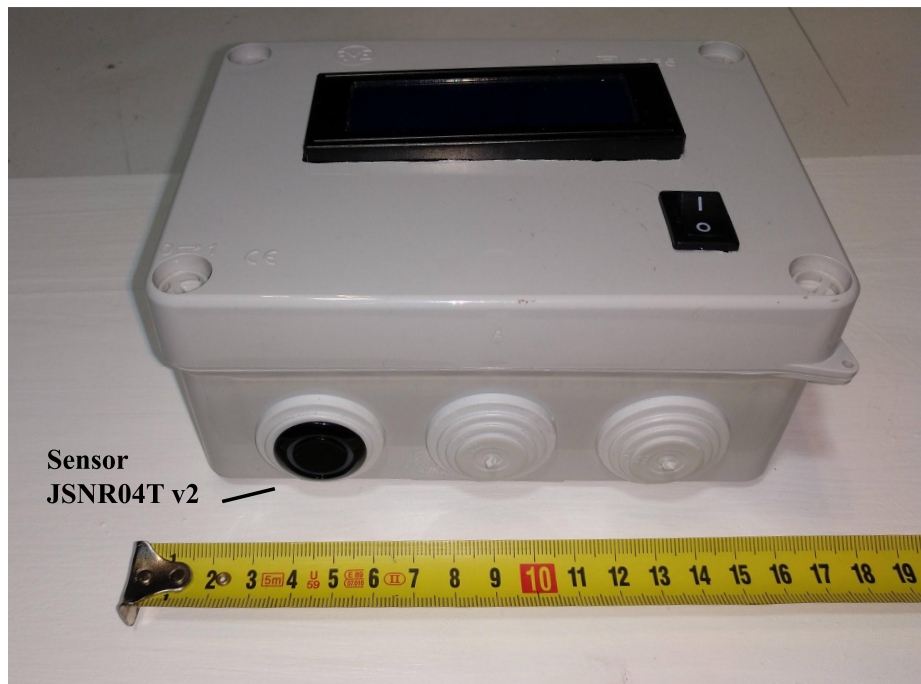


Figura 25. Alojamiento del dispositivo en una caja estanca.

3.1.2. Software

3.1.2.1. Programación del prototipo en Arduino IDE

Para el desarrollo del prototipo se ha utilizado el entorno de programación Arduino IDE. Arduino cuenta con el entorno de programación de código abierto, Arduino IDE (Integrated Development Environment). El código mediante el cual escribimos en nuestra computadora es traducido en Lenguaje C. Posteriormente, el compilador del IDE traduce a código binario el programa, el cual puede ser interpretado por la placa Arduino. En el epígrafe 1.3.2. del presente trabajo se detalla más información sobre el IDE de Arduino. Una vez completado el programa, se compila y carga en la placa Arduino.

a) Instrucciones Setup

Como se puede apreciar en el diagrama de flujo del programa (Fig. 27), el sketch cuenta con una primera secuencia de órdenes conocidas como “Setup” en las que se definen las variables que concurren en el resto del programa y se inician los sensores y módulos que forman el hardware del prototipo.

a.1) Variables

Todas y cada una de las variables que forman parte del programa son importantes para que el prototipo diseñado cumpla con los objetivos propuestos. No obstante, para entender el sketch, conviene explicar la función de las cinco variables principales que se presentan en la siguiente Tabla 10.

Tabla 10. Principales variables del programa

Nombre	Tipo	Función
<i>tempo</i>	long	Duración (longitud del pulso) en microsegundos
<i>hcalculada</i>	float	Distancia del sensor a la lámina de agua
<i>hsensor</i>	float	Distancia desde la base del cauce al sensor
<i>hlamina</i>	float	Altura de la lámina de agua $h_{lamina} = h_{sensor} - h_{calculada}$
<i>hAlerta</i>	float	Altura de la lámina de agua determinada como crítica
<i>hincremento</i>	float	Una vez sobrepasado <i>hAlerta</i> , el sistema reportará cada vez que se incremente el valor indicado

En la Figura 26 se pueden interpretar gráficamente las distintas variables. La altura de la lámina de agua (*h_{lamina}*) se obtiene al restar a la distancia que hay del sensor a la base del cauce (*h_{sensor}*), la distancia que el dispositivo calcula que hay entre el sensor y la superficie de la lámina de agua (*h_{calculada}*). Este cálculo se realiza al contar el tiempo que transcurre entre el envío del pulso de ultrasonidos y la recepción de su eco (*tempo*).

La altura de la lámina de agua considerada como crítica o de riesgo es la variable h_{Alerta} . Esta variable irá incrementando su valor después de cada envío de mensaje de alerta al sumar a h_{Alerta} , el valor de $h_{Incremento}$.

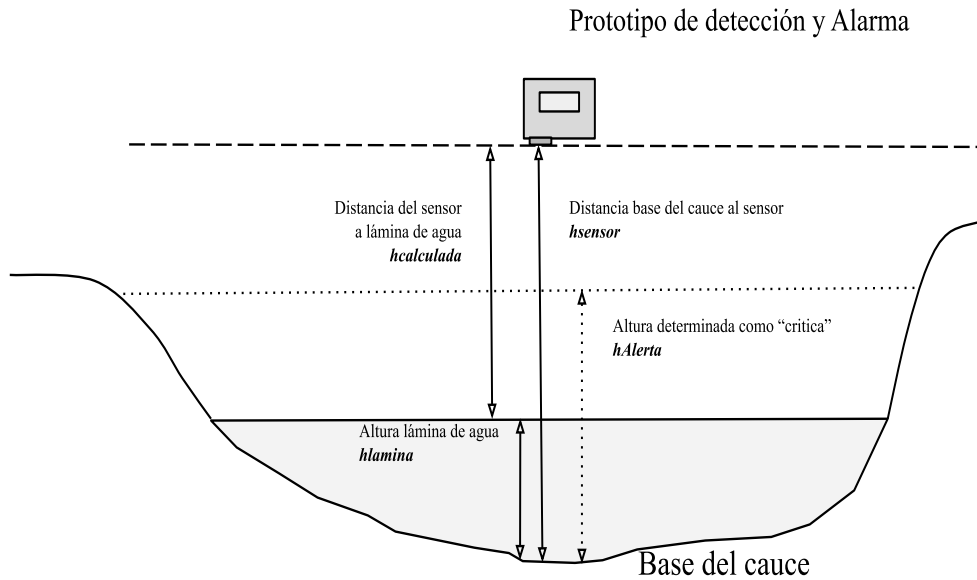


Figura 26. Esquema de interpretación de variables del programa

b) Bucle de instrucciones Loop

b.1) Cálculo de altura de lámina

Una vez ejecutadas las órdenes del Setup, se inicia la secuencia de instrucciones en bucle. El prototipo inicia su secuencia calculando el tiempo que tarda el pulso de ultrasonidos desde que es enviado, hasta que es recibido por el sensor (variable $tempo$). Este valor de tiempo es convertido en distancia mediante la siguiente operación:

$$h_{calculada} = tempo * 0.0003432 / 2$$

El tiempo entre envío y recepción de pulso es dividido entre dos (distancia de ida y eco) y multiplicado por la velocidad del sonido. Se ha considerado una temperatura del aire de 20°C para el cálculo de la velocidad del sonido. Mediante la operación obtenemos

hcalculada en metros. Al restar este valor a la distancia entre el lecho del cauce y el sensor, se obtiene la altura de la lámina de agua, *hlamina*.

b.2.) Condicional y envío de SMS de Alerta

Una vez calculada la altura de la lámina, el programa, mediante una instrucción condicional, interpreta si se debe iniciar la secuencia de envío de SMS de Alerta, o seguir con la secuencia de instrucciones.

Si el valor de la altura de lámina (*hlamina*) es mayor que el nivel de lámina determinado como crítico, el programa iniciará la secuencia de envío de SMS. El envío de mensaje de SMS se realiza a un número de teléfono que tiene que definirse en la orden de ejecución de envío.

b.3.) Envío de datos a la plataforma ThingSpeak

Para el envío de datos a la plataforma ThingSpeak, el módulo SIM 900 debe previamente conectarse a internet. Toda una secuencia de órdenes en código AT permite la conexión a servidor remoto y envía el valor de la altura de lámina de agua a un canal previamente definido de ThingSpeak. Una vez los datos son agregados a la plataforma, estos serán rescatados por una aplicación móvil diseñada en App Inventor y una página Web para facilitar la publicación de los datos. Desde la propia plataforma ThingSpeak se establece un diálogo con la red social Twitter, emitiendo tuits de alarma cada vez que la lámina de agua rebase un nivel declarado como crítico.

Una vez enviados los datos, el dispositivo se desconecta de internet e inicia de nuevo las instrucciones contenidas en el bucle.

En la siguiente Figura 27 se presenta el diagrama de flujo de programación de la placa Arduino UNO Rev 3, así como el conjunto de interacciones configuradas en el ámbito de IoT.

A continuación, se incluyen el conjunto de instrucciones que forman el Sketch de programación de la placa Arduino UNO Rev3, realizado en el entorno Arduino IDE. Para facilitar su interpretación, los códigos van acompañados de las necesarias explicaciones y detalles.

```

/* Universitat Politècnica de València
 * Grado en Ingeniería Agroalimentaria y del Medio Rural
 * Trabajo de Fin de Grado
 * Título: Sistema de detección y alerta de inundaciones mediante
hardware y software con licencia de código abierto
 * Autor: José Moscardó Sáez
 * Tutores: Pablo González Altozano y María Gasque Albalate
 */

/*Se definen salida y entrada del sensor JSN R04Tv2*/
#define trigPin 3 // Pin de salida de ultrasonido
#define echoPin 2 // Pin de recepción de eco

/* Se definen las variables*/
long tiempo;
float hcalculada; // Altura directamente calculada. Resultado de
interpretar el tiempo desde que se envía el pulso hasta su recepción
float hsensor = 1.66; //Altura desde la base del cauce al sensor. En
el caso que nos ocupa: 1.66 m
float hlamina =0; // Altura de la lámina de agua desde la base del
cauce hasta la superficie
float hlaminaAnt=0;
float hAlerta=0.50; // Altura determinada como crítica o de Alerta.
En este caso 0.50m
float hincremento=0.20; /* Una vez sobrepasada la altura crítica
(hAlerta) el sistema reportará cada vez que se incremente en el valor
indicado
en este caso se han considerado 0.20m */
String mensaje = "";

/* Pantalla LCD, con módulo de conexiones I2C*/
#include <Wire.h> // Se incluye la librería Wire para conexiones I2c
#include <LiquidCrystal_I2C.h> // Se incluye la librería para la
configuración de pantallas LCD con módulo I2C
LiquidCrystal_I2C lcd(0x27,20,4);

/* Modulo GSM/GPRS SIM900*/
#include <SoftwareSerial.h> // Incluye la librería SoftwareSerial
necesaria para programar el módulo
SoftwareSerial gprsSerial(7,8);//Configuración de los pines de
transmisión (7) y recepción (8) de datos del módulo GSM/GPRS

void setup() { /* Comienza secuencia de ejecución del Setup*/

  lcd.init(); // Inicia la pantalla LCD
  lcd.backlight(); // Iluminación del LCD
  lcd.begin(20,4); // Se define el LCD de 20x04

