



**UNIVERSIDAD ESTADAL DE MILAGRO
FACULTAD CIENCIAS E INGENIERÍA**

**TRABAJO DE TITULACIÓN DE GRADO PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS
COMPUTACIONALES**

PROYECTO TÉCNICO

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA
MONITOREO DE UN AMBIENTE DE BEBÉS CON LA
UTILIZACIÓN DE RASPBERRY PI.**

Autores:

Sr. HUAIPATIN VELASCO JOSÉ LUIS

Sr. PEÑAFIEL PARRA FREDDY MAURICIO

Acompañante:

MBA. PÁNCHEZ HERNÁNDEZ RAÚL

Milagro, Octubre 2019

ECUADOR

DERECHOS DE AUTOR

Ingeniero.

Fabricio Guevara Viejó, PhD.

RECTOR

Universidad Estatal de Milagro

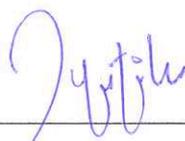
Presente.

Yo, HUAIPATIN VELASCO JOSÉ LUIS, en calidad de autor y titular de los derechos morales y patrimoniales de alternativa de Titulación – Proyecto Técnico modalidad presencial, mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor del Proyecto Técnico realizado como requisito previo para la obtención de mi Título de Grado, como aporte a la Línea de Investigación **TECNOLOGÍA DE LA INFORMACIÓN Y DE LA COMUNICACIÓN**, de conformidad con el Art. 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, concedo a favor de la Universidad Estatal de Milagro una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Milagro para que realice la digitalización y publicación de este Proyecto Técnico en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Milagro, 31 de octubre de 2019



HUAIPATIN VELASCO JOSÉ LUIS

Autor 1

CI: 0916870405

DERECHOS DE AUTOR

Ingeniero.

Fabricio Guevara Viejó, PhD.

RECTOR

Universidad Estatal de Milagro

Presente.

Yo, PEÑAFIEL PARRA FREDDY MAURICIO, en calidad de autor y titular de los derechos morales y patrimoniales de alternativa de Titulación – Proyecto Técnico modalidad presencial, mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor del Proyecto Técnico realizado como requisito previo para la obtención de mi Título de Grado, como aporte a la Línea de Investigación TECNOLOGÍA DE LA INFORMACIÓN Y DE LA COMUNICACIÓN, de conformidad con el Art. 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, concedo a favor de la Universidad Estatal de Milagro una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Milagro para que realice la digitalización y publicación de este Proyecto Técnico en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Milagro, 31 de octubre de 2019

PEÑAFIEL PARRA FREDDY MAURICIO

Autor 2

CI: 1803074127

APROBACIÓN DEL TUTOR DE

PROYECTO TÉCNICO

Yo, PÁNCHEZ HERNÁNDEZ RAÚL en mi calidad de tutor del Proyecto Técnico, elaborado por los estudiantes HUAIPATIN VELASCO JOSÉ LUIS y PEÑAFIEL PARRA FREDDY MAURICIO, cuyo título es DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO DE UN AMBIENTE DE BEBÉS CON LA UTILIZACIÓN DE RAPSBERRY PI, que aporta a la Línea de Investigación INGENIERO EN SISTEMAS COMPUTACIONALES previo a la obtención del Grado; considero que el mismo reúne los requisitos y méritos necesarios en el campo metodológico y epistemológico, para ser sometido a la evaluación por parte del tribunal calificador que se designe, por lo que lo APRUEBO, a fin de que el trabajo sea habilitado para continuar con el proceso de titulación de la alternativa de Proyecto Técnico de la Universidad Estatal de Milagro.

Milagro, 31 de octubre de 2019



PÁNCHEZ HERNÁNDEZ RAÚL

Tutor

C.I: 0602179202

APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

MBA. PÁNCHEZ HERNÁNDEZ RAÚL RUPERTO

MBA. LÓPEZ BRIONES JHONNY RODDY

MBA. MENDOZA CABRERA DENIS DARIO

Luego de realizar la revisión del Proyecto Técnico, previo a la obtención del título (o grado académico) de INGENIERA EN SISTEMAS COMPUTACIONALES presentado por el estudiante HUAIPATIN VELASCO JOSÉ LUIS

Con el tema de trabajo de Titulación: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO DE UN AMBIENTE DE BEBÉS CON LA UTILIZACIÓN DE RASPBERRY PI.

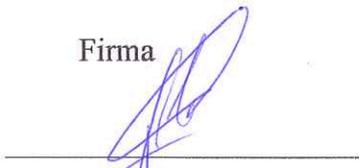
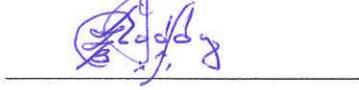
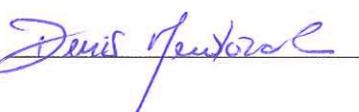
Otorga al presente Proyecto Técnico, las siguientes calificaciones:

Proyecto Técnico	[80]
Defensa oral	[20]
Total	[100]

Emite el siguiente veredicto: (aprobado/reprobado) Aprobado

Fecha: 31 de octubre de 2019

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	PÁNCHEZ HERNÁNDEZ RAÚL RUPERTO	
Secretario /a	LÓPEZ BRIONES JHONNY RODDY	
Integrante	MENDOZA CABRERA DENIS DARIO	

APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

MBA. PÁNCHEZ HERNÁNDEZ RAÚL RUPERTO

MBA. LÓPEZ BRIONES JHONNY RODDY

MBA. MENDOZA CABRERA DENIS DARIO

Luego de realizar la revisión del Proyecto Técnico, previo a la obtención del título (o grado académico) de INGENIERO EN SISTEMAS COMPUTACIONALES presentado por el estudiante PEÑAFIEL PARRA FREDDY MAURICIO

Con el tema de trabajo de Titulación: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO DE UN AMBIENTE DE BEBÉS CON LA UTILIZACIÓN DE RASPBERRY PI.

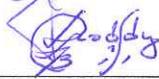
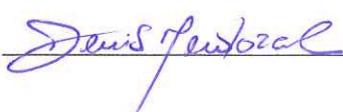
Otorga al presente Proyecto Técnico, las siguientes calificaciones:

Proyecto Técnico	[80]
Defensa oral	[20]
Total	[100]

Emite el siguiente veredicto: (aprobado/reprobado) Aprobado

Fecha: 31 de octubre de 2019

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	PÁNCHEZ HERNÁNDEZ RAÚL RUPERTO	
Secretario /a	LÓPEZ BRIONES JHONNY RODDY	
Integrante	MENDOZA CABRERA DENIS DARIO	

DEDICATORIA

Dedico esta tesis a mi padre quien me enseñó que el mejor conocimiento que se puede tener es el de aprender y adquirir conocimiento día a día.

HUAIPATIN VELASCO JOSÉ LUIS

Este trabajo está dedicado con todo mi cariño para mi esposa e hijos; quienes han puesto toda su confianza para lograr un objetivo más en mi vida.

PEÑAFIEL PARRA FREDDY MAURICIO

AGRADECIMIENTO

Mi profundo agradecimiento para todas aquellas personas que siempre estuvieron en cada momento del proceso de mi carrera universitaria.

HUAIPATIN VELASCO JOSÉ LUIS

Quiero expresar un sincero agradecimiento, a Dios por guiarme en mi camino y por permitirme concluir con mi objetivo.

También hago extenso este reconocimiento a mi Madre querida por su confianza y apoyo incondicional para poder cumplir una meta más.

A mi hermano Darwin. Por creer en mí quien supo apoyarme en todo momento para que pueda terminar esta carrera.

PEÑAFIEL PARRA FREDDY MAURICIO

ÍNDICE GENERAL

DERECHOS DE AUTOR.....	ii
DERECHOS DE AUTOR.....	iii
APROBACIÓN DEL TUTOR DE	iv
PROYECTO TÉCNICO	iv
APROBACIÓN DEL TRIBUNAL CALIFICADOR	v
APROBACIÓN DEL TRIBUNAL CALIFICADOR	vi
DEDICATORIA	vii
AGRADECIMIENTO.....	viii
ÍNDICE GENERAL.....	ix
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS	xiii
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN	3
CAPÍTULO 1	4
PROBLEMA	4
CAPÍTULO 2.....	5
ANTECEDENTES Y MARCO TEÓRICO	5
2.1 ANTECEDENTES.....	5
2.2 MARCO TEÓRICO.....	6
2.2.1 Raspberry PI.....	6
2.2.2 Cámara PI NoIR	7
2.2.3 Arduino Uno R3.....	9
2.2.4 Módulo de Gas	10
2.2.5 Sensor de Movimiento PIR-HC-SR501	11
2.2.6 Módulo de Temperatura y Humedad.....	13
2.2.7 Servidor.....	13
2.2.8 Servidor Web	14
2.2.9 Lenguaje de Programación.....	14
2.2.10 Mensajería Instantánea.....	15
2.2.11 OPEN HARDWARE	15
2.2.12 HARDWARE LIBRE	16
2.2.13 OPEN SOURCE.....	17

2.2.14 SOFTWARE LIBRE.....	17
2.2.15 APLICACIÓN WEB.....	18
2.2.16 PROTOTIPO.....	18
2.2.17 PYTHON.....	18
CAPÍTULO 3.....	20
DESCRIPCIÓN DEL PROYECTO TÉCNICO.....	20
3.1 TITULO Y DESCRIPCIÓN DEL PROYECTO.....	20
3.2 OBJETIVO GENERAL Y ESPECÍFICO.....	20
3.2.1 Objetivo General.....	20
3.2.2 Objetivo Específico.....	21
3.3 DESARROLLO.....	21
3.4.1 DISEÑO DE HARDWARE.....	22
3.4.2 CONEXIÓN DE SENSORES.....	22
3.4.3 DESARROLLO DE SOFTWARE.....	40
3.4.4 PRUEBAS DE INSTALACIÓN DEL PROTOTIPO.....	47
3.4 CRONOGRAMAS DE ACTIVIDADES.....	50
3.5 ANÁLISIS DE RECURSOS.....	52
CAPÍTULO 4.....	54
METODOLOGÍA Y EJECUCIÓN DEL PROYECTO TÉCNICO.....	54
4.1 METODOLOGÍA UTILIZADO PARA EL PRESENTE PROYECTO.....	54
CAPÍTULO 5.....	65
EVALUACIÓN DEL PROYECTO TÉCNICO.....	65
5.1 EVALUACIÓN.....	65
5.2 CRITERIOS DE ACEPTACIÓN.....	68
5.3 INFORME DE ASEGURAMIENTO DE CALIDAD.....	69
5.4 MECANISMOS DE CONTROL.....	70
CONCLUSIONES.....	72
RECOMENDACIONES.....	73
ANEXOS.....	74
REFERENCIAS BIBLIOGRÁFICAS.....	115

ÍNDICE DE FIGURAS

Figura 1. Raspberry PI Modelo B+	7
Figura 2. Cámara PI NoIR.....	7
Figura 3. Placa Arduino Uno R3	10
Figura 4. Sensor de Gas MQ2	11
Figura 5. Sensor de Movimiento PIR.....	12
Figura 6. Sensor PIR HC-SR501.....	12
Figura 7. Módulo de Humedad y Temperatura	13
Figura 8. Figura Esquema General del funcionamiento del prototipo	21
Figura 9. Figura Esquema del circuito lógico	22
Figura 10. Figura Sensor de temperatura y humedad.....	22
Figura 11. Figura Conexión de sensor de temperatura.....	23
Figura 12. Ejecutar Arduino IDE	24
Figura 13. Consola de Arduino IDE.....	25
Figura 14. Figura esquema de montaje sensor de gas	27
Figura 15. Esquema de montaje módulo de gas	27
Figura 16. Figura esquema del sensor de gas con el Arduino	28
Figura 17. Iniciar la consola de Arduino IDE	28
Figura 18. Figura detección de movimiento.....	33
Figura 19. Figura sensor de movimiento.....	34
Figura 20. Figura Esquema de patillaje de un sensor PIR.....	34
Figura 21. Figura esquema eléctrico	35
Figura 22. Figura esquema de conexión de sensor de movimiento con el Arduino.....	35
Figura 23. Programa Arduino IDE	36
Figura 24. Programa de Arduino IDE	36
Figura 25. Figura Raspberry PI	38
Figura 26. Figura Camara conectado al Raspberry PI.....	39
Figura 27. Terminal de Raspberry PI.....	41
Figura 28. Funcionamiento del prototipo para el monitoreo del ambiente de bebé	47
Figura 29. Funcionamiento del prototipo para el monitoreo del ambiente de bebé	47
Figura 30. Conexión de sensores.....	48
Figura 31. Notificación de las alertas.....	49
Figura 32. Cálculo de tamaño de muestra	54
Figura 33. Resultado Pregunta 1	56
Figura 34. Resultado pregunta 2	57

Figura 35. Resultado pregunta 3	58
Figura 36. Resultado pregunta 4	59
Figura 37. Resultado pregunta 5	60
Figura 38. Resultado pregunta 6	61
Figura 39. Resultado pregunta 1	62
Figura 40. Resultado pregunta 2	63
Figura 41. Resultado pregunta 2	64
Figura 42. Consola de Terminal de Raspberry PI	86
Figura 43. Página de Twilio	87
Figura 44. Formulario de registro para Twilio	87
Figura 45. Modelo Entidad Relación	101
Figura 46. Pantalla Principal	102
Figura 47. Interfaz de inicio de sesión	103
Figura 48. Interfaz de ingreso de usuario	104
Figura 49. Menú de administración.....	105
Figura 50. Menú cuenta.....	106
Figura 51. Figura Información personal.....	106
Figura 52. Figura Datos del bebe	107
Figura 53. Figura Cambio de clave	108
Figura 54. Figura Lecturas de los sensores	108
Figura 55. Figura Ingreso y seguridad	109
Figura 56. Figura limpiar cache y logs.....	110
Figura 57. Figura Sistema de Usuarios	110
Figura 58. Figura Sistema de Usuarios Group	111
Figura 59. Figura Logs	112
Figura 60. Figura Video tiempo real	112
Figura 61. Figura: Reporte	113