```

```

Serial.begin(9600); // Establece la velocidad bits por segundo
(baudios) para la transmisión de datos en serie.

/* Mensaje de inicio*/
lcd.setCursor(0,0); //Sitúa el cursor del LCD
lcd.print("J.Moscardo TFM"); // Mensaje en LCD
lcd.setCursor(0,1); /
lcd.print("Alarma de Crecidas");
lcd.setCursor(0,2);
lcd.print("Iniciando GPRS");

/* Inicia módulo GSM/GPRS y establece conexión*/
gprsSerial.begin(9600); // Establece la velocidad bits por segundo
(baudios) para la transmisión del GSM/GPRS
gprsSerial.println("AT"); //Inicia las comunicaciones mediante
comandos AT
delay(1000); //Tiempo de espera
gprsSerial.println("AT+CPIN=\"XXXX\""); // Se introduce la clave PIN
de la tarjeta SIM. Substituir XXXX por la clave.
delay(27000); // Tiempo de espera para asegurar conexión con el
servicio contratado GSM/GPRS
gprsSerial.println("AT+CREG?"); // Comando AT para solicitar
confirmación de conexión
delay(1000); //Tiempo de espera
gprsSerial.println("AT+CGATT?"); //Comando AT para solicitar
confirmación de que el dispositivo se ha anexo a la conexión
establecida
delay(1000); //Tiempo de espera
gprsSerial.println("AT+CIPSHUT"); // Para que la conexión sea
estable, se desconecta el GPRS
delay(1000); //Tiempo de espera

/* Mensaje informativo en LCD*/
lcd.setCursor(0,2);
lcd.print(" "); // Limpia la línea 2 del LCD
lcd.setCursor(0,2);
lcd.print("Iniciando Sensor");

/* Se inician lecturas del módulo JSN R04Tv2*/
pinMode(trigPin, OUTPUT); // Se define trigPin (3) como Salida
pinMode(echoPin, INPUT); //Se define echoPin (2) como Entrada
delay(1000);

lcd.clear(); //Limpia el contenido del LCD

}/* Termina la secuencia Setup*/

void loop() { /* Inicia secuencia Loop que se repetirá indefinidamente*/

/*Envío y recepción de ultrasonidos y cálculo de distancia*/
digitalWrite(trigPin, LOW); // Para inicial el loop, desconecta el
pin de salida de pulso
delayMicroseconds(5); // Tiempo de espera de 5 microsegundos
digitalWrite(trigPin, HIGH); // Conecta el pin de salida de pulso
delayMicroseconds(10); // Tiempo de espera de 10 microsegundos. Esta
secuencia es la recomendada para lecturas estables
digitalWrite(trigPin, LOW); // Desconecta el pin de salida de pulso

```

```

tempo = pulseIn(echoPin, HIGH); //Lee el echoPin y devuelve la
duración (longitud del pulso) en microsegundos
// Calcula la distancia del sensor a la lámina de agua en m.  $d=v*t/2$ 
hcalculada = tempo*0.0003434/2; // Divide entre 2. Tiempo de ida del
pulso y tiempo de retorno del eco
//Se obtiene la velocidad a partir de  $V=331.4 + 0.6T$  (T en °C)
//Se ha tomado como valor para la temperatura 20°C
hlamina = hsensor-hcalculada; // La altura de la lámina de agua se
obtiene restando la distancia

/*Imprime en Monitor y pantalla LCD el resultado de las lecturas */
Serial.print("Dist. Calculada = ");
Serial.print(hcalculada); //Imprime variable
Serial.println(" m");
Serial.print("Altura Lamina = ");
Serial.print(hlamina);
Serial.println(" m");

lcd.setCursor(0,0);
lcd.print("Distancia m:"); lcd.print(hcalculada);
lcd.setCursor(0,1);
lcd.print("Alt. Lamina m:"); lcd.print(hlamina);

/*Secuencia condicional para iniciar la alerta si la lámina de agua
sobrepasa el nivel crítico (hAlerta) */
if(hlamina >hAlerta&& hlaminaAnt <=hAlerta){ //Núcleo de la
instruccion condicional
    delay(2000); //Tiempo de espera
    hAlerta=hAlerta+hincremento; //Incrementa el siguiente nivel de
alerta con el valor de incremento hincremento

    //Conjunto de informaciones en pantalla LCD y Monitor
    Serial.print("Envío de SMS de Alerta");
    Serial.println("Envío de SMS de Alerta");
    lcd.setCursor(0,2);
    lcd.print("Nivel Alerta");
    Serial.print("Iniciando envío");
    lcd.setCursor(0,3);
    lcd.print("Envío SMS Alerta");

    //Inicia secuencia de envío de SMS de Alerta
    gprsSerial.print("AT+CMGF=1\r"); // comando AT para configurar
el SIM900 en modo texto
    delay(200);
    gprsSerial.println("AT + CMGS = \"123456789\"); //Reemplazar
123456789 por el número de teléfono al que enviar el mensaje
    delay(200);
    String dataMessage = ("Se ha detectado agua Nivel m:"+
String(hlamina)); // Mensaje que se envía. Incluye la variable hlamina
    gprsSerial.print(dataMessage); // Envía el SMS
    gprsSerial.println((char)26); //Finaliza el envío del mensaje
con el comando de código Ascii 26, carácter (→)
    delay(1000);
    gprsSerial.println();
    delay(6000); //Tiempo de espera para asegurar que se ha enviado
el SMS
    hlaminaAnt=round(hlamina); //igual a hlaminaAnt con un valor
redondeado de hlamina

    //Informaciones en pantalla LCD y Monitor

```



```

Serial.print("Enviado SMS");
lcd.setCursor(0,3);
lcd.print("SMS Enviado");
Serial.print("SMS Enviado");
delay(500);
lcd.setCursor(0,2);
lcd.print("                ");
lcd.setCursor(0,3);
lcd.print("                ");

}/*Finaliza Secuencia Condicional*/

hlaminaAnt=hlamina; // Iguala hlaminaAnt con lectura hlamina
delay(1000);//Tiempo de espera

/*Inicio secuencia de conexión a internet para envío de lecturas*/
if (gprsSerial.available())
Serial.write(gprsSerial.read());

gprsSerial.println("AT+CIPSTATUS"); // Consulta estado conexión
Serial.print("Iniciando Envío datos ThingSpeak");
delay(2000);

gprsSerial.println("AT+CIPMUX=0");//Configura el dispositivo para
una conexión IP (0=unica)
delay(2000);

ShowSerialData();//Sub-secuencia para visualizar datos en GPRS

gprsSerial.println("AT+CSTT=\"airtelwap.es\"");//Conexión del
módulo con el punto de acceso APN
delay(1000);

ShowSerialData();//Código para visualizar los datos del módulo
GPRS

gprsSerial.println("AT+CIICR");//Abre conexión inalámbrica
delay(3000);

ShowSerialData();

gprsSerial.println("AT+CIFSR");//Obtiene dirección IP
delay(2000);

ShowSerialData();

gprsSerial.println("AT+CIPSPRT=0"); //Establece un prompt (a la
espera de comandos)
delay(3000);

ShowSerialData();

gprsSerial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",
\"80\");//Empieza la conexión para enviar datos a ThingSpeak
delay(6000);

ShowSerialData();

gprsSerial.println("AT+CIPSEND");//Envía datos a servidor remoto
delay(4000);

```

```

    ShowSerialData();

    //En esta secuencia envía el resultado de la altura de la lámina
de agua (hlamina) al canal de ThingSpeak abierto para ser recopilado
    //La primera orden define el contenido del mensaje que será enviado
    String str="GET
https://api.thingspeak.com/update?api_key=CODIGOAPIDELCANAL&field1="
+ String(hlamina); // Substituir por el código API del canal
    Serial.println(str);// El mensaje se presenta en el monitor
    gprsSerial.println(str);//El mensaje es enviado al servidor
    delay(4000);

    ShowSerialData();

    gprsSerial.println((char)26);//Mensaje enviado. Finaliza con el
comando de código Ascii 26, carácter (→)
    delay(5000);//Tiempo de espera, importante para asegurar que se ha
enviado el mensaje
    Serial.print("Datos enviados a ThingSpeak");// En el monitor se
informa del envío del mensaje
    gprsSerial.println();

    ShowSerialData();

    gprsSerial.println("AT+CIPSHUT");//Solicita cortar comunicación
    delay(100);//

    ShowSerialData();

    Serial.print("Cortando Conexión"); // Se informa por monitor que
se ha cortado la conexión

    delay(100);
} /* Termina la secuencia Setup*/

void ShowSerialData() /* Sub-secuencia ShowSerialData*/
{
    while(gprsSerial.available() !=0)
        Serial.write(gprsSerial.read());
    delay(5000);
}/* Termina subsecuencia ShowSerialData*/

/* Fin del programa*/

```

3.1.2.2. ThinkSpeak

Lanzado en 2010, ThingSpeak es un servicio de código abierto (open source) de plataforma de análisis de IoT (Internet Of Things) que permite agregar, visualizar y analizar flujos de datos en vivo en la nube mediante el uso de código HTTP (ThinkSpeak, 2020). Los datos agregados a ThingSpeak pueden ser analizados mediante MATLAB®. ThingSpeak permite crear prototipos y construir sistemas IoT sin la necesidad de configurar servidores o desarrollar software web para la visualización de datos en la nube.

En la siguiente Figura 28, se puede ver el canal de ThingSpeak creado para el presente proyecto, con los visores de “Evolución de Nivel de Lámina de Agua” en la parte izquierda, y el “Valor del último dato agregado” en su parte derecha.

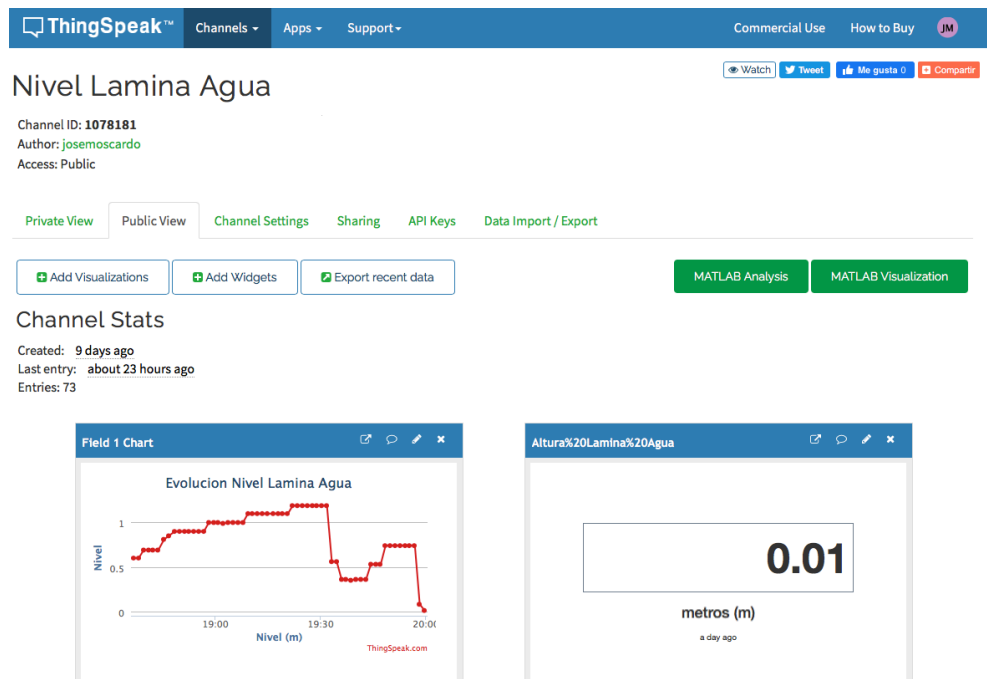


Figura 28. Canal de ThingSpeak del presente proyecto

El prototipo diseñado se comunica con el canal de ThingsSpeak “Nivel de Lámina de Agua” mediante los códigos de las API Keys (Application Programming Interface). En la siguiente figura (Fig. 29) se presentan las API Keys del proyecto.

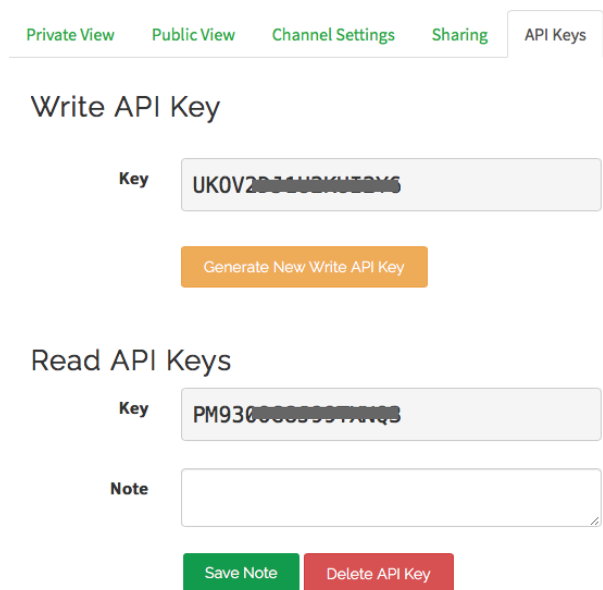


Figura 29. API Keys del canal de ThingSpeak

Mediante estos códigos API, el prototipo se enlaza con la plataforma de IoT ThingSpeak. A continuación, se detalla el bloque de instrucciones donde se vincula el prototipo con la plataforma ThingSpeak:

```
String str="GET
https://api.thingspeak.com/update?api_key=CODIGOAPIDELCANAL&field1="
+ String(hlamina); // Substituir por el código API
Serial.println(str); // El mensaje se presenta en el Monitor
gprsSerial.println(str); // El mensaje es enviado al servidor
delay(4000);
```

Como se puede apreciar, Arduino ordena al módulo SIM900 a conectarse a la plataforma de ThingSpeak y mediante el API Key publicar los datos en el canal.

3.1.2.3. Aplicación móvil con MIT App Inventor

MIT App Inventor, es una aplicación de código abierto (open source) de programación visual. Se fundamenta en el sistema de edición mediante bloques, y permite crear aplicaciones para smartphones. Fue desarrollado en los laboratorios Google Labs por el equipo de investigación del Massachusetts Institute of Technology (MIT) de Hal Abelson (MIT App Inventor, 2019). Actualmente, MIT App Inventor (Fig.30) es sostenida y actualizada por el MIT.



Figura 30. Logo de MIT App Inventor

Fuente MIT App Inventor, 2020

MIT App Inventor es utilizado por estudiantes, docentes, desarrolladores, aficionados y empresarios para desarrollar aplicaciones para la cooperación, la productividad, el uso personal, la enseñanza, el desarrollo social y el ocio (Crawford *et al.*, 2013)

La programación de aplicaciones con MIT App Inventor se realiza de manera visual, arrastrando bloques de códigos.

En la siguiente Figura 31, se presenta la interfaz de programación de MIT App Inventor y el desarrollo de la pantalla de inicio de la aplicación creada en el presente proyecto.

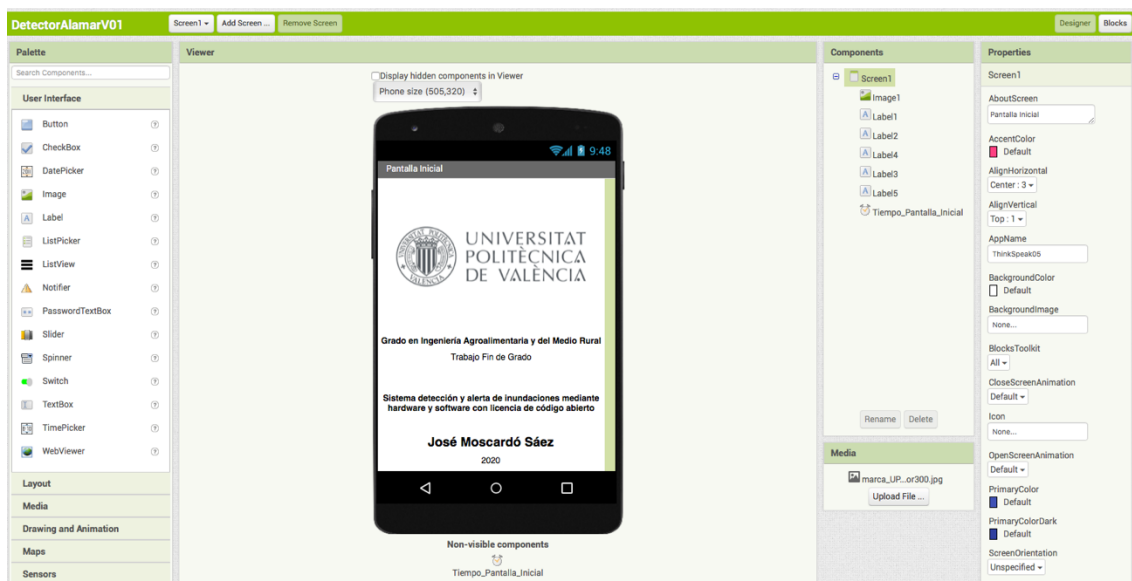


Figura 31. Diseño de la pantalla de inicio de la App con MIT App Inventor

La programación de aplicaciones con MIT App Inventor se realiza de manera visual, agrupando bloques de códigos que van configurando el conjunto de instrucciones que dan cuerpo al programa. En la siguiente figura (Fig. 32) se muestran los bloques de instrucciones de la App que permiten rescatar los datos agregados al canal de ThingSpeak.

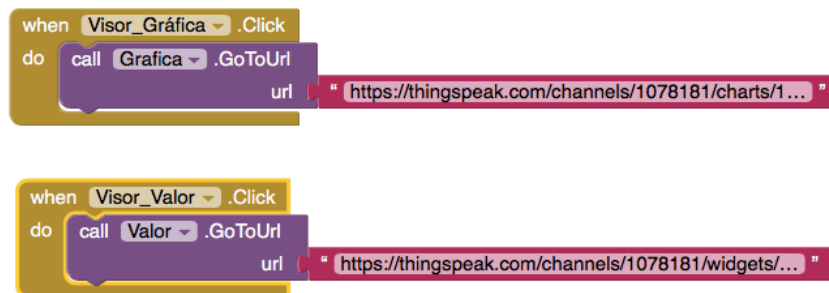


Figura 32. Bloques de instrucciones de MIT App Inventor

3.1.2.4. Publicación en página Web

Para facilitar el acceso de la población a los datos, se ha creado una página Web en la que se actualizan los datos recogidos por el prototipo y se presenta un histórico de los niveles de la lámina de agua.

La Web está realizada mediante la aplicación online Google Sites. La Web se alimenta automáticamente con los datos que el prototipo agrega a ThingSpeak.

La dirección de la Web es la siguiente:

<https://sites.google.com/view/sistemadealertadecrecidas/inicio>

En la Figura 33, se muestra la pantalla principal de la Web creada en el presente proyecto. La Web se alimentará con información de servicio de utilidad para la gestión de emergencias.



Figura 33. Página Web en la que se publican los datos

3.1.2.5. Publicación Redes Sociales

La plataforma IoT ThingSpeak permite vincular los canales publicados con la red social Twitter mediante ThingTwitter. Esta App incorporada en ThingSpeak permite programar “reacciones” (Reacts) que se vinculan con una cuenta de Twitter.

Se han programado dos reacciones de tipo condicional como la que se presenta en la siguiente figura (Fig.34). En este caso, la Reacción enviaría un Tuit siempre que el nivel de la lámina de agua supere 0.50 m

Apps / React / Envio de Alerta 0.50

Edit React

Name:	Envio de Alerta 0.50
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2020-06-17 17:44
Channel:	Nivel Lamina Agua
Condition:	Field 1 (Nivel) is greater than 0.50
ThingTweet:	pepemoscardo: Detectado Nivel Lámina de Agua 0.5m
Run:	Only the first time the condition is met
Created:	2020-06-17 5:42 pm

Figura 34. Programación de reacciones con ThingTwitter de ThingSpeak

3.1.5. Coste material del prototipo

Se detalla en la Tabla 11 el coste material del prototipo. Los precios recogidos en el presupuesto hacen referencia a compras en pequeñas cantidades. En grandes volúmenes de compra, este precio se vería considerablemente reducido. No se han considerado costes de desarrollo.

Tabla 11. Coste material del prototipo

	Descripción	Cantidad	Precio	Importe
1	Placa Arduino UNO Rev3	1	20,00	20,00
2	Sensor JSN R04T v2	1	4,70	4,70
3	LCD 20x4 I2C	1	6,55	6,55
4	Módulo SIM900	1	12,10	12,10
5	Cargador 9V-DC , 1000mA	1	5,15	5,15
6	Cableado de puente para protoboard	1	0,20	0,20
7	Cable alimentación con conector 5,5x 2,1mm	1	0,45	0,45
8	Caja estanca de registro eléctrico 11x11x6cm	1	2,55	2,55
Presupuesto sin IVA				51,70€
IVA (21%)				10,85€
Coste material del prototipo				62,55 €

3.2. ENSAYOS DE FUNCIONAMIENTO Y PRECISIÓN

3.2.1. Ensayo previo de precisión

Para comprobar el funcionamiento y la precisión de cálculo de distancia del prototipo se realizó una batería de mediciones, comparando el resultado obtenido mediante el dispositivo, con la distancia determinada con un flexómetro.

Se situó el dispositivo sobre una superficie horizontal y estable. A una distancia de 2,15 m se posiciona un obstáculo compuesto por una tabla perpendicular a la mesa sobre la que se realiza el ensayo. La posición de tabla se fue desplazando en dirección hacia el sensor a intervalos de 0,05 m anotando cada vez el resultado arrojado por el dispositivo. El ensayo se repitió tres veces.

En la siguiente figura (Fig.35) se presenta un croquis explicativo del ensayo.