ÍNDICE DE TABLAS

Tabla 1. Características de Raspberry PI.	6
Tabla 2. Características cámara PI NoIR	8
Tabla 3. Características técnicas y funcionamiento Arduino Uno R3.....	9
Tabla 4. Talento Humano.....	52
Tabla 5. Recursos técnicos	52
Tabla 6. Recursos financieros	53
Tabla 7. Resultado pregunta 1	56
Tabla 8. Resultado pregunta 2.....	57
Tabla 9. Resultado pregunta 3.....	58
Tabla 10. Resultado pregunta 4.....	59
Tabla 11. Resultado pregunta 5.....	60
Tabla 12. Resultado pregunta 6.....	61
Tabla 13. Resultado de la pregunta 1	62
Tabla 14. Resultado de la pregunta 2	63
Tabla 15. Resultado de la pregunta 2	64
Tabla 16. Raspberry PI 3 Modelo B+	65
Tabla 17. Arduino	65
Tabla 18. Cámara PI.....	65
Tabla 19. Sensor de temperatura y humedad	66
Tabla 20. Sensor de gas.....	66
Tabla 21. Sensor de detección de personas	66
Tabla 22. Aplicación Web.....	67
Tabla 23. Base de Datos.....	67
Tabla 24. Servicio de Monitoreo.....	67
Tabla 25. Servicio de mensajería	67
Tabla 26. Servicio de Streaming	67
Tabla 27. Matriz de criterio de aceptación	68
Tabla 28. Detalle de las actividades del SQA	69
Tabla 29. Historia de usuario 1	96
Tabla 30. Historia de usuario 2	96
Tabla 31. Historia de usuario 3	97
Tabla 32. Historia de usuario 4	97
Tabla 33. Historia de usuario 5	98
Tabla 34. Historia de usuario 6	98

Tabla 35. Historia de usuario 7	99
Tabla 36. Historia de usuario 8	99
Tabla 37. Historia de usuario 9	100

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO DE UN AMBIENTE DE BEBÉS CON LA UTILIZACIÓN DE RASPBERRY PI

RESUMEN

El objetivo del presente proyecto de tesis es buscar una opción tecnológica para afrontar el problema que en la actualidad afrontan las personas que están en constante vigilancia y cuidado del bebé. Para esto se realizará una investigación de las alternativas actuales que existen en el mercado para realizar la vigilancia del lugar donde se encuentra el infante. Mediante la metodología de investigación exploratoria realizaremos una encuesta a cien familias para conocer su opinión sobre el uso de la tecnología para el monitoreo de ambiente del bebé, que características debería ofrecer el dispositivo con la finalidad de lograr un prototipo que cumpla con las necesidades del usuario y el costo de inversión sea accesible, por tal motivo el software y hardware que se utilizarán para la construcción del prototipo son de código abierto. La metodología de desarrollo que se usó para el presente proyecto fue Extreme Programming XP, ya que es una metodología ágil que propone un desarrollo de software dentro de un corto periodo de tiempo luego de cada retroalimentación con el usuario. El dispositivo será capaz de monitorear en tiempo real la habitación del bebé, la temperatura, la humedad, enviar alertas mediante mensajería de texto, una base de datos para recopilar información y generar reportes.

PALABRAS CLAVES: Raspberry Pi, Ambiente de bebé, Metodología Ágiles.

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO DE UN AMBIENTE DE BEBÉS CON LA UTILIZACIÓN DE RASPBERRY PI

ABSTRACT

The objective of this thesis project is to look for a technological option to face the problem currently facing people who are constantly monitoring and caring for the baby. For this, an investigation of the current alternatives that exist in the market to monitor the place where the infant is located is activated. Through the exploratory research methodology, we will conduct a survey of one hundred families to know their opinion on the use of technology for monitoring the baby's environment, what features the device should provide with the application of achieving a prototype that meets the needs of the user and the cost of accessible sea investment, for this reason the software and hardware used for the construction of the prototype are open source. The development methodology used for this project was Extreme Programming XP, since it is an agile methodology that proposes a software development within a short period of time and then every feedback with the user. The device will be able to monitor in real time the baby's room, temperature, humidity, send alerts via text messaging, a database to collect information and generate reports.

KEY WORDS: Raspberry Pi, Environment baby, Extreme Programming XP.

INTRODUCCIÓN

Un recién nacido a la familia conlleva la responsabilidad de proteger al bebé, los padres están conscientes que se debe tener cuidados únicos debido que por el momento son pequeños y de mucho cuidado.

Los padres saben la responsabilidad que tienen en sus manos al tener su bebe en brazos, los progenitores deben sentirse atareados con el cambio de la rutina cotidiana antes del nacimiento del bebé, van a necesitar de tiempo, energía, dinero, paciencia y conocimientos para satisfacer todo lo que demanda tener un bebé en el seno familiar.

Este proyecto tiene el propósito de dar a conocer a los padres lo importante que es hoy en día utilizar la tecnología para el cuidado de un recién nacido, sin embargo al proponer soluciones tecnológicas para monitorear el ambiente del bebé, la primera pregunta que los padres tienen es el costo de la solución tecnológica, utilizando esta hipótesis, para desarrollar este prototipo de tesis de graduación se analizó los equipos de hardware y software libre, a fin de implementar un dispositivo para monitorear el ambiente de los niños recién nacidos e incluir con la mayoría de características que ofrecen actualmente los dispositivos para cuidado de los recién nacidos.

CAPÍTULO 1

PROBLEMA

El espacio en donde empieza a formarse la familia se llama hogar, basados en los estudios realizados del INEC dentro del Ecuador, la protección de los menores de edad, lo cumplen los papas, y familiares en sus hogares que representa el 59% de la población del territorio del Ecuador; pese a la seguridad que ofrece dentro el hogar, los integrantes de la familia se encuentran en riesgo de sufrir cualquier tipo de accidente, los niños son más propensos a tener accidentes y se convierte en el miembro vulnerable.

Las estadísticas de eventos familiares se encuentran entre las diez primeras causas de orígenes de mortandad de infantes, se debe tener en consideración que durante los primeros años de vida del niño existen factores que pueden provocar daños, como por ejemplo el ambiente contaminado, altas temperaturas, monóxido de carbono, puede causar problemas respiratorios que resulta difícil de identificar el grado de contaminación en los hogares.

El cambio de temperatura causa perjuicios en la salud del niño como pueden ser como alergia, irritaciones, comezón, malestar en el niño y peligro de muerte.

La mayoría de los padres no cuentan con la disponibilidad de tiempo para estar presente en la vigilancia de los niños y no tienen suficiente conocimiento de cuidado del infante, el menor en sus inicios de vida en donde son vulnerables y dependen por completo de alguna persona adulta.

CAPÍTULO 2

ANTECEDENTES Y MARCO TEÓRICO

2.1 ANTECEDENTES

En la ciudad de Guayaquil se realizó un prototipo de monitoreo de temperatura de infantes menores de dos años basados en Raspberry PI, donde el dispositivo realiza una transmisión en tiempo real y utiliza un sensor de temperatura infrarrojo para monitorear la temperatura corporal del niño, el Raspberry PI recibe la información del sensor que son enviados a una aplicación en un teléfono móvil. La propuesta que nosotros estamos elaborando es adicionar otras características al prototipo como son: sensor de gas, sensor de humedad y una base de datos para recopilar la información de las actividades de los sensores. ([Tomalá Asunción, Andrea](#))

Existe un proyecto de tesis que fue desarrollado en la ciudad de Quito que es el diseño de un sistema de control para una incubadora neonatal para monitorear los signos vitales tales como: temperatura corporal, saturación de oxígeno en la sangre y el ritmo cardiaco. El proyecto que antecede no cuenta con monitoreo en tiempo real de la incubadora ni tampoco de los gases nocivos que pueden estar en el ambiente de infante. La propuesta de nuestro presente proyecto realiza un monitoreo del ambiente del bebe una vez terminado su periodo de estancia en la maternidad que incluyendo diferentes sensores además las notificaciones por medio de la mensajería instantánea. ([Bustamante Campoverde Jaime y Cevallos Larrea Andrés](#))

Actualmente se implementó un sistema de monitoreo de contaminación del aire con redes de sensores inalámbricos en tiempo real, el sistema realiza las mediciones en tiempo real de las concentraciones de CO₂, PM_{2.5} y temperatura dentro de un parqueadero y subterráneo y al aire libre. Estos valores son codificados y enviados a través del Wireless hacia un Raspberry PI para ser almacenados en una base de datos. El ofrecimiento de nuestro proyecto está basado en monitorear el ambiente del bebe utilizando varios sensores cabe mencionar que la tesis que antecede no proporciona alertas inmediatas del cambio de concentraciones ni de temperatura. ([Ortiz Espinosa, Diego](#))

2.2 MARCO TEÓRICO

2.2.1 Raspberry PI

Raspberry PI, nace en 2012 en el Reino Unido, el objetivo es animar a los niños a aprender informática en las escuelas, es un ordenador de bajo coste y tamaño reducido, tanto así que cabe en la palma de la mano, y puedes conectarlo un monitor y un teclado para interactuar con ella exactamente igual que cualquier otra computadora.

Tabla 1. Características de Raspberry PI.

SOC	BroadCom BCM2837
CPU	ARM11 ARM Cortex-A53
GPU	Dual Core VideoCore IV. Open GL ES 2.0
RAM	1 GB LPDDR2 SDRAM
USB 2.0	4
VIDEO	HDMI 1.4 @ 1920X1200 píxeles
AUDIO	Jack de 3,5mm de salida de audio, HDMI
MEMORIA	Micro-SD
ETHERNET	10/100 Mbps
Wifi	802.11 b/g/n LAN Inalámbrica y Bluetooth 4.1
TAMAÑO	85,60 x 56,5 mm
CONSUMO	5v, 900mA (dependiendo de la carga de trabajo de los 4 Cores)
COSTO	\$80 dólares americanos

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>



Figura 1. Raspberry PI Modelo B+

2.2.2 Cámara PI NoIR

La cámara Raspberry Pi NoIR no tiene filtro infrarrojo, lo que la hace perfecta para tomar fotografías infrarrojas o fotografiar objetos en condiciones de poca luz. La conexión de la cámara se realiza a través del cable que viene incluido en la cámara, la conexión del cable contiene 15 pines. La cámara nos brinda imágenes con una resolución de 5 megapíxeles o grabaciones a 1080p con una tasa de refresco de 30fps.

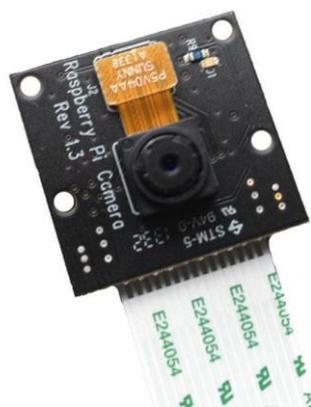


Figura 2. Cámara PI NoIR

Elaborado: José Huaipatin y Freddy Peñafiel

Las características del módulo son:

Tabla 2. Características cámara PI NoIR

Precio neto	\$25
Tamaño	Alrededor de 25 × 24 × 9 mm
Peso	3g
Resolución fija	5 megapíxeles
Modos de video	1080p30, 720p60 y 640 × 480p60 / 90
Integración de Linux	Controlador V4L2 disponible
API de programación en C	OpenMAX IL y otros disponibles
Sensor	OmniVision OV5647
Resolución del sensor	2592 × 1944 píxeles
Área de imagen del sensor	3.76 × 2.74 mm
Tamaño de píxel	1,4 μm × 1,4 μm
Tamaño óptico	1/4 "
Lente SLR de fotograma completo equivalente	35 mm
Relación S / N	36 dB
Gama dinámica	67 dB @ 8x ganancia
Sensibilidad	680 mV / lux-sec
Corriente oscura	16 mV / seg @ 60 C
Capacidad del pozo	4.3 Ke-
Foco fijo	1 m hasta el infinito
Longitud focal	3.60 mm +/- 0.01
Campo de visión horizontal	53.50 +/- 0.13 grados
Campo de visión vertical	41.41 +/- 0.11 grados
Relación focal (parada F)	2.9

Fuente: <https://www.raspberrypi.org/documentation/hardware/camera/>

2.2.3 Arduino Uno R3

Según la página Web *“Es una plataforma abierta que facilita la programación de un microcontrolador. Los microcontroladores nos rodean en nuestra vida diaria, usan los sensores para escuchar el mundo físico y los actuadores para interactuar con el mundo físico”*. (Qué es Arduino y Hardware Libre, 2016)

En referencia a la página web el proceso de alimentación eléctrica es *“Cuando se trabaja con Arduino normalmente lo alimentas a través del cable USB que va al ordenador. Sin embargo, una vez programado, o bien se deja conectado al ordenador o bien se busca otra forma alimentar Arduino. Alimentar Arduino por USB tiene la desventaja de que solo es capaz de suministrar hasta 500 mA”*. (Arduino, Alimentación Arduino, 2016)

Tabla 3. Características técnicas y funcionamiento Arduino Uno R3

Microcontrolador	ATmega328
Voltaje de funcionamiento	5V
Voltaje de entrada	7-12V
Voltaje de entrada (limites)	6-20V
Pasadores de entrada analogía	6
Memoria Flash	32KB
SRAM	2KB
EEPROM	1KB
Velocidad de reloj	16MHz

Fuente: <https://www.canakit.com/arduino-starter-kit.html>



Figura 3. Placa Arduino Uno R3

2.2.4 Módulo de Gas

Según la página Robotec MX *“El sensor de gas analógico MQ2 se utiliza en la detección de fugas de gas en muchos equipos de la actualidad y sobre todo en la industria, este sensor es adecuado para la detección de gas LP, i-butano, propano, metano, alcohol, hidrógeno, tiene una alta sensibilidad, un tiempo de respuesta rápido y dicha sensibilidad puede ser ajustada por el potenciómetro.*

Este pequeño sensor de gas detecta la presencia de gas combustible y humo en concentraciones de 300 a 10.000 ppm. Incorpora una sencilla interfaz de tensión analógica que únicamente requiere un pin de entrada analógica del microcontrolador. Con la conexión de cinco voltios en los pines el sensor se mantiene lo suficientemente caliente para que funcione correctamente”. (Robotec, Robotec MX, s.f.)