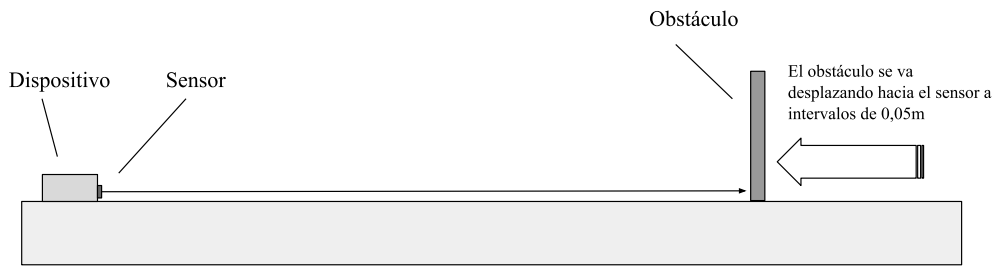


Figura 35. Croquis explicativo del ensayo de precisión

En la siguiente Figura 36 se muestra una imagen representativa del ensayo previo de precisión.

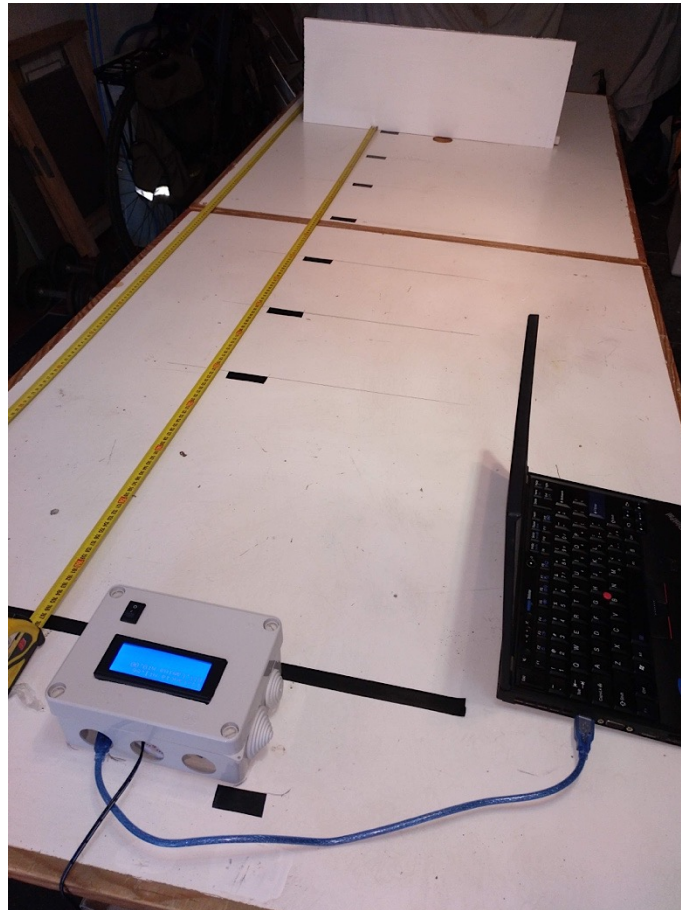


Figura 36. Ensayo de precisión

En la Tabla 12 se ofrecen los datos resultantes del ensayo. Se presenta el valor medio de los tres ensayos, junto con la desviación típica.

La interpretación de los datos se realiza en el apartado 4 del presente trabajo, Resultados y comprobación del funcionamiento.

Tabla 12. Resultados del ensayo de precisión del dispositivo

Distancia del sensor al objeto (m)	Valor Medio de las mediciones (m)	Desviación típica
0,3000	0,2976	0,00778
0,3500	0,3499	0,00346
0,4000	0,4019	0,00757
0,4500	0,4499	0,00453
0,5000	0,4990	0,00120
0,5500	0,5469	0,00184
0,6000	0,6015	0,00629
0,6500	0,6544	0,00318
0,7000	0,7066	0,00156
0,7500	0,7531	0,00042
0,8000	0,8068	0,00410
0,8500	0,8545	0,00269
0,9000	0,9088	0,00007
0,9500	0,9514	0,00219
1,0000	1,0065	0,00163
1,0500	1,0503	0,00332
1,1000	1,0977	0,00057
1,1500	1,1506	0,01047
1,2000	1,2029	0,00445
1,2500	1,2559	0,00226
1,3000	1,3048	0,00191
1,3500	1,3513	0,00658
1,4000	1,4024	0,00537
1,4500	1,4550	0,00290
1,5000	1,5034	0,00120
1,5500	1,5567	0,00021
1,6000	1,6047	0,00339
1,6500	1,6513	0,00290
1,7000	1,6995	0,01018
1,7500	1,7505	0,00382
1,8000	1,8009	0,00389
1,8500	1,8534	0,00884
1,9000	1,9073	0,00269
1,9500	1,9524	0,00742
2,0000	2,0003	0,00332
2,0500	2,0535	0,00728
2,1000	2,1075	0,00290
2,1500	2,1501	0,00283

3.2.2. Comportamiento del dispositivo en mediciones de altura de lámina de agua

Para verificar que el sensor de ultrasonidos detecta igualmente superficies sólidas como líquidas se realizó un ensayo de precisión similar al descrito en el anterior apartado. Se dispuso el prototipo verticalmente, dirigiendo el sensor hacia un recipiente al que progresivamente se le añadiría agua, a la vez que se realizaban mediciones con el sensor. El ensayo se repitió por tres veces. En la Figura 33 se presenta una imagen del desarrollo de la prueba.

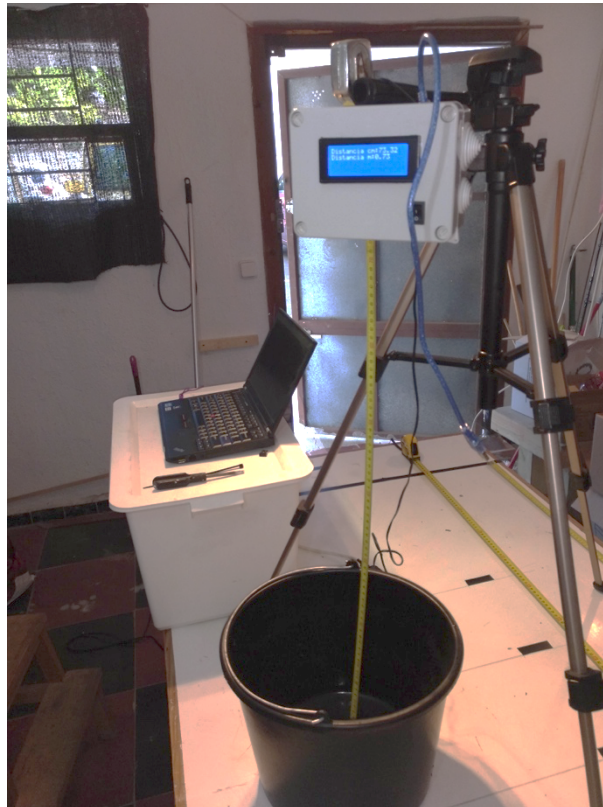


Figura 37. Comprobaciones de funcionamiento con lámina de agua

En la Tabla 13 se ofrecen los resultados del ensayo. Se presenta el valor medio de los tres ensayos, junto con la desviación típica. El dispositivo detecta con similar precisión tanto superficies sólidas como líquidas.

Tabla 13. Resultados del ensayo de precisión de mediciones de altura de lámina de agua

Distancia del sensor a la lámina de agua (m)	Valor Medio de las mediciones (m)	Desviación típica
0,8000	0,7987	0,0037
0,7900	0,7928	0,0046
0,7800	0,7829	0,0030
0,7700	0,7645	0,0021
0,7600	0,7570	0,0014
0,7500	0,7535	0,0049
0,7400	0,7400	0,0042
0,7300	0,7325	0,0007
0,7200	0,7255	0,0007
0,7100	0,7095	0,0078
0,7000	0,7030	0,0014

3.3. SIMULACIÓN DE MONITOREO Y ALERTA DE INUNDACIÓN EN UN TORRENTE

3.3.1. Descripción de la simulación realizada

Con el objeto de evaluar la respuesta del dispositivo para monitorear la evolución de la lámina de agua en un cauce, detectar posibles crecidas de agua y operar como sistema de alerta, se realizó una simulación en el torrente Sarrià a su paso por el Parc Bit, Palma de Mallorca, Illes Balears. Para la simulación se ancló el dispositivo en una estructura rígida, lo cual asegura lecturas estables.

Una vez anclado el dispositivo, se realiza la medición de la distancia del dispositivo al lecho del cauce. Este valor es fundamental para poder realizar las estimaciones de altura de la lámina de agua. Las mediciones se realizaron con el mismo dispositivo y fueron comprobadas con un flexómetro. Para esta medición de altura con el dispositivo, se utilizó una versión simplificada del programa presentado en el epígrafe 3.2.1. del presente trabajo. Se eliminaron las instrucciones de conexiones GSM/GPRS, de esta forma el tiempo entre lecturas es más rápido, lo que facilita la operativa de realización de mediciones.

Una vez obtenido este valor, es incorporado al programa objeto de análisis junto con el valor de las variables *hAlerta* y *hincremento*, las cuales determinarían el envío de SMS del sistema de alerta. De igual modo se programan las reacciones de la app ThingTwitter para que el envío de alertas en la red social vaya en consonancia con las alertas enviadas en SMS.

Para realizar la simulación de detección de la lámina de agua y de la publicación de datos en diferentes plataformas, así como confirmar el envío y publicación de los diferentes mensajes de alerta, se realizaron simulaciones en una superficie horizontal, en la cual se fijaba el prototipo y al que progresivamente se le acercaría un objeto que sería detectado por el sensor.

En los siguientes apartados del presente epígrafe se exponen con detalle las diferentes etapas de la simulación.

3.2.2. Ubicación del dispositivo

El dispositivo se ubica el 17 de junio a las 17.30 en el Torrent Sarrià en las coordenadas 39.637694, 2.631905. En la Figura 38 se representa la ubicación de la simulación.

El torrente de Sarrià nace en el municipio de Esporles y realiza su curso hacia Palma de Mallorca pasando por la antigua posesión Sarrià, de la cual toma su nombre. Al adentrarse en la ciudad de Palma de Mallorca, el torrente adquiere el nombre de na Bàrbara.

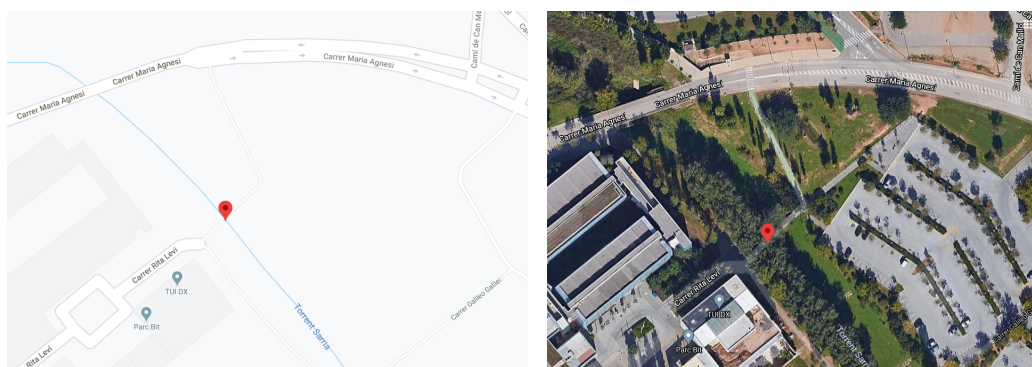


Figura 38. Ubicación del dispositivo para realizar la simulación

Se elige esta ubicación por su fácil acceso, y por la escasa altura entre el torrente y una pasarela que lo cruza, lo cual permite fijarlo a la estructura y poder realizar las necesarias manipulaciones. En la siguiente figura (Fig. 39) se puede apreciar la pasarela que cruza el torrente, en la cual se ubicará el dispositivo.



Figura 39. Torrent Sarrià, a su paso por el Parc Bit, Palma. Illes Balears.

El dispositivo se ancla (Fig. 40) a la estructura mediante bridas, lo cual asegura su fijación y estabilidad, a la vez que es fácilmente desmontable.



Figura 40. Anclaje del dispositivo

En la siguiente Figura 41, se puede apreciar el dispositivo anclado a la estructura sobre el torrente de Sarrià.



Figura 41. Dispositivo anclado sobre el torrente Sarrià

3.2.3. Determinación de la distancia del sensor al lecho del cauce e incorporación de datos al programa

Para poder calcular la altura de la lámina de agua que circulará en el cauce es necesario determinar la altura del sensor al lecho del mismo cauce. En la simulación se realizaron medidas directas con un flexómetro (Fig. 42) y mediciones con el propio prototipo.



Figura 42. Mediciones del lecho del cauce del torrente

Al no ser un cauce de base regular y presentar obstáculos como piedras, hoja y ramillas, se decidió asignar una precisión de centímetros a las mediciones. Una vez anclado y estable el dispositivo, con el sensor apuntando perpendicularmente a la base del torrente, se registró, tanto con el dispositivo como con flexómetro, un valor de 166 cm de distancia desde el sensor hasta la base del torrente.

Este importante dato se agregó al programa que controla el dispositivo, como valor de la variable *hsensor*. En el siguiente extracto del sketch, se ha resaltado la variable *hsensor* junto con el valor asignado.

```

long tiempo;
float hcalculada; // Altura directamente calculada. Resultado de
interpretar el tiempo desde que se envía el pulso hasta su recepción
float hsensor = 1.66; //Altura desde la base del cauce al sensor. En
el caso que nos ocupa: 1.66 m
float hlamina =0; // Altura de la lámina de agua desde la base del
cauce hasta la superficie
float hlaminaAnt=0;
float hAlerta=0.50; // Altura determinada como crítica o de Alerta.
En este caso 0.50m
float hincremento=0.20; /* Una vez sobrepasada la altura crítica
(hAlerta) el sistema reportará cada vez que se incremente en el
valor indicado en este caso se han considerado 0.20m */

```

3.2.4. Determinación del nivel de alerta e incorporación de datos al programa

Sin disponer de un estudio topográfico e hidrológico del torrente, y al ser el ensayo demostrativo de la funcionalidad del prototipo diseñado, se optó por determinar un nivel de lámina de agua del torrente cuyo valor debería considerarse de riesgo. Se inspeccionaron ambos márgenes, observando y midiendo la altura de la ribera que contendría un hipotético episodio de crecida del torrente. Se determinó que una altura de 0,5m debería considerarse de riesgo, lo cual justificaría emitir una alerta. Igualmente se decidió que las sucesivas alertas deberían publicarse al incrementarse en 0,20 cm.

Ambos valores se incorporaron al programa de control de la placa Arduino. Estos valores definirán la actuación del dispositivo a la hora de enviar los SMS de alerta. Se muestra a continuación la definición de las variables en el sketch:

```

long tiempo;
float hcalculada; // Altura directamente calculada. Resultado de
interpretar el tiempo desde que se envía el pulso hasta su recepción
float hsensor = 1.66; //Altura desde la base del cauce al sensor. En
el caso que nos ocupa: 1.66 m
float hlamina =0; // Altura de la lámina de agua desde la base del
cauce hasta la superficie
float hlaminaAnt=0;
float hAlerta=0.50; // Altura determinada como crítica o de Alerta.
En este caso 0.50m
float hincremento=0.20; /* Una vez sobrepasada la altura crítica
(hAlerta) el sistema reportará cada vez que se incremente en el
valor indicado en este caso se han considerado 0.20m */

```

Para que los mensajes de emergencia vía Twitter actúen paralelamente con el envío de SMS, se programaron las reacciones de la App ThingTwitter para que los mensajes sean

publicados al rebasar una altura de lámina de 0.5m y de 0.70m. En el apartado 3.1.2.1. del presente trabajo se detalla cómo se realizó la programación de envío de los mensajes.

3.2.5. Simulación de crecida del torrente con parámetros controlados

Tal y como se ha expuesto en la introducción del presente epígrafe 3.3, la simulación de una crecida de agua en el torrente se realizó en un entorno controlado en el cual la toma de datos y evaluación de la altura de lámina de agua fuera sencilla y fiable. Siguiendo la metodología del ensayo de precisión expuesto en el apartado 3.2.1. del presente trabajo, se ubicó el dispositivo sobre una superficie horizontal y estable. A una distancia de 1,66 m se situó un obstáculo compuesto por una tabla perpendicular a la mesa sobre la que se realiza el ensayo. La distancia de 1,66 m simula la base del torrente, y el obstáculo (una tabla perpendicular a la superficie sobre la que se realiza la simulación) simula la lámina de agua. En la Figura 43 se presenta un croquis explicativo de la simulación realizada.

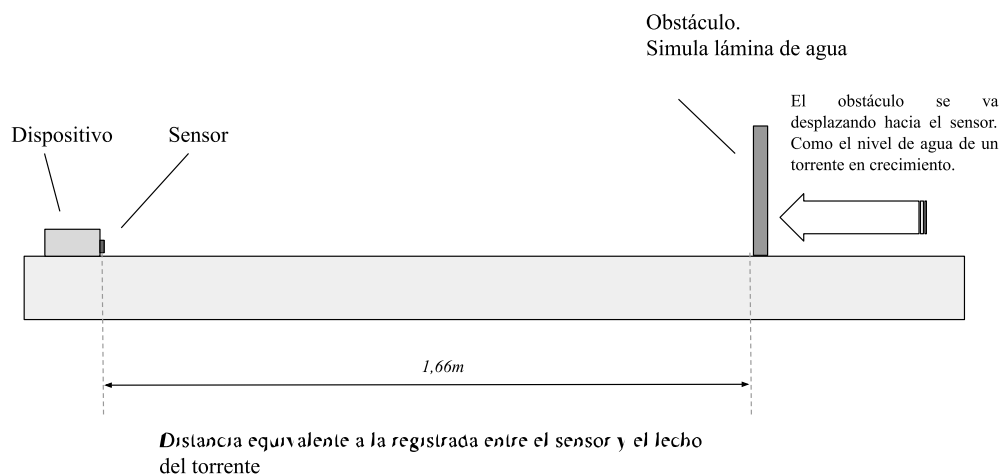


Figura 43. Croquis explicativo de la simulación

Se realizaron dos tipos de simulaciones:

- Simulación de una crecida progresiva, pero sin establecer un patrón de intervalo de crecimiento del nivel de agua.
- Simulación de una crecida en la que el nivel de agua se incrementa a intervalos de 0,20 m: 0,0 m ; 0,20 m ; 0,40 m ; 0,60 m; 0,80 m; 1,00m.

En la primera simulación se podrán observar el resultado de publicación de los datos en las diferentes plataformas (ThingSpeak, Web, App Android), así como el envío de SMS de alertas y publicación de mensajes de alarma en la red social Twitter.

La segunda simulación está confeccionada para comprobar la exactitud de las distancias registradas y los valores publicados.

4. RESULTADOS Y COMPROBACIÓN DEL FUNCIONAMIENTO

4.1. PRECISIÓN DEL DISPOSITIVO

En el ensayo de precisión se detectaron variaciones entre la distancia registrada con flexómetro y el valor calculado por el dispositivo. Atendiendo a los resultados (Tabla 12), pueden estimarse márgenes de error de ± 1 cm, lo cual entra dentro del rango de precisión del sensor (Jahankit, 2020), y son aceptables para el propósito del dispositivo diseñado.

En la siguiente Figura 44, se muestra la relación entre las medidas reales y el promedio de las medidas realizadas con el dispositivo. Se presenta también la desviación típica obtenida en el ensayo de precisión.

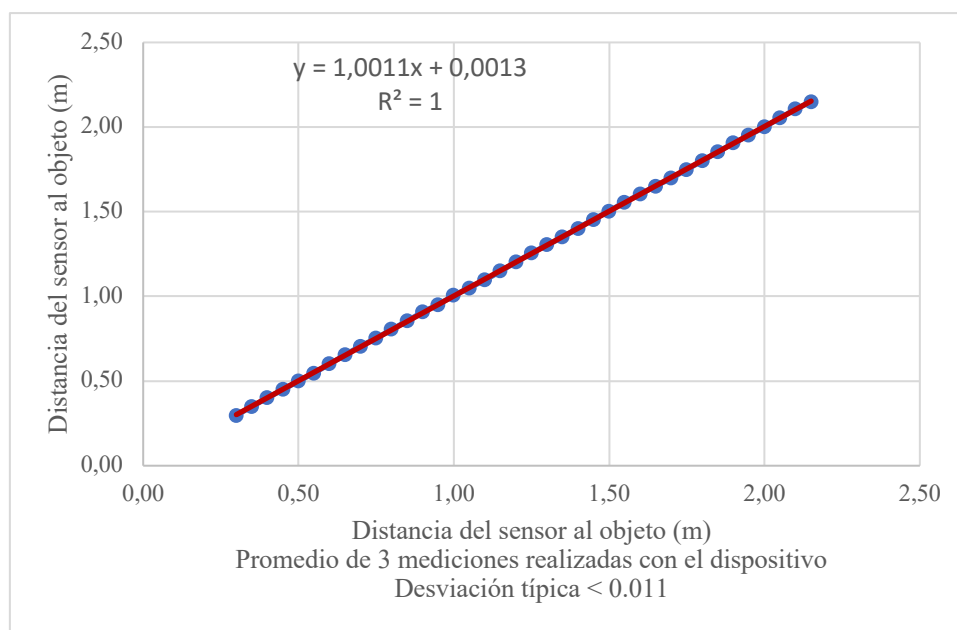


Figura 44. Resultados del ensayo de precisión

En lo que respecta a los ensayos realizados con lámina de agua. El dispositivo se comporta con similar precisión tanto cuando mide sobre superficies sólidas como líquidas.

4.2. ENVÍO DE DATOS A LA PLATAFORMA THINGSPEAK

Las mediciones realizadas por el dispositivo fueron enviadas, previa conexión a internet, al canal abierto en la plataforma ThingSpeak para este proyecto. Se puede ver en la siguiente Figura 45 el canal abierto para la recepción de los datos, así como la representación gráfica de los datos agregados. Tal y como se puede apreciar en la imagen, en la plataforma ThingSpeak se georreferencia la ubicación del dispositivo. En el caso de disponer de una red de sensores, este dato sería de vital importancia.