Según la página web de Hetpro *“La tarjeta del sensor cuenta con dos salidas de datos, una digital (DO) y otra analógica (AO). La salida digital manda una señal en estado alto cuando el sensor llega a un nivel deseado, el cual puede ser ajustado por medio del potenciómetro. La salida analógica va aumentando el valor del voltaje en proporción al nivel de gas que se detecta”.* (Torres Héctor, HetPro, 2014)

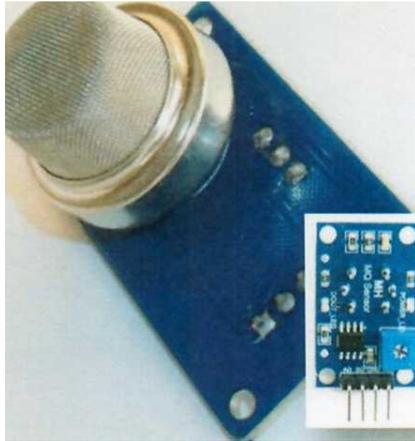


Figura 4. Sensor de Gas MQ2

Elaborado: José Huaipatin y Freddy Peñafiel

2.2.5 Sensor de Movimiento PIR-HC-SR501

En referencia a la página web “*El sensor HC-SR501, sólo funciona cuando alguien se mueve en la franja que puede barrer su detector. Puede detectar movimiento de 3 hasta 7 metros de distancia. Este sensor de movimiento PIR tiene 3 pines, VCC, OUTPUT y GND, 2 potenciómetros para ajustar la sensibilidad y la demora. El retardo se puede configurar entre 5 y 300 segundos mientras que el potenciómetro de sensibilidad ajusta el rango de detección de aproximadamente 3 metros a 7 metros*”. (Vicente, Sensor HC-SR501 con Arduino, 2017)

Las características son:

- *Sensor piroeléctrico (Pasivo) infrarrojo (También llamado PIR)*
- *El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización*
- *Rango de detección: 3 m a 7 m, ajustable mediante trimmer (Sx)*
- *Lente fresnel de 19 zonas, ángulo < 100°*
- *Salida activa alta a 3.3 V*

- *Tiempo en estado activo de la salida configurable mediante trimmer (Tx)*
- *Redisparo configurable mediante jumper de soldadura*
- *Consumo de corriente en reposo: $< 50 \mu A$*
- *Voltaje de alimentación: 4.5 VDC a 20 VDC*

(Sensor de movimiento PIR HC-SR501, s.f.)

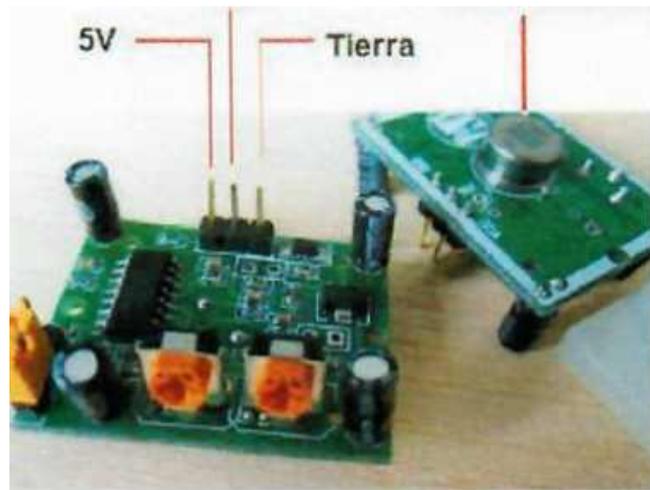


Figura 5.Sensor de Movimiento PIR

Fuente: <https://www.prometec.net/sensor-pir/>

Este sensor además incluye una cubierta plástica en forma de domo que actúa como un lente que permite que el sensor detecte con un ángulo de 180°.

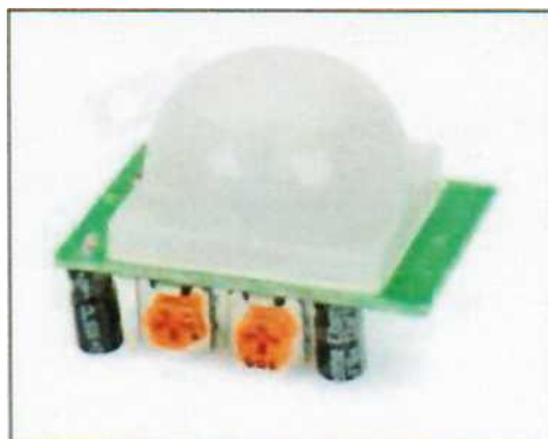


Figura 6. Sensor PIR HC-SR501

2.2.6 Módulo de Temperatura y Humedad

En una investigación realizada podemos decir que *“Este sensor trabaja con un rango de medición de temperatura de 0 a 50 °C con precisión de ± 2.0 °C y un rango de humedad de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura deben ser como mínimo 1 o 2 segundos.*

La resistencia Pull-Up puede ser un valor entre 4.7K y 10K. Si se desea trabajar con lógica de 3.3v solo hay que cambiar la alimentación a dicho voltaje al igual que la resistencia pull-up debe ir a 3.3V, en nuestro caso vamos a trabajar con el pin digital 2, pero pueden usar otro pin si lo desean”. (Tutorial sensor de temperatura y humedad DHT11 y DHT22, s.f.)

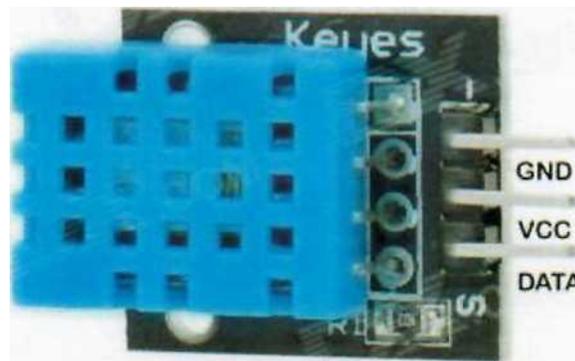


Figura 7. Módulo de Humedad y Temperatura

Fuente: <https://www.prometec.net/sensor-pir/#>

2.2.7 Servidor

En una investigación realizada para obtener concepto más sencillo es que *“Un servidor es un ordenador u otro tipo de equipo informático encargado de suministrar información a una serie de clientes, que pueden ser tanto personas como otros dispositivos conectados a él. La información que puede transmitir es múltiple y variada: desde archivos de texto, imagen o vídeo y hasta programas informáticos, bases de datos, etc”.* (Qué es un servidor y para qué sirve, 2016)

2.2.8 Servidor Web

En 2019, IONOS GUIDE DIGITAL establece que *“La tarea principal de un servidor web es la de guardar y organizar páginas web y entregarlas a clientes como navegadores web o crawlers. La comunicación entre servidor (software) y cliente se basa en HTTP, es decir, en el protocolo de transferencia de hipertexto o en HTTPS, la variante codificada. Por regla general, se transmiten documentos HTML y los elementos integrados en ellos, tales como imágenes, hojas de estilo o scripts. Los servidores web más populares son el servidor HTTP Apache, los servicios de Internet Information Server de Microsoft (ISS) o el servidor Nginx”*. (Tipos de servidores, 2019)

2.2.9 Lenguaje de Programación

En un estudio reciente de la página web nos dice que, *“Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.*

Un lenguaje de programación proporciona los elementos de lenguaje necesarios que son necesarios para traducir los pasos de un pseudocódigo en formato comprensible de la máquina. En otras palabras, el lenguaje de programación proporciona el puente para hacer la transición de pseudocódigo legible por humano instrucciones legibles por máquina. Las dos clasificaciones principales de lenguajes de programación son: bajo nivel de lenguaje de programación y lenguaje de programación de alto nivel”. (Gervacio, 2018)

2.2.10 Mensajería Instantánea

En referencia a la página web definiciónabc *“es un tipo de servicio que permite el intercambio de mensajes de texto que llegan en el mismo momento al destinatario.*

Pese a que hace años que estos servicios existen, ha sido con la explosión de los teléfonos móviles inteligentes y del gran éxito de WhatsApp, que dichos servicios se han popularizado, estallando una guerra por copar el mercado”. (DefiniciónABC tu diccionario hecho fácil, s.f.)

WhatsApp. – En 2018, colaboradores de Wikipedia dicen que el aplicativo WhatsApp *“Es una aplicación de mensajería para teléfonos inteligentes, que envía y recibe mensajes mediante Internet, complementando servicios de mensajería instantánea, servicio de mensajes cortos o sistema de mensajería multimedia. Además de utilizar la mensajería en modo texto, los usuarios de la libreta de contacto pueden crear grupos y enviarse mutuamente imágenes, vídeos y grabaciones de audio. Según datos de 2018, es líder en mensajería instantánea en gran parte del mundo, en el que supera los 1500 millones de usuarios, superando a otras aplicaciones como Facebook Messenger o Telegram, entre otros”.* (WhatsApp, 2018)

2.2.11 OPEN HARDWARE

Según la página web oshwa dice que *“Hardware de Fuentes Abiertas (OSHW en inglés) es aquel hardware cuyo diseño se hace disponible públicamente para que cualquier persona lo pueda estudiar, modificar, distribuir, materializar y vender, tanto el original como otros objetos basados en ese diseño. Las fuentes del hardware (entendidas como los ficheros fuente) habrán de estar disponibles en un formato apropiado para poder realizar modificaciones sobre ellas. Idealmente, el hardware de fuentes abiertas utiliza componentes y materiales de alta disponibilidad, procesos estandarizados, infraestructuras abiertas, contenidos sin restricciones, y herramientas de fuentes abiertas de cara a maximizar la habilidad de los individuos para materializar y usar el hardware. El hardware de fuentes abiertas da libertad de controlar la tecnología y al mismo tiempo compartir conocimientos*

y estimular la comercialización por medio del intercambio abierto de diseños”. (oshwa, 2016)

2.2.12 HARDWARE LIBRE

*“El proyecto surgió en el año 2010 durante el **Open Hardware Summit en Nueva York** donde desarrolladores, organizaciones y emprendedores relacionados con la cultura libre, se reunieron para discutir los modelos de negocios, norma, licencias y posibilidades del hardware abierto.*

Después de varios meses de colaboración a través de un foro y wiki, la gestora del proyecto (la investigadora libanesa Ayah Bdeir) en el mismo año anunciaba la Definición de Open Hardware 1.0, misma que aplica para artefactos, máquinas, dispositivos y cosas físicas que respetan la libertad de sus creadores de controlar su tecnología y al mismo tiempo compartir conocimiento y fomentar el comercio a través del intercambio abierto de diseños.

Esta definición está inspirada en la definición de Open Source de Bruce Perens y desarrolladores de Debian.

El grupo que redactó esta definición reconoce que el open-source es una de muchas formas de compartir información, por lo que dejan claro su apoyo y fomento a todas las demás formas de colaboración abierta y colaborativa, aun cuando no se ajusten a esta definición.

Sin embargo, la existencia de esta definición es un gran avance para todos aquellos interesados en trasladar la libertad y flexibilidad que ofrecen licencias como GPL y Creative Commons a productos físicos que los derechos de autor no cubren. Las patentes y el copyright siguen perdiendo terreno.

Impresoras 3d, hardware abierto, manufactura distribuida... otra revolución se acerca. Primero fueron los medios de distribución, ahora vamos por los de producción y manufactura. Esto es Open-Source Hardware y llega para quedarse.” (Juárez, ¿Que es el Open Hardware?, 2011)

2.2.13 OPEN SOURCE

“El Open Source o código abierto es un modelo de desarrollo de software que se basa en la colaboración abierta, es decir, se permite que el programa en cuestión sea modificado de manera abierta, con libertad de que los usuarios expertos en el tema puedan elegir cómo funcionará el programa y cambiar ciertas características de éste.

Muchos programas de código abierto son utilizados en el ámbito de la empresa como herramientas de trabajo, tan funcionales como otros de código cerrado. Para no confundirnos con el software libre, podemos indicar que en el código abierto se da más importancia a los beneficios prácticos que a las cuestiones éticas o de libertad. Es importante que mencionemos esto, pues muchas personas confunden el concepto de desarrollo de código abierto con el de software libre, cuando no son lo mismo.” (David, s.f.)

2.2.14 SOFTWARE LIBRE

En referencia a la página web concepto de software libre dice que: *“El término Software Libre se le atribuye al estadounidense Richard Stallman, quien fue fundador de la Free Software Foundation (Fundación del Software Libre) en la cual un número importante de computistas expertos aspiraba a desarrollar un Sistema Operativo totalmente libre, que llamaron Proyecto GNU (Para distinguirlo de Unix).*

Se denomina Software Libre a aquellos programas informáticos que le dan a sus usuarios, por decisión explícita de sus programadores y diseñadores, el acceso al código fuente o código de programación original en que fueron fabricados, para que pueda copiar, modificar, personalizar y distribuirlo libremente”. (Software Libre: Concepto, Libertades, Tipos y Ejemplos, 2018)

2.2.15 APLICACIÓN WEB

“Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (Hypertext Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones”. (Luján, 2002)

2.2.16 PROTOTIPO

Según la página web BioDic el concepto que más se asemeja al termino prototipo es *“El primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otros iguales, o molde original con el que se fabrica: por ejemplo, el prototipo de coche eléctrico. En medicina y ciencia del deporte, el ser humano hipotético empleado como modelo para examinar las proporciones, en especial entre los deportistas de élite. El muñeco unisex se define por la longitud del cuerpo, sus perímetros, anchuras y mediciones de los pliegues cutáneos, y tiene una estatura arbitraria de 1,7018 m y una masa corporal de 64,58 kg.” (prototipo)*

2.2.17 PYTHON

“Guido van Rossum escribió el lenguaje de programación Python a finales de los años ochenta y desde su primer lanzamiento en 1991 ha evolucionado hasta su versión más reciente (3.1.1), desarrollado como un proyecto de código abierto”. (maestrodeweb)

Las características de Python son las siguientes:

- *Lenguaje de Propósito general: significa que no esta orientado a un fin concreto, como puede ser PHP, pensando sobre todo para hacer páginas de internet*
- *Es multiparadigma: Pues, aunque su fuerte sea la programación orientada a objetos (es un lenguaje de alto nivel), existen otros paradigmas o estilos de programación para sus usuarios, como es la programación imperativa (con sentencias de bucle) o la programación funcional (con módulos y funciones).*

- *Multiplataforma: Al contrario que muchos lenguajes como visual basic, que principalmente solo puedes hacer cosas para Windows, con Python tienes la posibilidad de usarlo en muchos dispositivos y sistemas operativos, ya que se han creado intérpretes para Unix, Linux, Windows y sistemas Mac Os.*
- *Es de tipado dinámico: usando declaramos una variable, no es necesario decirle de que tipos son los datos (si es int, string, float, etc.). La variable se adapta a lo que escribimos cuando se ejecuta el programa.*

Antes esta característica siempre ha sido criticada en otros lenguajes, por la optimización de la memoria, errores a la hora de escribir código, etc. pero con Python el objetivo es que el lenguaje ayude a la creación de software, no tener que lidiar con peculiaridades propias del lenguaje. (lenguajeprogramacion)

CAPÍTULO 3

DESCRIPCIÓN DEL PROYECTO TÉCNICO

Este proyecto pretende proporcionar un prototipo para monitorear el ambiente de infantes, proporcionando el constante monitoreo en la habitación, y se utilizara la tecnología para el servicio de adultos, para el cuidado de infantes para la prevención de accidentes.

Actualmente las investigaciones que han propuesto el monitoreo de ambiente para bebés que son a través de intercomunicador o cámaras son usadas en diferentes sitios tales como: guarderías y centros educativos, además existen sensores que permiten la vigilancia del infante en las maternidades para el control de las temperaturas de los recién nacidos en las termo cunas, sin embargo en nuestro país a la fecha no cuenta con un prototipo que integre tecnología libre que colabore con el cuidado de niños en el hogar.

La propuesta actual es factible por el creciente desarrollo de tecnología, en el mercado se puede encontrar herramientas de hardware y software a bajo costo, que permitirá efectuar un prototipo para evitar enfermedades o incidentes a través de un monitoreo.

El prototipo de monitoreo bebés, cumplirá con pruebas para establecer los motivos primordiales de los accidentes que tienen los niños dentro del hogar.

3.1 TITULO Y DESCRIPCIÓN DEL PROYECTO

Diseñar e implementar un prototipo para monitoreo de un ambiente de bebés utilizando Raspberry PI.

3.2 OBJETIVO GENERAL Y ESPECÍFICO

3.2.1 Objetivo General

- Diseñar e Implementar un prototipo de monitoreo de un ambiente de bebés utilizando Raspberry.

3.2.2 Objetivo Específico

- Estudio de los diferentes monitoreos del ambiente de bebés utilizando la tecnología actual.
- Analizar los diferentes tipos de componentes para un ambiente de bebés con el uso de Raspberry PI.
- Desarrollar e implementar la aplicación con la ayuda de la web que permita visualizar en tiempo real el ambiente del infante.
- Capacitación a los usuarios para el manejo de este tipo de sistema.

3.3 DESARROLLO

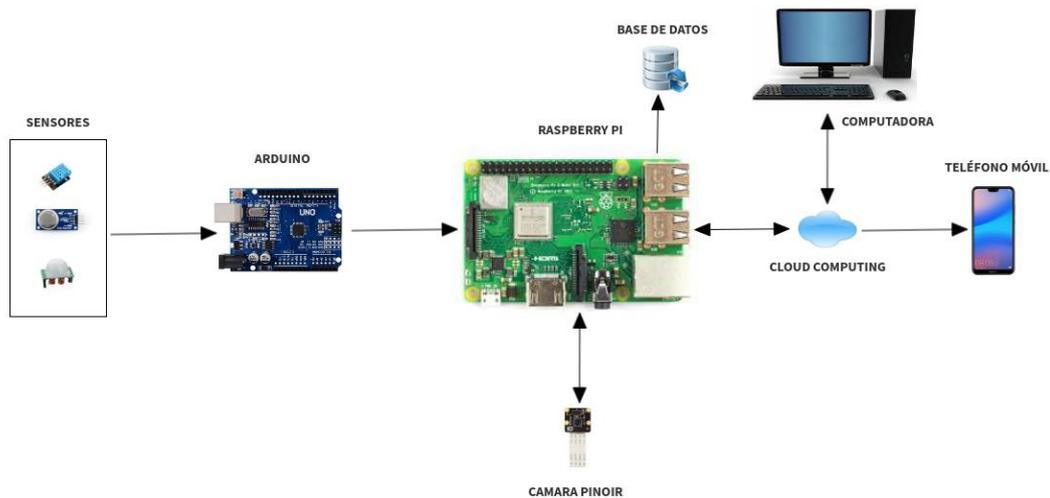


Figura 8. Figura Esquema General del funcionamiento del prototipo

Elaborado: José Huaipatin y Freddy Peñafiel

El bosquejo general del prototipo nos muestra que la información de los sensores adquiridos a través del Raspberry PI serán transmitidos en tiempo real hacia la base de datos, esta plataforma se encargará de almacenar los datos recibidos. Además, se muestra que lo que recibirá el usuario en su Smartphone serán las siguientes notificaciones: alerta de cambio de temperatura, alerta de gases nocivos y alerta de humedad.

3.4.1 DISEÑO DE HARDWARE

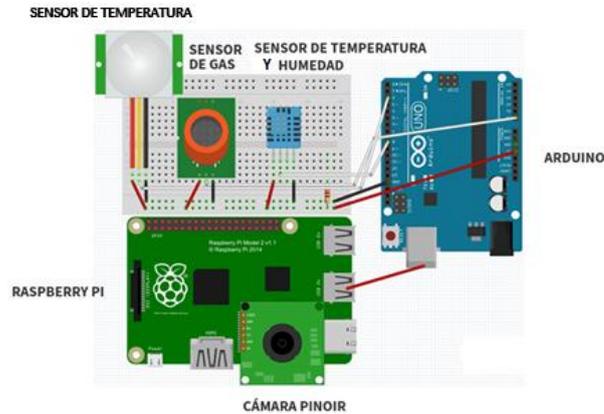


Figura 9. Figura Esquema del circuito lógico

Elaborado: José Huaipatin y Freddy Peñafiel

En este esquema se observa los requerimientos, características y especificaciones del prototipo, se muestra un bosquejo que exponga el funcionamiento del prototipo. Para que el prototipo funcione es la incorporación de los componentes en el Raspberry Pi, utilizando el programa “Fritzing” se realizó el diagrama esquemático del prototipo, en la cual se muestra cómo debería conectarse los sensores y la cámara en el Raspberry Pi.

3.4.2 CONEXIÓN DE SENSORES

3.4.2.1 Sensor de Temperatura y Humedad DHT11



Figura 10. Figura Sensor de temperatura y humedad

“El DHT11 es un sensor que mide humedad y temperatura. Es ideal para sistemas de medición climatológicos o para controles de temperatura y humedad. Este sensor además incluye un transductor interno de temperatura del tipo NTC. También el módulo tiene una gran relación señal a ruido ante la interferencia. Por ejemplo cada circuito, se calibra estrictamente en el laboratorio. Esto permite que sea extremadamente preciso en calibración de humedad. Los coeficientes de calibración se almacenan como programas en la memoria OTP, que son utilizados por el proceso de detección de la señal interna del sensor”. (Eduardo, 2015)

3.4.2.1.1 Diagrama de conexión el Sensor de Temperatura y Humedad DHT11 con Arduino

A través de la página web de Geek Factory muestra cómo debe estar conectado el diagrama. “El siguiente esquema ilustra como debe realizarse la conexión del sensor DHT11 o DHT22 con Arduino. Como podemos ver la conexión es bastante simple y se realiza de la misma forma para el DHT11 y DHT22. Solamente se requiere de un componente externo para la comunicación y es una simple resistencia de 10K. También podemos agregar un cerámico de 100nF cerca de los pines de alimentación del DHT11 para ayudar a reducir los ruidos que puedan filtrarse en la alimentación”. (DHT11 con Arduino sensor temperatura y humedad - Geek Factory, 2018)

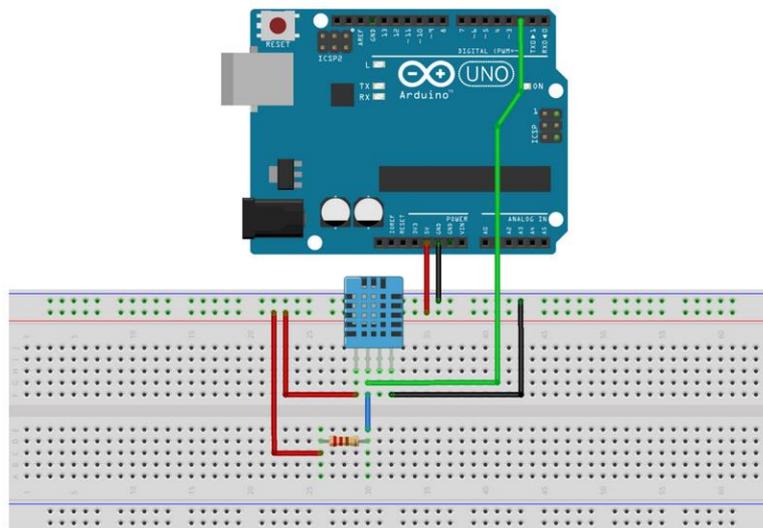


Figura 11. Figura Conexión de sensor de temperatura

Fuente: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/dht11-con-arduino/>

Los pines de la versión con PCB del DHT11 son:

- GND: conexión con tierra
- DATA: transmisión de datos
- VCC: alimentación

3.4.2.1.2 Configuración del Sensor de Temperatura y Humedad DHT11

Para realizar la configuración del sensor debemos conectar el Arduino con el Raspberry. (Los pasos para iniciar el Raspberry Pi revisar la página 41 del documento).

Dentro del Raspberry procedemos dar clic en inicio y ejecutar el aplicativo Arduino IDE.



Figura 12. Ejecutar Arduino IDE

Elaborado: José Huaipatin y Freddy Peñafiel

Mostrará la consola del Arduino donde se ingresará la configuración del sensor.

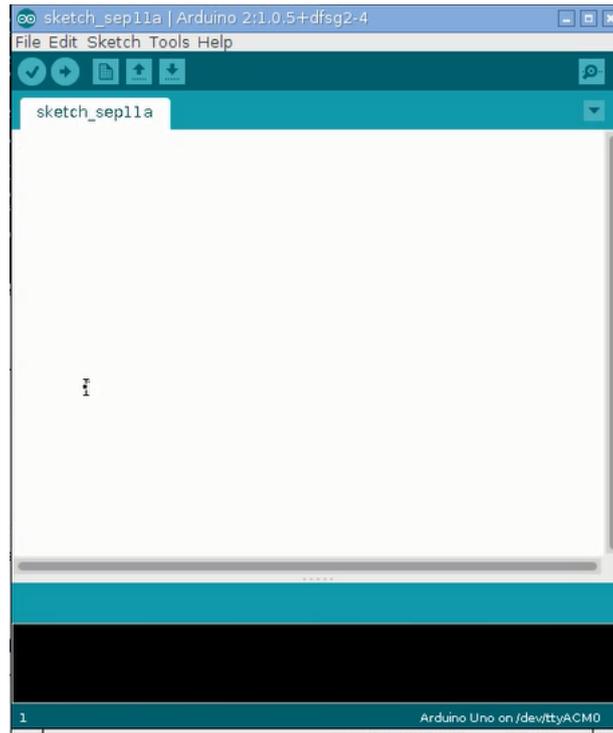


Figura 13. Consola de Arduino IDE

Elaborado: José Huaipatin y Freddy Peñafiel

```
#include "dht11.h"
int DHT11::read(int pin)
{
    uint8_t bits[5];
    uint8_t cnt = 7;
    uint8_t idx = 0;
    for (int i=0; i< 5; i++) bits[i] = 0;
    pinMode(pin, OUTPUT);
    digitalWrite(pin, LOW);
    delay(18);
    digitalWrite(pin, HIGH);
    delayMicroseconds(40);
    pinMode(pin, INPUT);
    unsigned int loopCnt = 10000;
    while(digitalRead(pin) == LOW)
        if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
```

```

loopCnt = 10000;
while(digitalRead(pin) == HIGH)
    if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
for (int i=0; i<40; i++)
{
    loopCnt = 10000;
    while(digitalRead(pin) == LOW)
        if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
    unsigned long t = micros();
    loopCnt = 10000;
    while(digitalRead(pin) == HIGH)
        if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
    if ((micros() - t) > 40) bits[idx] |= (1 << cnt);
    if (cnt == 0) // next byte?
    {
        cnt = 7; // restart at MSB
        idx++; // next byte!
    }
    else cnt--;
}
omitted in formulas.
humidity = bits[0];
temperature = bits[2];
uint8_t sum = bits[0] + bits[2];
if (bits[4] != sum) return DHTLIB_ERROR_CHECKSUM;
return DHTLIB_OK;
}

```

3.4.2.2 Sensor de Gas MQ2

El portal web dice que “Nuestro sensor viene en un módulo para una integración fácil con cualquier placa (nosotros usaremos un Arduino UNO como es costumbre). En especial, el MQ-2 puede detectar: gas natural, butano, propano, metano, alcohol, hidrógeno y humo”. (Daniel, 2017)

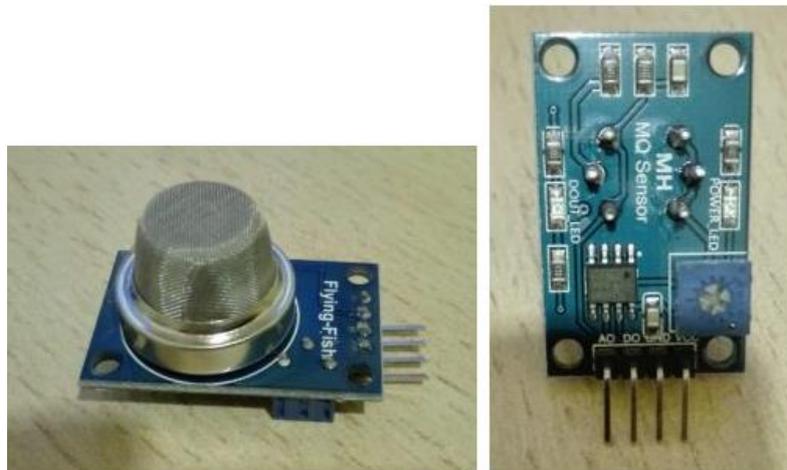


Figura 14. Figura esquema de montaje sensor de gas

Fuente: <https://www.arduinohobby.com/deteccion-de-gas-con-arduino/>

3.4.2.2.1 Esquema de montaje del Sensor de Gas MQ2

Para realizar la conexión eléctrica del módulo debemos tener conectado GND y 5V con los pines proporcionados con el sensor.