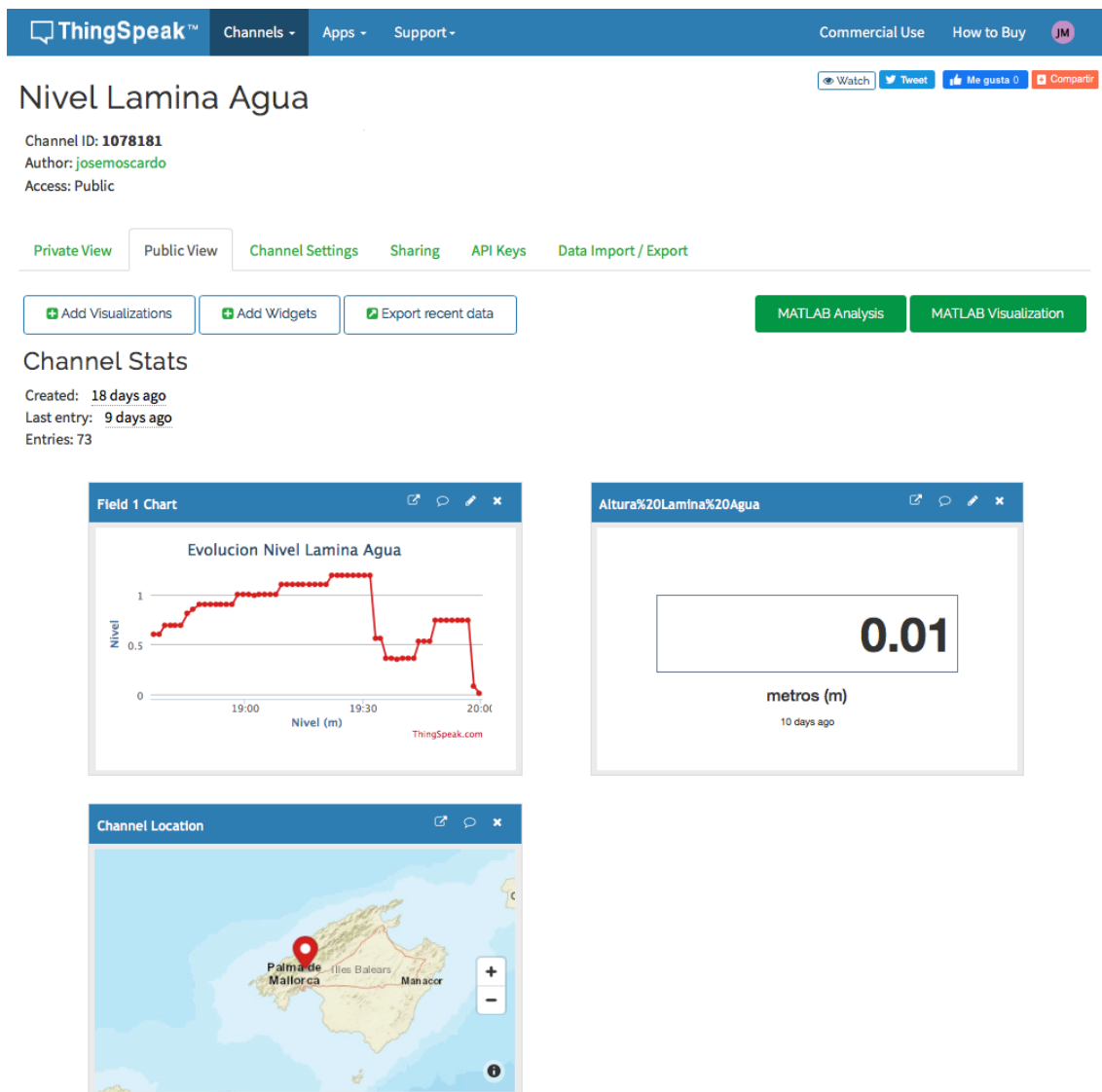


Figura 45. Canal de ThingSpeak donde se agregan los datos enviados por el prototipo

La gráfica (Fig.46) es interactiva, y permite ver el detalle de los datos recorriendo la curva con el cursor.

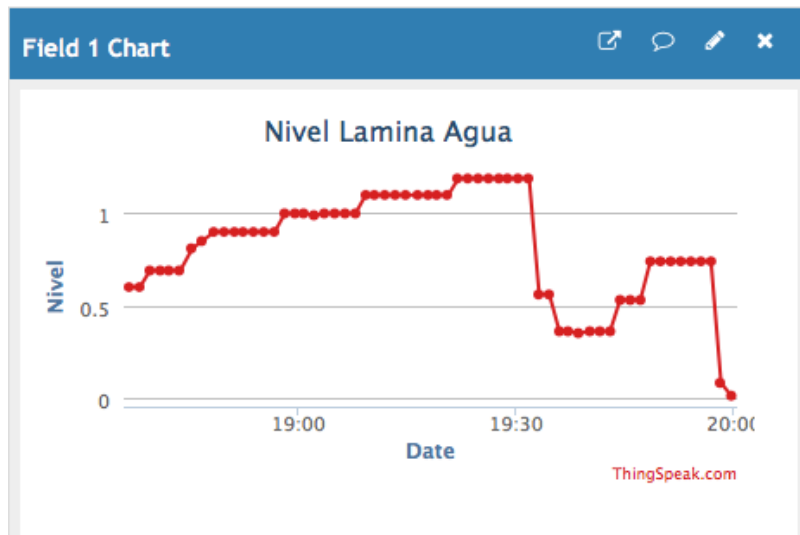


Figura 46. Detalle de la gráfica generada automáticamente con ThingSpeak

Los datos que se agregan a la plataforma se pueden descargar en diferentes formatos para su análisis. En la siguiente Tabla 14, se presentan los primeros 20 datos enviados y agregados al canal de ThingSpeak, descargados en formato CSV e incorporados a una hoja de cálculo convencional. El resto de los datos se presentan en el Anexo I del presente trabajo.

Tabla 14. Datos recogidos por la plataforma ThingSpeak

created at date	created at time	entry id	Nivel (m)
17/6/20	18:18:27	1	-0.00
17/6/20	18:19:50	2	-0.00
17/6/20	18:21:13	3	-0.01
17/6/20	18:22:37	4	0.20
17/6/20	18:24:00	5	0.19
17/6/20	18:25:24	6	0.30
17/6/20	18:26:47	7	0.41
17/6/20	18:28:10	8	0.41
17/6/20	18:29:34	9	0.41
17/6/20	18:30:58	10	0.41
17/6/20	18:32:21	11	0.41
17/6/20	18:33:45	12	0.50
17/6/20	18:35:08	13	0.50
17/6/20	18:36:46	14	0.60
17/6/20	18:38:10	15	0.60
17/6/20	18:39:33	16	0.69
17/6/20	18:40:57	17	0.69
17/6/20	18:42:20	18	0.69
17/6/20	18:43:44	19	0.69
17/6/20	18:45:23	20	0.81

Tal y como se puede apreciar en la tabla anterior, las entradas (entry_id) 1, 2 y 3 corresponde a la posición en la que se simula la base del cauce. El programa mide la distancia al obstáculo y este valor es restado a 1.66 m, siendo esta medida la distancia que se determinó entre el sensor y el lecho del cauce.

Para confirmar que los datos agregados a la plataforma ThingSpeak se corresponden con los valores medidos por el dispositivo, y que en efecto recogen correctamente la distancia entre el sensor y el obstáculo, se diseñó una simulación con datos exactos de distancia entre sensor y obstáculo: 0,0 m; 0,20 m; 0,40 m; 0,60 m; 0,80 m; 1,00 m.

Según se puede apreciar en la Figura 47 y en la tabla 15, el canal de ThingSpeak recogió con precisión los datos enviados por el dispositivo.

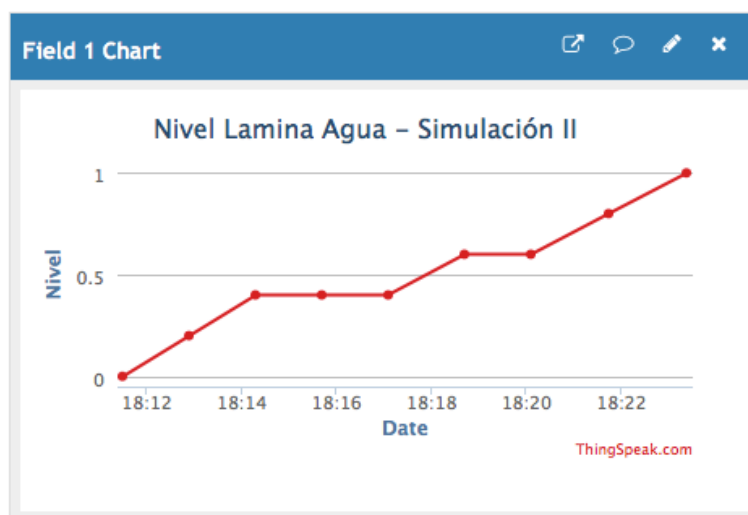


Figura 47. Datos de la Simulación II agregados a la plataforma ThnigSpeak

Tabla 15. Datos de la Simulación II agregados a la plataforma ThnigSpeak

created_at	entry_id	Nivel (m)
2020-06-19 18:11:30 CEST	1	-0.00
2020-06-19 18:12:54 CEST	2	0.20
2020-06-19 18:14:18 CEST	3	0.40
2020-06-19 18:15:41 CEST	4	0.40
2020-06-19 18:17:05 CEST	5	0.40
2020-06-19 18:18:42 CEST	6	0.60
2020-06-19 18:20:06 CEST	7	0.60
2020-06-19 18:21:44 CEST	8	0.80
2020-06-19 18:23:23 CEST	9	1.00

4.3. PUBLICACIÓN AUTOMÁTICA EN PÁGINA WEB

Los datos que son agregados en la plataforma IoT ThingSpeak se publican automáticamente en la Web creada en el presente proyecto:

<https://sites.google.com/view/sistemadealertadecrecidas/inicio>

En la Figura 48 se presenta el aspecto de la Web. Es importante resaltar que la Web permite incorporar otros datos importantes, como la localización del dispositivo, fotografías e información de servicio que en caso de riesgo de inundación podría ser gran utilidad para la población, como teléfonos de emergencia, instrucciones de Protección Civil o planes de evacuación.



Figura 48. Publicación de datos en la página Web del proyecto

4.4. PUBLICACIÓN AUTOMÁTICA EN APP PARA ANDROID

La aplicación para sistemas Android creada con MIT App Inventor, recibe de manera automatizada la información que es agregada en la plataforma IoT ThingSpeak. En la Figura 49, podemos observar la pantalla de inicio de la App (derecha), junto con la pantalla principal de la aplicación (izquierda).