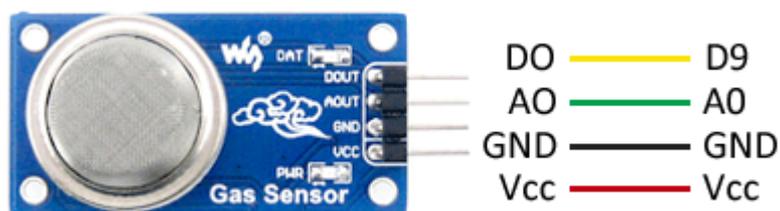


Figura 15. Esquema de montaje módulo de gas

Fuente: <https://www.luisllamas.es/arduino-detector-gas-mq/>

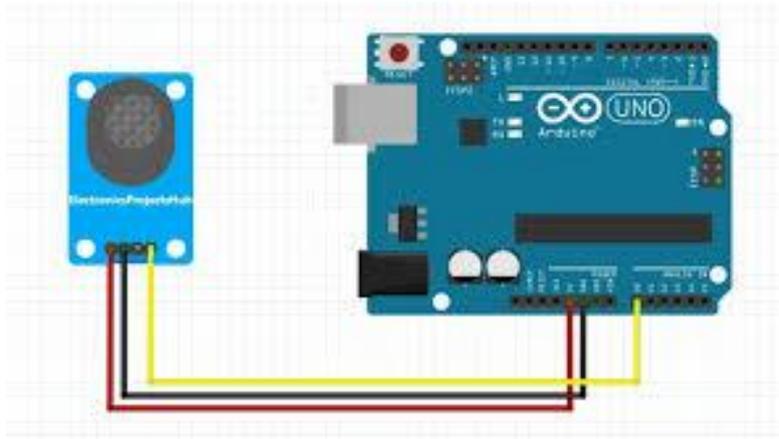


Figura 16. Figura esquema del sensor de gas con el Arduino

Fuente: <https://create.arduino.cc/projecthub/Junezriyaz/how-to-connect-mq2-gas-sensor-to-arduino-f6a456>

3.4.2.2.2 Configuración del Sensor de Gas MQ2

Para definir los parámetros de configuración del sensor, conectamos el Arduino con el minicomputador Raspberry, encender e iniciar el Raspberry diríjase a la página 41 del documento

Una vez iniciado el Raspberry damos clic en inicio y posterior a eso ejecutamos Arduino IDE



Figura 17. Iniciar la consola de Arduino IDE

Elaborado: José Huaipatin y Freddy Peñafiel

Dentro de la Consola de configuración del Arduino se registra el siguiente código de datos

```
#include "Arduino.h"
#include "MQ2.h"
MQ2::MQ2(int pin) {
  _pin = pin;
}
void MQ2::begin(){
  Ro = MQCalibration();
  Serial.print("Ro: ");
  Serial.print(Ro);
  Serial.println(" kohm");
}
float* MQ2::read(bool print){
  lpg = MQGetGasPercentage(MQRead()/Ro,GAS_LPG);
  co = MQGetGasPercentage(MQRead()/Ro,GAS_CO);
  smoke = MQGetGasPercentage(MQRead()/Ro,GAS_SMOKE);

  if (print){
    Serial.print("LPG:");
    Serial.print(lpg);
    Serial.print( "ppm" );
    Serial.print("  ");
    Serial.print("CO:");
    Serial.print(co);
    Serial.print( "ppm" );
    Serial.print("  ");
    Serial.print("SMOKE:");
    Serial.print(smoke);
    Serial.print( "ppm" );
    Serial.print("\n");
  }
  lastReadTime = millis();
  static float values[3] = {lpg,co,smoke};
```

```

    return values;
}
float MQ2::readLPG(){
    if (millis() < (lastReadTime + 10000) && lpg != 0){
        return lpg;
    }else{
        return lpg = MQGetGasPercentage(MQRead()/10,GAS_LPG);
    }
}
float MQ2::readCO(){
    if (millis() < (lastReadTime + 10000) && co != 0){
        return co;
    }else{
        return co = MQGetGasPercentage(MQRead()/10,GAS_CO);
    }
}
float MQ2::readSmoke(){
    if (millis() < (lastReadTime + 10000) && smoke != 0){
        return smoke;
    }else{
        return smoke = MQGetGasPercentage(MQRead()/10,GAS_SMOKE);
    }
}
float MQ2::MQResistanceCalculation(int raw_adc) {
    return (((float)RL_VALUE*(1023-raw_adc)/raw_adc));
}
float MQ2::MQCalibration() {
    float val=0;
    for (int i=0;i<CALIBARAION_SAMPLE_TIMES;i++) { //take multiple samples
        val += MQResistanceCalculation(analogRead(_pin));
        delay(CALIBRATION_SAMPLE_INTERVAL);
    }
    val = val/CALIBARAION_SAMPLE_TIMES; //calculate the average value
}

```

```

    val = val/RO_CLEAN_AIR_FACTOR;           //divided by
RO_CLEAN_AIR_FACTOR yields the Ro
//according to the chart in the datasheet
    return val;
}
float MQ2::MQRead() {
    int i;
    float rs=0;
    int val = analogRead(_pin);
    for (i=0;i<READ_SAMPLE_TIMES;i++) {
        rs += MQResistanceCalculation(val);
        delay(READ_SAMPLE_INTERVAL);
    }
    rs = rs/READ_SAMPLE_TIMES;
    return rs;
}
float MQ2::MQGetGasPercentage(float rs_ro_ratio, int gas_id) {
    if ( gas_id == GAS_LPG ) {
        return MQGetPercentage(rs_ro_ratio,LPGCurve);
    } else if ( gas_id == GAS_CO ) {
        return MQGetPercentage(rs_ro_ratio,COCurve);
    } else if ( gas_id == GAS_SMOKE ) {
        return MQGetPercentage(rs_ro_ratio,SmokeCurve);
    }
    return 0;
}
int MQ2::MQGetPercentage(float rs_ro_ratio, float *pcurve) {
    return (pow(10,(((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0])));
}

#ifndef MQ2_h
#define MQ2_h
#if ARDUINO >= 100

```

```

#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#endif
class MQ2 {
public:
    MQ2(int pin);
    float* read(bool print);
    float readLPG();
    float readCO();
    float readSmoke();
    void begin();
private:
    int _pin;
    int RL_VALUE = 5;    //define the load resistance on the board, in kilo ohms
    int RO_CLEAN_AIR_FACTOR = 9.83;
    int CALIBRATION_SAMPLE_TIMES = 5;
    int CALIBRATION_SAMPLE_INTERVAL = 50;
    int READ_SAMPLE_INTERVAL = 50;
    int READ_SAMPLE_TIMES = 5;
    float LPGCurve[3] = {2.3,0.21,-0.47};
    float COCurve[3] = {2.3,0.72,-0.34};
    float SmokeCurve[3] = {2.3,0.53,-0.44};
    float Ro = 10;
    int GAS_LPG = 0;
    int GAS_CO = 1;
    int GAS_SMOKE = 2;
    float lpg = 0;
    float co = 0;
    float smoke = 0;
    float MQRead();
    float MQGetGasPercentage(float rs_ro_ratio, int gas_id);

```

```

int MQGetPercentage(float rs_ro_ratio, float *pcurve);
float MQCalibration();
float MQResistanceCalculation(int raw_adc);
int lastReadTime = 0;
};
//#endif

```

3.4.2.3 Sensor de Movimiento PIR HC-SR 501

La página web dice que “Los PIR más frecuentes son sensores de movimiento, y para ello están divididos en dos mitades de forma que detecten el cambio de radiación IR que reciben uno y otro lado, disparando la alarma cuando perciben ese cambio”. (designthemes, s.f.)

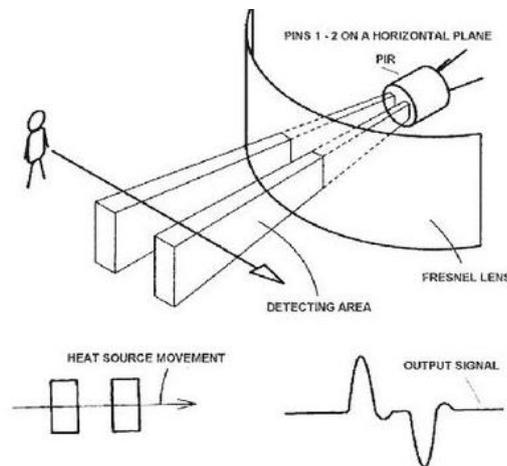


Figura 18. Figura detección de movimiento

Fuente <https://www.prometec.net/sensor-pir/>

“Lo normal además es que estos sensores se recubran con pequeñas lentes de plástico que mejoren su ángulo de detección”. (designthemes, s.f.)



Figura 19. Figura sensor de movimiento

Fuente <https://www.prometec.net/sensor-pir/>

3.4.2.3.1 Esquema eléctrico del Sensor de Movimiento PIR HC-SR 501

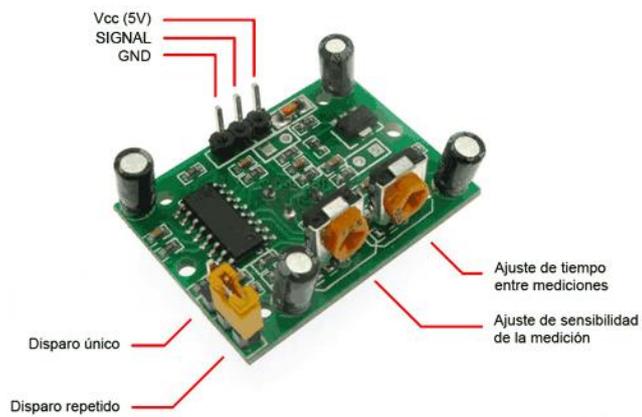


Figura 20. Figura Esquema de patillaje de un sensor PIR

Fuente: <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

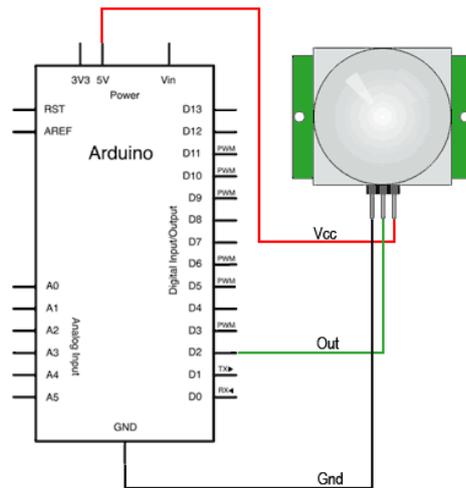


Figura 21. Figura esquema eléctrico

Fuente: <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

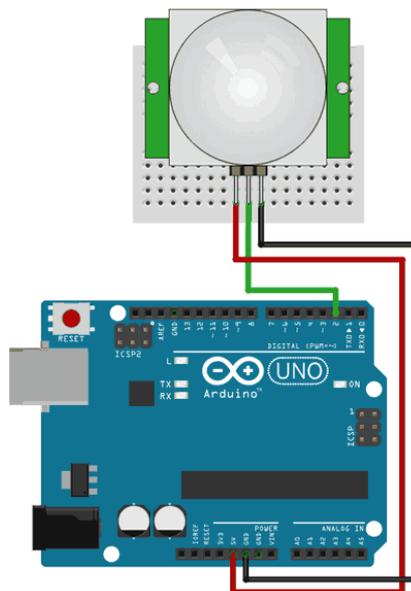


Figura 22. Figura esquema de conexión de sensor de movimiento con el Arduino

Fuente: <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

3.4.2.3.2 Configuración de Sensor de Movimiento PIR HC-SR 501

Para realizar la configuración del sensor debemos conectar el esquema eléctrico del sensor de movimiento con el Raspberry PI.

Dentro del Sistema Operativo del minicomputador buscamos el programa Arduino IDE



Figura 23. Programa Arduino IDE

Elaborado: José Huaipatin y Freddy Peñafiel

Una vez abierto la consola del Arduino IDE dentro del Raspberry PI, agregamos el siguiente código fuente para configurar el sensor de movimiento.

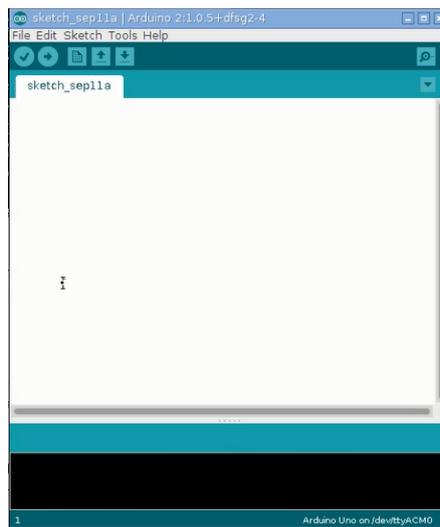


Figura 24. Programa de Arduino IDE

Elaborado: José Huaipatin y Freddy Peñafiel

```

long unsigned int lowIn;
long unsigned int pause = 5000;
boolean locklow = true;
boolean takeLowTime;

int pirPin = 2;
int MD = 0;
int MF = 0;

void setup()
{
  pinMode (pirPin , INPUT);
}

void loop()
{ if (digitalRead(pirPin) == HIGH) {
  If (lockLow){
    Locklow = false;
    Serial.print("|| Movimiento Detectado ||");
    Serial.print(1);
    MD = millis()/1000;
    Delay(50);
  }
  takeLowTime = true;
}
if (digitalRead(pirPin) == LOW){
  If (takeLowTime){
    lowIn = millis();
    takeLowTime = false;
  }
  If (!Locklow && millis() - lowIn > pause){
    locklow = true;
    MF = ((millis-pause)/1000) - MD;
  }
}
}

```

```

Serial.print(1);
Serial.print("|| Duración de Movimiento en segundos ||");
Serial.print(MF);
Serial.print("\t");
delay 50;
}
}
if (MF > 1){
    MF=0;
}
Serial print ("\n");
Delay (1000);

```

3.4.2.4 Módulo Cámara PiNoir

La conexión entre la cámara y Raspberry es a través del puerto J3.

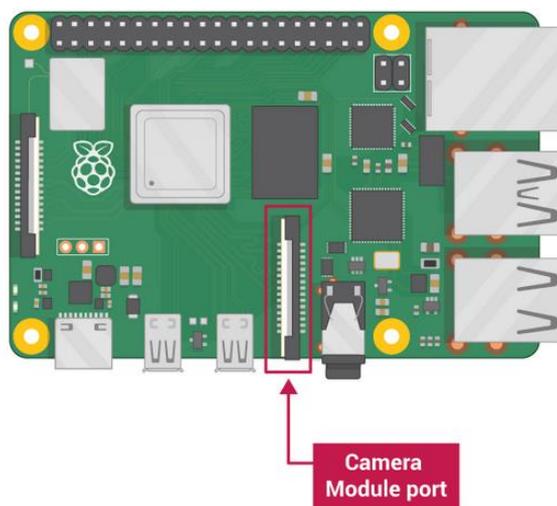


Figura 25. Figura Raspberry PI

Fuente: <https://hardzone.es/2018/08/25/como-configurar-modulo-camara-raspberry-pi-3/>

- Después de conectar los componentes correctamente, es el momento de encender la Raspberry Pi. Una vez se haya terminado de encender debemos dirigirnos a la pantalla de Configuración de la Raspberry Pi. En la interfaz de Raspbian se encuentra en el menú de Preferencias.
 - Otra forma de acceder es mediante el comando: `sudo raspi-config`
- Debemos acceder a la configuración, dirígete a las Opciones de Interfaz y selecciona Cámara. En la ventana emergente selecciona Sí, ahora sólo debes esperar a que se reinicie el dispositivo.