Figura 49. Aplicación para sistemas Android, creada en el presente proyecto

4.5. EMISIÓN DE MENSAJES SMS DE ALERTA

Siguiendo las instrucciones de la estructura condicional que establece el envío de mensajes SMS, el dispositivo debe realizar la siguiente secuencia:

- Envío de mensaje SMS si la lámina supera el nivel 0,50 m
- Envío de mensaje SMS si la lámina supera el nivel 0,70 m
- Envío de mensaje SMS si la lámina supera el nivel 0,90 m
- Envío de mensaje SMS si la lámina supera el nivel 1,10 m

Tal y como fue programado, el dispositivo emitió mensajes de alerta mediante SMS informando de la altura de la lámina de agua cuando el nivel de esta alcanzaba los valores arriba indicados. Los mensajes que se recibieron se presentan en la siguiente Tabla 16. En la tabla se ha relacionado la hora de entrada de datos en el canal de ThingSpeak con

la hora de recepción de SMS. Como se puede apreciar, la recepción de SMS se produce con antelación a la recepción de datos en la plataforma ThingSpeak. Esto es debido a que, en la secuencia de instrucciones del programa, cuando se detecta que el nivel de la lámina de agua ha superado el nivel de alerta, en primera instancia se envía el mensaje SMS. Una vez realizada la secuencia de envío de SMS, el dispositivo se conecta a internet y agrega los datos en el canal de ThingSpeak.

Tabla 16. Secuencia de envío y recepción de SMS

Datos recogidos por la plataforma de IoT ThingSpeak				
Fecha y hora de recepción de la entrada de datos	Nº de entrada	Nivel detectado (m)	Mensaje Nº	Hora de recepción de SMS en teléfono
2020-06-17 18:36:46	14	0.60	1	18:35
2020-06-17 18:45:23	20	0.81	2	18:44
2020-06-17 18:48:24	22	0.90	3	18:47
2020-06-17 19:09:16	37	1.10	4	19:15

- Mensaje 1, es enviado al superar el nivel de la lámina de agua el primer nivel de alerta de 0,5: $0,6m > 0,5m$.
- Mensaje 2, es enviado al superar el nivel de la lámina de agua el segundo nivel de alerta de 0,7m: $0,81m > 0,7m$
- Mensaje 3, es enviado al superar el nivel de la lámina de agua el tercer nivel de alerta de 0,90m: $0,90m > 0,90m$. El envío de datos se realiza con valores que contienen dos decimales. Debemos suponer por lo tanto que el nivel detectado supera en algunos milímetros el valor de 0,90m.
- Mensaje 4, es enviado al superar el nivel de la lámina de agua el tercer nivel de alerta de 1,10m: $1,10m > 1,10m$. Al igual que se ha expuesto anteriormente, podemos suponer que el nivel detectado supera en milímetros el valor de 1,10m.

En la siguiente Figura 50 se puede ver la secuencia de recepción de datos en la plataforma ThingSpeak y la recepción de mensajes SMS. Se han identificado las entradas de datos que cumplen la instrucción condicional del sketch y generan el envío de SMS, que corresponden con las recogidas en la Tabla 16.

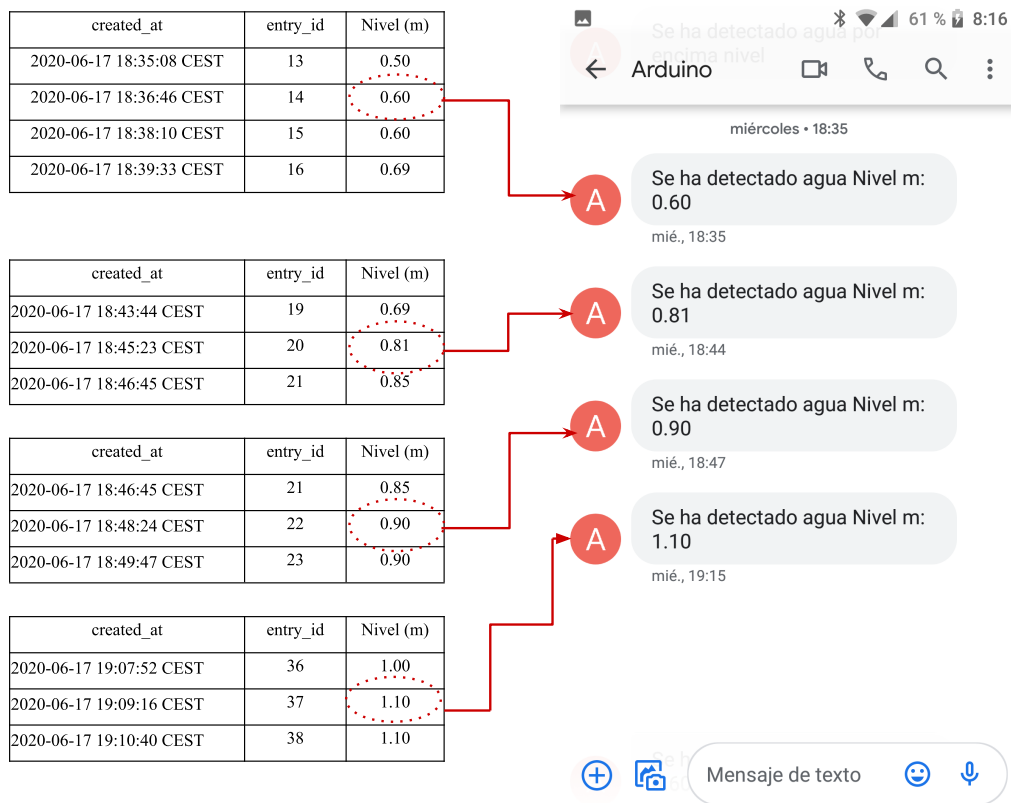


Figura 50. Envío de SMS de alerta

4.6. PUBLICACIÓN DE MENSAJES DE ALERTA EN LA RED SOCIAL TWITTER

La App ThingTwitter de ThingSpeak permite configurar la publicación automatizada de mensajes en la red social Twitter. Se definieron dos reacciones condicionales:

- Envío de mensaje tuit si la lámina de agua supera 0,50 m
- Envío de mensaje tuit si la lámina de agua supera 0,70 m

Siguiendo las instrucciones programadas, los mensajes se publicaron en la red social. En la siguiente Figura 51, se puede ver el registro en ThingSpeak de los mensajes enviados, así como los tuits publicados en la red social.

Link Twitter Account

Twitter Account	API Key	Action
pepemoscardo	D0ZZYPX[REDACTED]	Regenerate API Key Unlink Account

Reacts using ThingTweet

Name	Message	Last Sent	Twitter Account
Envío de Alerta 0.50	Detectado Nivel Lámina de Agua 0.5m	2020-06-17 17:44	pepemoscardo
Envío de Alerta 0.70 m	Detectado Nivel Lámina de Agua 0.70 m	2020-06-17 17:48	pepemoscardo



Figura 51. Publicación automática de mensajes en la red social Twitter

5. CONCLUSIONES

Se ha desarrollado, mediante hardware y software con licencia de código abierto, un sistema de monitoreo, detección y alerta de inundaciones aplicable a torrentes, ramblas y ríos.

El dispositivo diseñado reporta de manera automática los datos del nivel de la lámina de agua a la plataforma IoT de código abierto ThingSpeak, que almacena los datos en la nube y permite su posterior análisis e interpretación.

Mediante la aplicación MIT App Inventor, se ha desarrollado una aplicación móvil que recupera la información agregada en la plataforma ThingSpeak y permite el acceso a los datos de manera instantánea.

Se ha desarrollado una página Web que de manera automática difunde los datos agregados a la plataforma ThingSpeak, junto con información de servicio para actuar en caso de emergencia.

Se ha desarrollado un sistema automatizado de avisos que mediante el uso de mensajes SMS y mensajes en la red social Twitter facilitan la difusión de alertas de inundación.

El prototipo ha respondido conforme fue diseñado y programado. Los resultados han sido satisfactorios, a la vez que se abren nuevas líneas de trabajo e investigación.

Una vez desarrollado el prototipo, éste puede ser replicado, modificado y adaptado a nuevas y diferentes necesidades. La fácil instalación y portabilidad del mismo facilita su uso en diferentes escenarios.

Como conclusión final, podemos afirmar que mediante el hardware y software de código abierto pueden desarrollarse tecnologías económicas, versátiles y adaptadas a las necesidades del contexto rural.

6. BIBLIOGRAFÍA

ADAFRUIT INDUSTRIES. (2020). Página web de la empresa Adafruit. Disponible en: <https://www.adafruit.com/>. Accedido: 15 de abril 2020.

ALDERMAN, K., TURNER, L.R., TONG, S. (2012) Floods and human health: A systematic review. *Environment international*, 47, pp. 37–47.

ARDUINO. (2020). About us. Arduino-Home web page. Disponible en: <https://www.arduino.cc/en/Main/AboutUs>. Accedido: 1 de junio de 2020.

AZACORRA, A., VALERA, F., VIDAL, I. (2017). Evaluación y propuesta estratégica para la mejora de los sistemas de comunicación en los organismos de cuenca. IMDEA Networks.

BANZI, M. (2009). Getting started with Arduino. Ed. O'Reilly Media, Inc. Sebastopol

BERNAL, A., BARO, J., DÍAZ, C. (2016). Estimación de pérdidas económicas directas provocadas por inundación. Aplicación de las curvas inundación-daños en países en desarrollo. *Investigaciones Geográficas*. 65, pp. 169-180.

BODOQUE, J.M., DÍEZ-HERRERO, A., AMÉRIGO M., GARCÍA, J.A., OLCINA, J. (2019). Enhancing flash flood risk perception and awareness of mitigation actions through risk communication: A pre-post survey design. *Journal of Hydrology*, 568, pp. 769-779.

BUSTILLOS, A., EBERS, M., RIBBE, L. (2017). What influences disaster risk perception? Intervention measures, flood and landslide risk perception of the population living in flood risk areas in Rio de Janeiro state, Brazil. *International Journal of Disaster Risk Reduction*, 25, pp. 227-237.

CAJIGAL, E., MALDONADO, A.L. (2019). Metodología para el análisis de vulnerabilidad ante inundaciones. Un ejercicio emergente ante el cambio climático. *Economía, Sociedad y Territorio*, vol. XIX, núm. 61, pp. 543-574.

CAMARASA, A.M. (2002). Crecidas e inundaciones. En: F.J. Ayala-Carcedo y J. Olcina Cantos (Coords.), *Riesgos Naturales*. 46, pp. 859-877, Ed. Ariel Ciencia, Barcelona.

CAROLAN, M. (2018). Smart Farming: Wither Do It Yourself (DIY) Farmers. *Sociology Lens*. Disponible en: <https://www.sociologylens.net/article-types/opinion/smart-farming-wither-diy-farmers/20125>. Accedido: 10 de mayo 2020.

CASTILLO, J. (2017) Diseño e implementación de un dispositivo IoT de bajo coste para entornos agrícolas. TFM. ETSIIT. Universidad de Granada.

CEBALLOS, A.I., BARÓ, J.E., DÍAZ, C. (2016). Estimate of direct economic losses caused by floods. Application of flood-damage curves in developing countries. *Investigaciones Geográficas*, 65.

CENTRE FOR RESEARCH ON THE EPIDEMIOLOGY OF DISASTERS. CRED. (2018). Pérdidas económicas, pobreza y Desastres 1998-2017. Université catholique de Louvain.