Cómo tomar fotos en la Raspberry Pi con el módulo de cámara

Una vez se haya finalizado el reinicio de la Raspberry Pi ya puedes tomar una foto. Para ello escribe en la ventana de comandos: `raspistill -o image.jpg`

Esta foto de prueba se almacena en el directorio Imágenes en el escritorio de la Raspberry Pi. Este no es el único comando que podemos hacer, por ejemplo, podemos usar `-vf` o `-hf` para cambiar la rotación de la imagen. Si quieres conocer todos los comandos de la cámara utiliza `raspistill` en la ventana de comandos.

Cómo grabar vídeos en la Raspberry Pi

Para grabar vídeos debemos aplicar este comando: `raspivid -o video.h264`

Si añades detrás de ese comando “`-t X`” siendo X el tiempo en milisegundos que quieres que grabe, el vídeo parará de forma automática al transcurrir el tiempo.



Figura 26. Figura Camara conectado al Raspberry PI

3.4.3 DESARROLLO DE SOFTWARE

3.4.1.2.1 Instalación del Sistema Operativo

Para realizar la instalación del S.O. del Raspberry diríjase a la sección de Anexo (Anexo A Instalación del Sistema Operativo).

3.4.1.2.2 Instalación del servidor VNC en nuestra Raspberry Pi

Para iniciar la configuración del servidor VNC en Raspberry Pi realizamos lo siguiente:

1.1. Actualizamos la Raspberry Pi.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

1.2. Instalamos el server VNC.

```
sudo apt-get install tightvncserver
```

1.3. Ejecutamos el programa, seguidamente ponemos el password.

```
Tightvncserver
```

1.4. Una vez que hemos instalado el servidor reiniciamos la Raspberry Pi.

```
sudo reboot
```

1.5. Ejecutamos el servidor VNC.

```
vncserver :0
```

Luego podremos acceder a la Raspberry a través de la IP o Hostname.

3.4.1.2.3 Instalación de Apache

Ejecutamos la consola Terminal de Raspberry PI e ingresamos el siguiente código

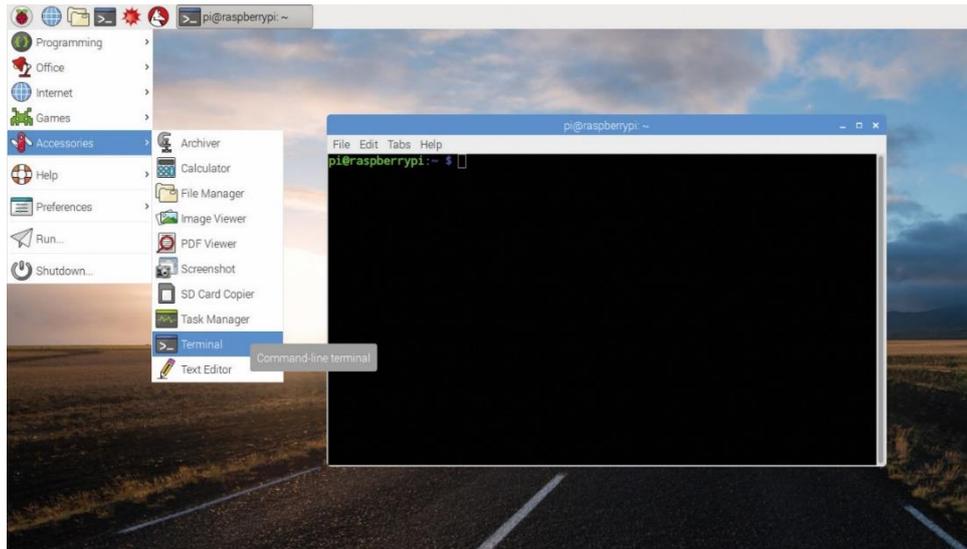


Figura 27. Terminal de Raspberry PI

Elaborado: José Huaipatin y Freddy Peñafiel

```
sudo apt install apache2 --instalación de apache
sudo chown -R pi:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/
wget -O check_apache.html http://192.168.100.24
cat ./check_apache.html
sudo apt install php php-mbstring --instalación de php
sudo rm /var/www/html/index.html
echo "<?php phpinfo ();?>" > /var/www/html/index.php
```

Una vez finalizado, vemos que se ha creado un directorio en /var/www donde se ubicaran los archivos HTML y PHP del servidor. Por defecto, hay un archivo llamado index.php. Si ponemos en un navegador la dirección IP de la Raspberry Pi veremos una página con la frase ;It works! Esto quiere decir que el servidor se ha instalado correctamente.

3.4.1.2.4 Instalación de Python

Abrir el Terminal de Raspberry PI.

```
sudo apt-get installpython -dateutil
```

```
sudo apt install phpmyadmin
```

```
root@raspberrypi:~# sudo phpenmod mysqli
```

```
root@raspberrypi:~# sudo /etc/init.d/apache2 restart
```

```
[ ok ] Restarting apache2 (via systemctl): apache2.service.
```

```
root@raspberrypi:~# sudo ln -s /usr/share/phpmyadmin/var/www/html/phpmyadmin
```

```
root@raspberrypi:~# sudo apt-get install ssmtp
```

```
root@raspberrypi:~# sudo apt-get install mailutils
```

```
sudo apt-get install python3-dev libffi-dev libssl-dev -y
```

```
wget https://www.python.org/ftp/python/3.6.3/Python-3.6.3.tar.xz
```

```
tar xJf Python-3.6.3.tar.xz
```

```
cd Python-3.6.3
```

```
./configure
```

```
make
```

```
sudo make install
```

```
sudo pip3 install --upgrade pip
```

3.4.1.2.5 Instalación del Arduino IDE

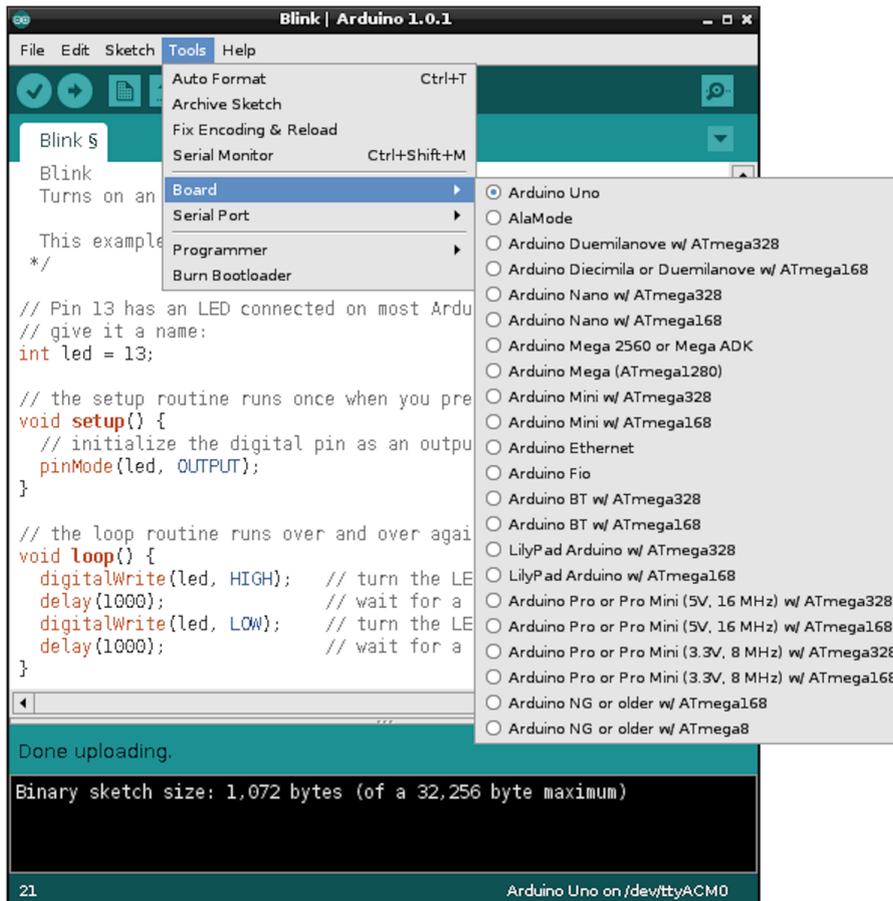
Abrir el Terminal de Raspberry PI.

Para instalarlo utilice los siguientes comandos:

```
$ Sudo update apt-get
```

```
$ Sudo apt-get install Arduino-core
```

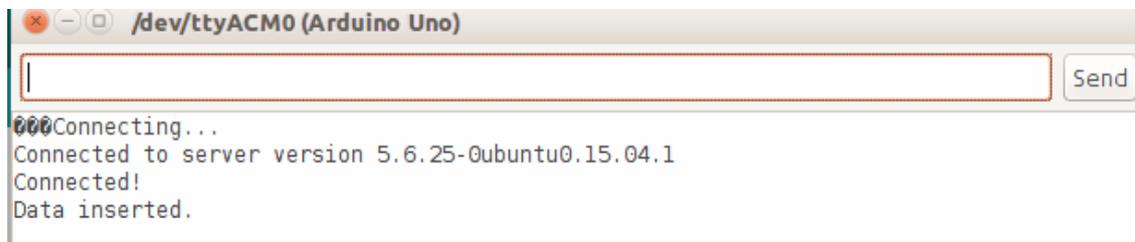
Después de la instalación, debemos elegir el BOARD Arduino UNO.



Conectar el arduino a la red y realizar la prueba de conexión.

Al conectar el Arduino al puerto USB y antes de empezar a programar o acceder a la consola serial es indispensable modificar los atributos de acceso al puerto, digitando el siguiente comando:

```
sudo chmod 777 /dev/ttyACM0
```



3.4.1.2.6 Instalación de PHP

Abrir el Terminal de Raspberry PI.

Procedemos a instalar el lenguaje PHP en la Raspberry Pi:

```
sudo apt install phpmyadmin
```

Después instalaremos sus paquetes:

```
sudo phpenmod mysqli
```

```
sudo /etc/init.d/apache2 restart
```

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

3.4.1.2.7 Instalación de MySQL

Abrimos el Terminal desde el Raspberry para realizar la instalación de la base de datos

Instalación

Para su instalación escribimos lo siguiente:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Nos pedirá que pongamos una contraseña que nosotros queramos. Al terminar la instalación, debemos iniciar el MySQL:

```
sudo service mysql start
```

Para comprobar si se ha instalado correctamente, tenemos que escribir el siguiente comando poniendo la contraseña que hemos establecido antes en la instalación de MySQL después de la letra p:

```
mysql -uroot -Pcontraseña
```

Si aparece el comando *mysql>* quiere decir que funciona correctamente. Para salir pulsamos CTRL+C.

3.4.1.2.8 Instalación de Streaming

Abrir el Terminal de Raspberry PI.

```
# Update & Install Tools
```

```
sudo apt-get update
```

```
sudo apt-get upgrade -y
```

```
sudo apt-get install build-essential libjpeg8-dev imagemagick libv4l-dev cmake -y
```

```
# Clone Repo in /tmp
```

```
cd /tmp
```

```
git clone https://github.com/jacksonliam/mjpg-streamer.git
```

```
cd mjpg-streamer/mjpg-streamer-experimental
```

```
# Make
```

```
make
```

```
sudo make install
```

```
# Run
```

```
/usr/local/bin/mjpg_streamer -i "input_uvc.so -r 1280x720 -d /dev/video0 -f 30 -q 80" -o  
"output_http.so -p 8080 -w /usr/local/share/mjpg-streamer/www"
```

3.4.1.2.9 Instalación MJPG Streamer

Iniciar el Raspberry Pi e instale MJPG Streamer.

Ejecutamos el siguiente script

```
nano /home/pi/mjpg-streamer.sh
```

El código fuente de la configuración de MJPG Streamer está en la sección de Anexos.
(Anexo B Instalación de MJPG Streamer)

3.4.1.2.10 Instalación de PiJuice

Es una plataforma de comunicación entre Twilio y WhatsApp

Instalación de PiJuice Software

Para instalar la plataforma, abra una ventana de Terminal en su Raspberry Pi o inicie sesión a través de SSH e ingrese el siguiente comando:

```
sudo apt-get update
```

```
sudo apt-get install pjuice-base
```

3.4.1.2.11 Instalación de Twilio

Para realizar la notificación por WhatsApp se utilizará la plataforma de comunicaciones en nube “Twilio” que permite recibir llamadas entrantes y hacer llamadas salientes, y puede enviar y recibir mensajes multimedia desde y hacia números. (Ver Anexo C Instalación de Twilio)

3.4.1.2.12 Instalación de NoIP

Utilización de plataforma gratuita llamada “no-ip” para Raspberry Pi que sirve para poder acceder a la Raspberry desde cualquier sitio que tenga internet. Revisar el Anexo D Instalación y Configuración de NoIP

3.4.4 PRUEBAS DE INSTALACIÓN DEL PROTOTIPO



Figura 28. Funcionamiento del prototipo para el monitoreo del ambiente de bebé

Elaborado: José Huaipatin y Freddy Peñafiel



Figura 29. Funcionamiento del prototipo para el monitoreo del ambiente de bebé

Elaborado: José Huaipatin y Freddy Peñafiel

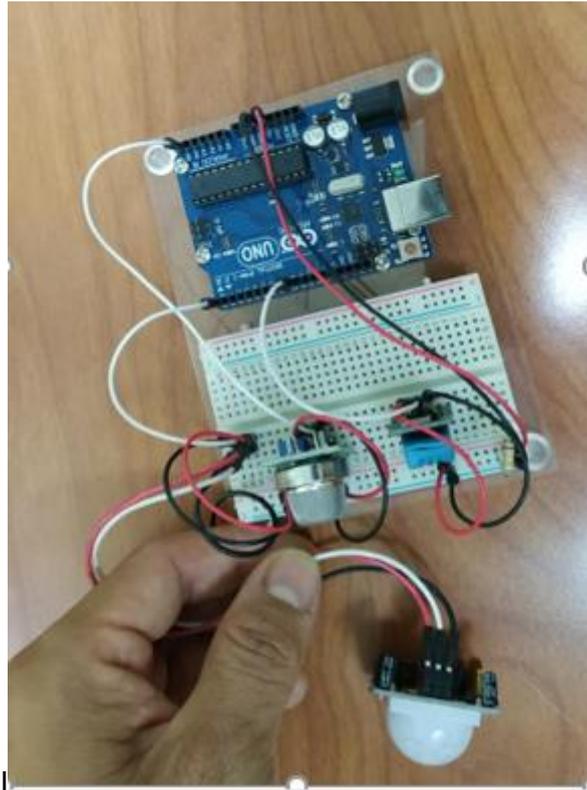


Figura 30. Conexión de sensores

Elaborado: José Huaipatin y Freddy Peñafiel

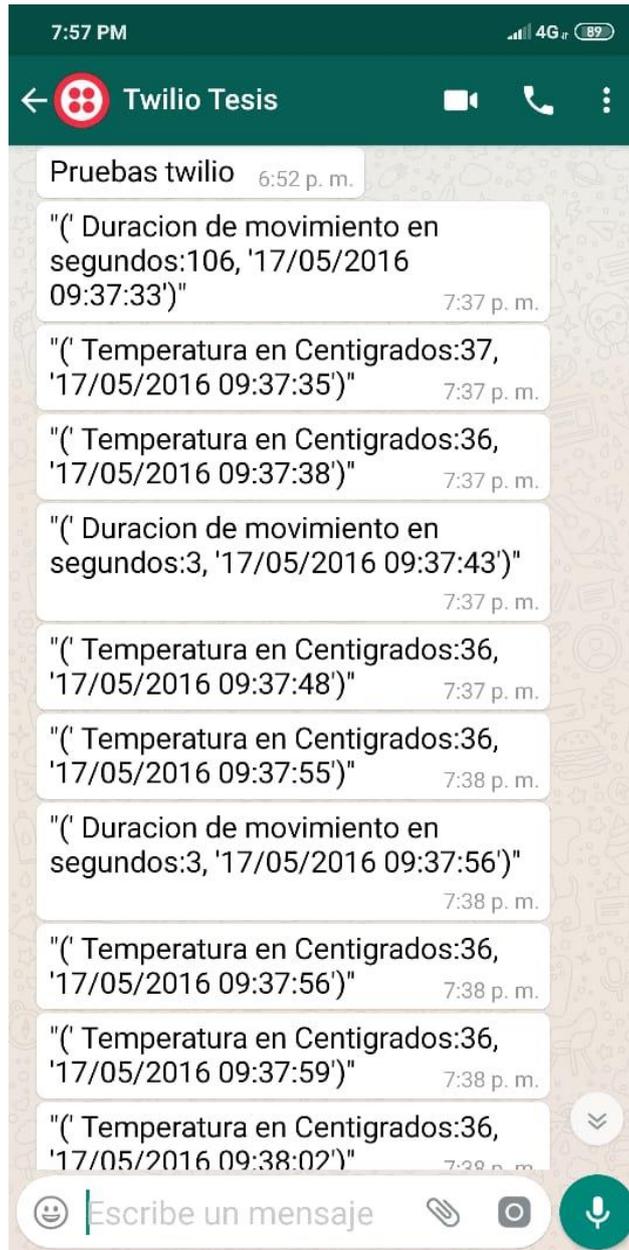
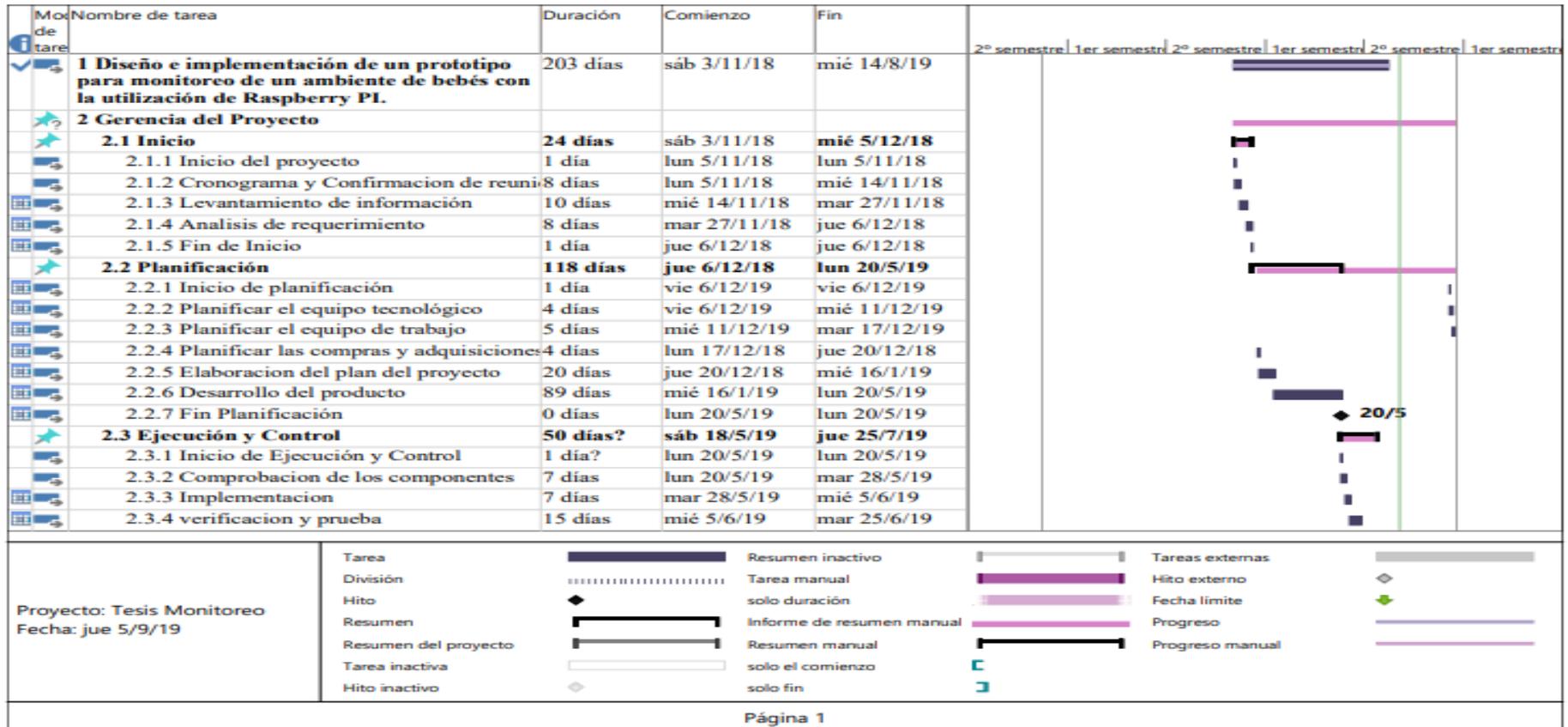


Figura 31. Notificación de las alertas

Elaborado: José Huaipatin y Freddy Peñafiel

3.4 CRONOGRAMAS DE ACTIVIDADES

En este diagrama de tiempo identificaremos las actividades específicas y generales del proceso del desarrollo del proyecto en función del tiempo, que obedece a los objetivos planteados para llevar acabo la implementación del proyecto.



Mo de tare	Nombre de tarea	Duración	Comienzo	Fin	2º semestre	1er semestre	2º semestre	1er semestre	2º semestre	1er semestre
	2.3.5 Ajustes y mejoras	7 días	mar 16/7/19	mié 24/7/19						
	2.3.6 Implementacion del dispositivo	3 días	mié 24/7/19	vie 26/7/19						
	2.3.7 Fin de Ejecución y Control									
	2.4 Cierre	14 días	vie 26/7/19	mié 14/8/19						
	2.4.1 Inicio de Cierre	7 días	vie 26/7/19	lun 5/8/19						
	2.4.2 Presentacion del dispositivo	5 días	lun 5/8/19	vie 9/8/19						
	2.4.3 Entrega de los manuales del usuario	4 días	vie 9/8/19	mié 14/8/19						
	2.4.4 Fin de Cierre	1 día	mié 14/8/19	mié 14/8/19						

Proyecto: Tesis Monitoreo Fecha: jue 5/9/19	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha limite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin				

Página 2

3.5 ANÁLISIS DE RECURSOS

Para la elaboración del proyecto se utilizaron diferentes recursos que a continuación se detallan:

Tabla 4. Talento Humano

Descripción	Funciones	Cantidad	Responsable
Digitador	Redactar la información y documentar el proceso del Proyecto.	2	Freddy Peñafiel José Huaipatin
Programador	Desarrollo del código HTML	2	Freddy Peñafiel José Huaipatin

Tabla 5. Recursos técnicos

Recurso	Descripción	Cantidad	Costo	Subtotal
Software	PHP	1	\$0,00	\$0,00
	Apache	1	\$0,00	\$0,00
	MySQL	1	\$0,00	\$0,00
	Raspbian	1	\$0,00	\$0,00
	ARDUINO	1	\$0,00	\$0,00
	HTML5	1	\$0,00	\$0,00
	Python	1	\$0,00	\$0,00
	Frtizing	1	\$0,00	\$0,00
Hardware	Raspberry Pi 3 Modelo B+	1	\$79,95	\$79,95
	Arduino UNO R3	1	\$44,95	\$44,95
	Sensor MQ-2	1	\$3,00	\$3,00
	Sensor DHT11	1	\$3,00	\$3,00
	Sensor PIR	1	\$5,00	\$5,00
	Módulo RFS1000A	1	\$4,00	\$4,00
	Board	1	\$5,00	\$5,00
	Cámara Pi	1	\$30,00	\$30,00
	Conectores Dipont	50	\$0,15	\$7,50
	Resistencias	5	\$0,20	\$1,00
			Total	\$182,40

Tabla 6. Recursos financieros

Descripción	Cantidad	Precio
Laptop Dell Inspiron 14 3467 Core I5	1	\$600,00
Impresora Multifunción HP WL315	1	\$231,71
Resma de Hoja A4	3	\$30

La implementación de este proyecto es factible por el creciente desarrollo de la tecnología, tenemos la disponibilidad de encontrar diversas herramientas de software y hardware a bajo costo, nos permitirá brindar una solución para prevenir de probables accidentes y enfermedades mediante el monitoreo de un ambiente bebés, sin embargo en el transcurso de las pruebas podemos encontrar restricciones tales como: El lugar donde se encuentre el infante falle el suministro de energía eléctrica, el acceso a internet se encuentre caído, dispositivo donde debe llegar las notificaciones se encuentre apagado, fuera de cobertura o sin acceso a plan de celular.

CAPÍTULO 4

METODOLOGÍA Y EJECUCIÓN DEL PROYECTO TÉCNICO

4.1 METODOLOGÍA UTILIZADO PARA EL PRESENTE PROYECTO

Para efectuar la etapa de investigación se efectuó una extensa exploración y análisis del problema trazado en el Capítulo 1, para esto se consultó a varios papás de para conocer su criterio, utilizando la metodología XP. Los requisitos se hicieron a través de historia de usuarios (Revisar Anexo E Historia de Usuarios), en base a esto se puede definir el alcance del proyecto.

Utilizamos la metodología técnica para efectuar las entrevistas para limitar las historias de usuarios para una posible ejecución. El estudio realizado fue basado en el tiempo, conocimientos y costo requeridos para materializar el producto terminado.

4.1.2 MUESTRA

Población: Personas entre los 18 a 45 años que trabajan dentro de la ciudad de Guayaquil y viven en la Coop. Río Guayas.

Margen de error: 5%

Nivel de Confianza: 95%

Para obtener el tamaño de la muestra aplicamos la siguiente formula

$$\frac{N * (\alpha_c * 0,5)^2}{1 + (e^2 * (N - 1))} =$$

Figura 32. Cálculo de tamaño de muestra

Donde:

N: 110 personas que componen la población descrita (universo).

α_c Valor de nivel de confianza, es el riesgo que aceptamos de equivocarnos al presentar nuestros resultados, el nivel habitual de confianza es del 95%.)

e : Margen de error, es el error que estamos dispuesto a aceptar de equivocarnos al seleccionar nuestra muestra; este margen error suele ponerse en torno a un 3%

Resultado obtenido es 100 personas, luego de tener el resultados de las encuestas se procedió a realizar las pruebas con el prototipo, para esto se eligió a cinco familias debido que tenemos dos dispositivos de ambiente de bebés.

Fuente: http://doc.contraloria.gob.pe/libros/2/pdf/Metodo_estadistico_2.xls

Pregunta 1

¿Ha estado alguna vez a cargo del cuidado de un bebé?

Si ___ No ___

Tabla 7. Resultado pregunta 1

RESULTADO PREGUNTA 1			
Opciones	Frecuencia Absoluta	Frecuencia Relativa	
Si		63	63
No		37	37
Total		100	100

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

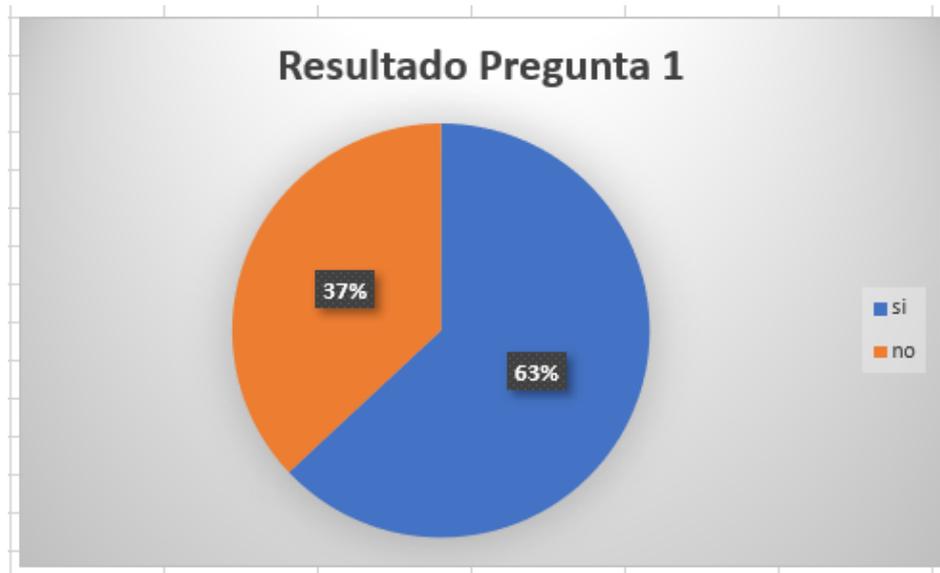


Figura 33. Resultado Pregunta 1

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

De la muestra obtenida el 63% estuvieron en constante vigilancia de un recién nacido, no precisamente eran los papás.

Pregunta 2

¿En alguna ocasión ha escuchado sobre la tecnología que ayude con el monitoreo del bebe?

Si ___

No ___

Tabla 8. Resultado pregunta 2

RESULTADO PREGUNTA 2		
Opciones	Frecuencia Absoluta	Frecuencia Relativa
Si	71	71
No	29	29
Total	100	100

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 34. Resultado pregunta 2

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

EL 71% habían oído sobre diferentes tecnologías para la ayuda del monitoreo de bebes.

Pregunta 3

¿Es indispensable el uso de algún programa que facilite el monitoreo del ambiente de bebés?