CENTRE FOR RESEARCH ON THE EPIDEMIOLOGY OF DISASTERS. CRED. (2020). Disaster Year 2019 in Review. Abril 2020. Issue n 58.

CONFEDERACIÓN HIDROGRÁFICA DEL JÚCAR. (2020). Red de Estaciones de Aforos y Embalses. Disponible en <https://www.chj.es/es-es/medioambiente/redescontrol/Paginas/RedEstacionesAforosyEmbalses.aspx>. Accedido: 23 de mayo de 2020.

CREATIVE COMMONS. (2019). Main web page. Disponible en: <https://creativecommons.org/>. Accedido: 27 de mayo de 2010.

CUAUHTEMONC, J., ROMANTCHIK, E. LOPEZ, G. (2018). Arduino Board in the Automation of Agriculture in Mexico, a Review. *International Journal of Horticulture*, Vol. 8, No. 6, 52-68.

D'AUSILIO, A. (2012). Arduino: A low-cost multipurpose lab equipment. *Behavior Research Methods*. June 2012, Volume 44, Issue 2, pp 305–313.

DIDY. (2015). Foundational and interpretation of DiDY. Disponible en <http://www.didiy.eu/public/deliverables/didiy-d2.2.pdf>. Accedido: 27 de abril de 2020.

DÍEZ, A., LLORENTE, M., BALLESTEROS, J. RUIZ, V.(2009). Riesgos por avenidas e inundaciones fluviales. *Enseñanza de las Ciencias de la Tierra*, 2009. pp 254-263

DIRECCIÓN GENERAL DE PROTECCIÓN CIVIL Y EMERGENCIAS. DGPCE. (2020). Inundaciones. Disponible en: <http://www.proteccioncivil.es/riesgos/inundaciones/presentacion>. Accedido: 22 de mayo de 2020.

EUROPEAN ENVIROMENT AGENCY. EEA (2017). Climate change impacts and vulnerabilities 2016. An indicator based report. Ed. European Union. Luxembourg.

FIORETTI, M. (2015). Digital DIY for self-sustainability of rural areas. *Liuc Papers. Università Carlo Cattaneo*. Vol. 297.

FISHER, K. (2012). Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research. *Modern Instrumentation*. Vol.1. 8-20.

FOX, S. (2014). Third Wave Do-It-Yourself (DIY): Potential for prosumption, innovation, and entrepreneurship by local populations in regions without industrial manufacturing infrastructure. *Technology in Society*, 39, pp. 18-30.

GARIJO, C., MEDIERO, L., GARROTE, L. (2018). Utilidad de las proyecciones climáticas generadas por AEMET para estudios de impacto del cambio climático sobre avenidas a escala nacional. *Ingeniería del Agua*, 22, pp153-166. Ed. UPV. Valencia.

HANDSOME TECHNOLOGY (2020). I2c Serial Interface 1602LCD Module. Disponible en http://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf . Accedido: 2 de junio 2020.

HASIM, Z., MAZLAN, S., ABD, A., (2015). Agriculture monitoring system: A study. *Journal Technology*. 77.

HENRY, E. (2016). Precision apiculture: development of a wireless sensor network for honeybee hives. Thesis. Department of Bioresource Engineering. Macdonald Campus, McGill University Montreal, Quebec, Canada

INTERGOVERNAMENTAL PANEL ON CLIMATE CHANGE. IPCC (2015). Climate change 2014: Synthesis report. Contribution of working groups I, II and III to the fifth assessment report of the Intergovernmental Panel on Climate Change. Geneva, Switzerland.

JAHANKIT (2020) JSN-SR04T-2.0. Ultrasonic Waterproof Module. Disponible en: <https://www.jahankitshop.com/getattach.aspx?id=4635&Type=Product> Accedido: 3 mayo 2020.

JONGMAN, B., WARD, P., AERTS, J.C. (2012). Global exposure to river and coastal flooding: Long term trends and changes. *Global Environmental Change*, 22, pp. 823-835.

KELLENS, W., TERPSTRA, T., DE MEYER, P. (2012). Perception and Communication of Flood Risks: A Systematic Review of Empirical Research. *Risk Analysis*, Vol 33, pp. 24-49.

KOUÏA, E. (2016). Development of an Arduino-Based embedded system. Case: Greenhouse monitoring. University of Applied Sciences Europe, Hamburg.

KUANTAMA, E., SETYAWAN, L., DARMA, J. (2012). Early flood alerts using Short Message Service (SMS). *International Conference on System Engineering and Technology (ICSET)*. pp 1-5.

KUZMETSOV, A., PAULO, E., (2010). Rise of the expert amateur: DIY projects, communities, and cultures. *Proceedings of the 6th Nordic Conference on Human-Computer Interaction*, pp. 295-304.

LÓPEZ, V., SÁNCHEZ, A. (2016). Rediseño de un seguidor solar. B.Sc., Thesis. Chapingo Autonomous University, México.

LOWRANCE, C. (2012). Open source hardware and software in agriculture: An autonomous sap flow measurement wireless network & user friendly management zone delineation tool. Master of Science. Thesis. University of Georgia.

MEEMNAKSI, M. (2016). Advance Cattle Health Monitoring System Using Arduino and IOT. *Electronics and Instrumentation Engineering*. Vol. 5, Issue 4, pp. 3365-3370.

MERIGO, M., GARCÍA, J.A., PÉREZ-LÓPEZ, R., TALAYERO, F. (2019). Retos para afrontar el cambio climático: Análisis de la percepción del riesgo de inundación ante riadas. *Papéles del Psicólogo / Psychologist Papers*, 2019 Vol. 40(3), pp. 197-204.

MINISTERIO DE LA PRESIDENCIA (2010). Real Decreto 903/2010, de 9 de julio, de evaluación y gestión de riesgos de inundación. BOE num. 171. Pp. 61954 a 61967

MINISTERIO PARA LA TRANSICIÓN ECOLÓGICA Y EL RETO DEMOGRÁFICO. MITECO. (2018). Perfil Ambiental de España 2017. Ed. MITECO, Madrid.

MOHD, Z., TENGKU, I., RAZAK, RAHIM, A. (2016). Development of portable water level sensor for flood management system. *Journal of Engineering and Applied Sciences*. 11, pp. 53-52.

MOORE, R., BELL, A., COLE, S., JONES, D. (2006). Issues in flood forecasting: ungauged basins, extreme floods and uncertainty. *Frontiers in Flood Research, 8th Kovacs Colloquium*. 30, pp. 103–122 Ed. UNESCO, Paris.

MOSCARDO, J. (2019). Hardware de código abierto para la producción y la investigación agraria. TFM. Universitat de les Illes Balears, Palma.

MOY, M., DICHT, S., LEITAO, J., (2017). Urban flash flood experiments monitored with conventional and alternative sensors. *Earth System Science*. 9, pp. 657-666.

NACIONES UNIDAS ONU (2002). Cumbre Mundial sobre el Desarrollo Sostenible, Johannesburgo. Documentos de la Conferencia. Issue: A/CONF.199/20. Ed. UN, New York.

OPEN SOURCE HARDWARE ASSOCIATION OSHW. (2019) Página web de OSHW. Disponible en: <https://www.oshwa.org/definition/spanish/> (Accedido: 11/05/2020)

ORGANIZACIÓN MUNDIAL DE METEOROLOGÍA. OMM. (2008). Guía Prácticas Hidrológicas. OMM, n. 168. Ginebra.

ORGANIZACIÓN MUNDIAL DE METEOROLOGÍA. OMM. (2011). Manual sobre predicción y avisos de crecidas. OMM, n.107. Ginebra

ORGANIZACIÓN MUNDIAL DE METEOROLOGÍA. OMM. (2019). Gestión integrada de crecidas: documento conceptual. OMM n. 1047. Ed. OMM. Ginebra.

PEARCE, J.M., (2015). Quantifying the Value of Open Source Hardware Development. *Modern Economy*, 6, pp.1-11.

POOJARY, R., DUDHE, R., RAJAN, A. (2016). Efficient Automated Fruit and Vegetable Sorter Based on Colour Detection. *SSRG International Journal of Electronics and Communication Engineering*. Vol. 4 Issue 4 , pp. 30-34.

RASHID, H., IFTEKHAR, A., TASLIM, R. (2017). Solar Powered Smart Ultrasonic Insects Repellent with DTMF and Manual Control for Agriculture. Imaging, Vision & Pattern Recognition. *IEEE International Conference*. Disponible en:<https://doi.org/10.1109/> (Accedido el 20 de abril de 2020)

RODRIGUEZ, J.M., RAJAN, S., MUKHOPADADHYAY, A. (2016). Exposure is Associated with Higher Prevalence of Child Undernutrition in Rural Eastern India. *Int. J. Environ.* 13, pp.210.

RUÍZ, O., RODRÍGUEZ, J.P. (2013). Impacto de la biotecnología en los sectores agrícola y ganadero 2050. Ed. Fundación Española para la Ciencia y la Tecnología, FECYT. Madrid.

SADLER, J.M., AMES, D.P. , KHATTAR, P. (2016). A recipe for standards-based data sharing using open source software and low-cost electronics. *Journal of Hydroinformatics*, 2016. 18(2): pp. 185-197.

SÁNCHEZ, J.A. (2005). Análisis y estudio de Redes GPRS.TFG Universidad Austral de Chile

SHUVORDEB, B., REDHAWAN, R., SHAJARATUL, I. MAHMUDUL, H. (2017). Proposed Design for Automated Agro-Quadcopter "Agro-ropter". *1st International Conference on Mechanical Engineering and Applied Science Military Institute of Science and Technology*, Dhaka, Bangladesh.

SIMCOM (2010). SIM900 Reference Design Guide V102. Disponible en: https://simcom.ee/documents/SIM900/AN_SIM900%20Reference%20Design%20Guide_V1.02.pdf. Accedido: 6 de mayo de 2020.

SPARKFUN ELECTRONICS (2020) *Página web de la empresa Sparkfun*. Disponible en: <https://www.sparkfun.com/>. Accedido: 15 de mayo de 2020.

TABARÉS, R. (2018). La importancia de la cultura tecnológica en el movimiento maker. *Arbor*, 194 (789).

TABARÉS, R., SOPELANA, A., GARCÍA, J., MORENO, M. (2016). Social Manufacturing: Towards the popularization of personalized fabrication. *Science and Technology EASST / 4S* 2016. Barcelona.

THARMATHARSANAN, R., ALVAPPILLAI, P. (2016). Fabrication of Low Cost Automated Lime Colour Sorter. *International Journal of Environmental & Agriculture Research*. Vol-2, Issue-12.

UNION EUROPEA. (2007). Directiva 2007/60/CE del Parlamento Europeo y del Consejo de 23 de octubre de 2007, relativa a la evaluación y gestión de los riesgos de inundación. Diario Oficial de la Unión Europea. L 288/27

WORDL METEROLOGICAL ORGANIZATION. WMO. (1992). International Glossary of Hydrology. WMO, n. 385.

YILMAZ, M. (2017). Arduino bases automation in the soilless agriculture. 65th ISERD International Conference, Mecca. pp 51-57.