Muy indispensable ___ Indispensable ___ No indispensable___

Tabla 9. Resultado pregunta 3

RESULTADO PREGUNTA 3		
Opciones	Frecuencia Absoluta	Frecuencia Relativa
Muy indispensable	60	60
Indispensable	40	40
No indispensable	0	0
Total	100	100

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 35. Resultado pregunta 3

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

El 60% considera usar la tecnología para monitorear al infante.

Pregunta 4

¿Le gustaría utilizar de monitoreo ambiente para bebés?

Muy interesado ___ Interesado ___ No estoy interesado ___

Tabla 10. Resultado pregunta 4

RESULTADO PREGUNTA 4		
Opciones	Frecuencia Absoluta	Frecuencia Relativa
Muy Interesado	30	30
Interesado	57	57
No estoy interesado	13	13
Total	100	100

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 36. Resultado pregunta 4

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

El 30% manifestó estar muy interesado, el 57% expresó estar interesado y el restante no estaba interesado.

Pregunta 5

¿Cuáles serían las opciones que convendría brindar un equipo de monitoreo de bebés?

Video en línea ____ Envío de alertas ____ Reconocimiento facial ____

Temperatura del ambiente ____ Otras opciones ____

Detalle que opciones: _____

Tabla 11. Resultado pregunta 5

RESULTADO PREGUNTA 5		
Opciones	Frecuencia Absoluta	Frecuencia Relativa
Video en tiempo real	38	38
Temperatura	24	24
Humedad	11	11
Reconocimiento facial	12	12
Envío de alertas	15	15
Total	100	100

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 37. Resultado pregunta 5

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

En la pregunta que antecede la mayoría de los encuestados vieron con agrado tener un monitoreo en tiempo real del infante, el segundo mejor resultado fue tener un detector de temperatura en la habitación del bebe luego envío de alertas, la penúltima característica es la detección de personas y por último la humedad.

Pregunta 6

¿Estaría listo en adquirir en un sistema de monitoreo de ambiente para bebés?

De \$400 o mayor ___

Entre \$200 y \$400 ___

No mayor a \$200 ___

Tabla 12. Resultado pregunta 6

RESULTADO PREGUNTA 6		
Opciones	Frecuencia Absoluta	Frecuencia Relativa
>= \$400	2	2
>= \$200 y <= \$400	39	39
<= \$200	59	59
Total	100	100

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 38. Resultado pregunta 6

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

El 59% no estaba dispuesto a pagar más de \$200 por un sistema de monitoreo, EL 39% estaba dispuesto en pagar entre \$200 y \$400, además una persona podría pagar más de \$400 por un sistema de monitoreo de ambiente para bebés.

4.1.3 MUESTRA DE PRUEBA

Resultados obtenidos luego de la implementación del prototipo realizada a cinco familias

Pregunta 1

¿Los sensores de prototipo funcionaron?

Si ___ No ___

Tabla 13. Resultado de la pregunta 1

RESULTADO PREGUNTA 1	
Si	4
No	1
Total	5

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 39. Resultado pregunta 1

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

El prototipo de monitoreo de ambiente de bebés cumple con las expectativas en los hogares

Pregunta 2

¿Se recibieron las notificaciones de alertas por WhatsApp?

Si ___

No ___

Tabla 14. Resultado de la pregunta 2

RESULTADO PREGUNTA 1	
Si	5
No	0
Total	5

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 40. Resultado pregunta 2

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

Los cambios de temperatura, gases y humedad fueron enviados a través de la mensajería instantánea fueron exitosos.

Pregunta 3

¿El manual del usuario para configurar el dispositivo fue sencillo?

Si ___

No ___

Tabla 15. Resultado de la pregunta 2

RESULTADO PREGUNTA 1	
Si	5
No	0
Total	5

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel



Figura 41. Resultado pregunta 2

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Resultados Obtenidos:

El manual del usuario cumplió con la expectativa para configurar el dispositivo

CAPÍTULO 5

EVALUACIÓN DEL PROYECTO TÉCNICO

5.1 EVALUACIÓN

En la siguiente tabla se muestra la ficha con los datos técnicos del producto BEBECAM con cada una de sus partes.

Tabla 16. Raspberry PI 3 Modelo B+

Modelo	Raspberry PI 3 B+
Sistema Operativo	Raspbian JSSEI
Memoria	1GB 1.2 GHz
Procesador	ARM Quad Core
Almacenamiento	Micro SD 8 GB+
Temperatura Ideal	< 50°
Voltaje Recomendado	5 V.

Tabla 17. Arduino

Modelo	Arduino Uno R3
Temperatura ideal	<50°
Microcontrolador	ATmega328P
Voltaje Recomendado	5-9 V.
Método de comunicación	Serial

Tabla 18. Cámara PI

Pixeles	5MP HD
Método de Comunicación utilizado	Serial
Regulador de 1.5 V	Incorporado
Resolución del sensor	2592 x 1955 pixeles

Tabla 19. Sensor de temperatura y humedad

Nombre	DHT11
Temperatura ideal	<80°
Voltaje recomendado	3 - 5 V.
Burn in	No es necesario
Lectura de valores	cada 2 segundos
Resistencia recomendada	5KOhm
Lecturas de humedad	20 a 80% con 5% de precisión
Lecturas de temperatura	0 a 50 ° C con +/- 2° C de precisión
Distancia de detección	Hasta 5 metros

Tabla 20. Sensor de gas

Nombre	MQ2
Temperatura ideal	<50°
Voltaje recomendado	3 - 5 V.
Burn in	3 minutos
Lectura de valores	cada 0,5 segundos
Resistencia recomendada	5KOhm
Distancia de detección	Hasta 5 metros

Tabla 21. Sensor de detección de personas

Nombre	PIR HC_SR501
Alimentación	3,3 a 5 Vdc_Vdc
Rango de temperatura	0 a +50 ° C_C
Ángulo de detección	180°_180°
Distancia de detección	Hasta 7 metro
Burn-in	60 segundos.
Lectura de valores	cada 1 segundo
Resistencia recomendada	5KOhm_5Kohm

Tabla 22. Aplicación Web

Servidor web	Apache 2.4.10
Vista	HTML5
Modelo	Laravel
Controlador	PHP

Tabla 23. Base de Datos

Versión	MySQL
----------------	-------

Tabla 24. Servicio de Monitoreo

Nombre	lecturarduino.py
Lenguaje de Programación	Python 2.7
Frecuencia de lectura	2 segundos

Tabla 25. Servicio de mensajería

Nombre	twilio.py
Lenguaje de Programación	Python 2.7
Frecuencia de lectura	5 segundos

Tabla 26. Servicio de Streaming

Nombre	atream.sh
Lenguaje de Programación	Python 2.7
Frecuencia de lectura	0,1 segundos

5.2 CRITERIOS DE ACEPTACIÓN

Tabla 27. Matriz de criterio de aceptación

Criterios de aceptación	Alcance	Cumple
Debe permitir vigilar bebés de sexo masculino como femenino	La solución propuesta está dirigida a bebés de sexo femenino o masculino luego que haya sido dados de alta en la maternidad	Si
Debe permitir realizar el monitoreo de un espacio de 4 metros cuadrados	El perímetro de monitoreo de BEBECAM es un espacio de 4 metros cuadrados	Si
Debe indicar los valores de temperatura, humedad, detección de persona y gas	Para realizar el monitoreo se utilizarán sensores de temperatura, detección de personas y gases tóxicos los cuales serán conectados a una placa ARDUINO	Si
El servicio de monitoreo RPI debe leer y luego almacenar en la base de datos	Para el desarrollo del servicio de monitoreo RPI se utilizará el lenguaje de programación Python, el servicio realizará la consulta de lo captado por los sensores y almacenará dicha información en la base MySQL	si
Se requiere visualizar la página web a través de una computadora	La página WEB será desarrollada en PHP y estará publicada en la INTRANET del hogar por lo que podrá ser visualizada a través de una computadora	Si
En la base de datos debe almacenar los eventos captados por los sensores, no los videos	Se almacenará en la base de datos MySQL el historial de máximo un mes de los eventos presentados, los videos no serán almacenados	Si
Se requiere recibir alertas cuando los sensores estén midiendo eventos definidos afuera de los rangos normales	Las alertas o notificaciones que se susciten serán enviadas por WhatsApp a los contactos que se definan en la configuración inicial del software	

Elaborado: José Huaipatin y Freddy Peñafiel

5.3 INFORME DE ASEGURAMIENTO DE CALIDAD

En el siguiente cuadro se mostrará el aseguramiento de la calidad del software que usará el prototipo.

Tabla 28. Detalle de las actividades del SQA

Tarea	Detalle
Revisión del software	Se valida el código cumpla con las mejoras prácticas y estándares de programación
Verificación	<ul style="list-style-type: none">• Verificación cumplimiento del alcance.• Validación de la documentación.• Validación de la arquitectura requerida para cumplir con los requerimientos del proyecto
Pruebas	Validación de la arquitectura requerida para cumplir con los requerimientos del proyecto

Elaborado: José Huaipatin y Freddy Peñafiel

Fuente: José Huaipatin y Freddy Peñafiel

Por medio de los diferentes ejercicios realizados se busca asegurar que la calidad del producto final sea aceptable.

Para la evaluación de la calidad del software y hardware, se utilizarán el modelo de evaluación FURP.

Por medio de este modelo el proyecto tiene como objetivo cumplir con los siguientes aspectos de la calidad de software y hardware:

Características	Detalles	Métricas
Funcionalidad	Seguridad de acceso al modulo	Prevención del uso indebido de los datos.
	Cumplimiento de la funcionalidad	Cumplimiento del alcance. Cumplimiento de estándares de funcionamiento del hardware.
Usabilidad	Documentación del usuario.	Manuales de usuarios.
	Operatividad	Mensajes y claros y precisos. Operatividad del hardware.
Contabilidad	Tolerancia a errores	Reversar operaciones incorrectas
Eficiencia	Disponibilidad	Disponibilidad del módulo.
Soporte	Capacidad de configuración	Prueba de configuración de software
		Prueba de configuración de hardware Elaborado: José Huaipatin y Freddy Peñafiel Fuente: José Huaipatin y Freddy Peñafiel

5.4 MECANISMOS DE CONTROL

Para la protección del prototipo se emplearon varios mecanismos de control. Para la creación y configuración revisar el Anexo G Manual del Usuario.

- Autenticación con usuario y contraseña.
- Autorización a usuarios por módulos.
- Control por roles de usuario tales como:
 - Rol Administrador. – Acceso total a las configuraciones.
 - Rol Supervisor. – Acceso para crear y registrar usuarios, asignación de permisos y reportes.

- Rol Usuario. – Acceso para acceder a la pantalla principal.

CONCLUSIONES

- Realizando las investigaciones podemos decir que el presente proyecto se viable debido que en el mercado no existe un dispositivo con sensores que cumpla con las características similares al propuesto en esta tesis.
- El Raspberry Pi en la actualidad es el minicomputador más amigable, cómodo para desarrollar, en base a esto podemos concluir que es sencillo realizar configuraciones de los sensores en el presente proyecto.
- Para visualizar el monitoreo del ambiente del infante desde cualquier lugar, teniendo acceso a internet, se utilizó la plataforma web gratuita No-IP, la cual nos permite acceder a nuestra cámara de forma remota y así poder visualizar en tiempo real lo que sucede en la habitación del infante.
- El tiempo de respuesta de las alertas entre Raspberry Pi y los sensores del presente proyecto es contigua en los dispositivos móviles.
- La fabricación del prototipo con respecto a equipos que existe en el mercado son de bajo costo.

RECOMENDACIONES

Como trabajo para más adelante se recomienda la realización de las siguientes mejoras:

- Para el monitoreo de bebés es recomendable una ventilación adecuada para no tener problema con el calentamiento de equipos.
- Analizar cuidadosamente la ubicación donde será instalado el dispositivo dentro de una habitación, los puntos críticos de vigilancia pueden variar de acuerdo con la utilidad que se le quiera dar al sistema y al diseño de la vivienda en que se instale.
- Eliminación del cableado correspondiente para la comunicación entre Arduino y Raspberry PI, gracias a esta eliminación, el prototipo se transformaría en un producto más práctico y funcional, entre las opciones disponibles se recomienda utilizar antenas RF de bajo costo en conjunto con los pines GPIO.
- Diseñar e implementar un sistema de almacenamiento de video, con la finalidad de que el usuario pueda revisar lo grabado por la cámara, Raspberry PI cuenta con 4 puertos USB que pueden ser utilizados para conectar un Disco Duro Externo.
- Diseñar y elaborar un case que integre y proteja los sensores.

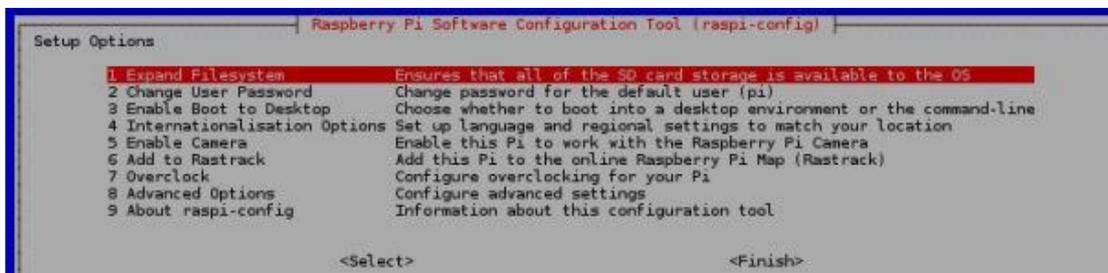
ANEXOS

ANEXO A Instalación del Sistema Operativo

Al reiniciar el dispositivo el primer programa que se ejecuta se llama raspi-config, este programa solo se ejecuta en inglés. En el caso de necesitar realizar alguna modificación, lo puede hacer ejecutando el siguiente comando desde la terminal:

```
sudo raspi-config
```

El primer menú que muestra contiene 9 diferentes opciones disponibles, a continuación, revisaremos cada una de las opciones disponibles:



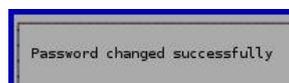
Opción 1 – Expandir el sistema de archivos (Expand Filesystem)

Esta opción permite expandir el sistema operativo para que utilice todo el espacio disponible en la tarjeta. Cuando se instala Raspbian “Wheezy” la imagen copiada en la tarjeta solo ocupa 2 GB, por lo tanto, es necesario ejecutar esta opción para que todo el espacio de la tarjeta SD sea utilizado.

Opción 2 – Cambiar la contraseña del usuario Pi (Change User Password)

En el Raspberry Pi vienen predeterminados los usuarios “root” y “pi”.

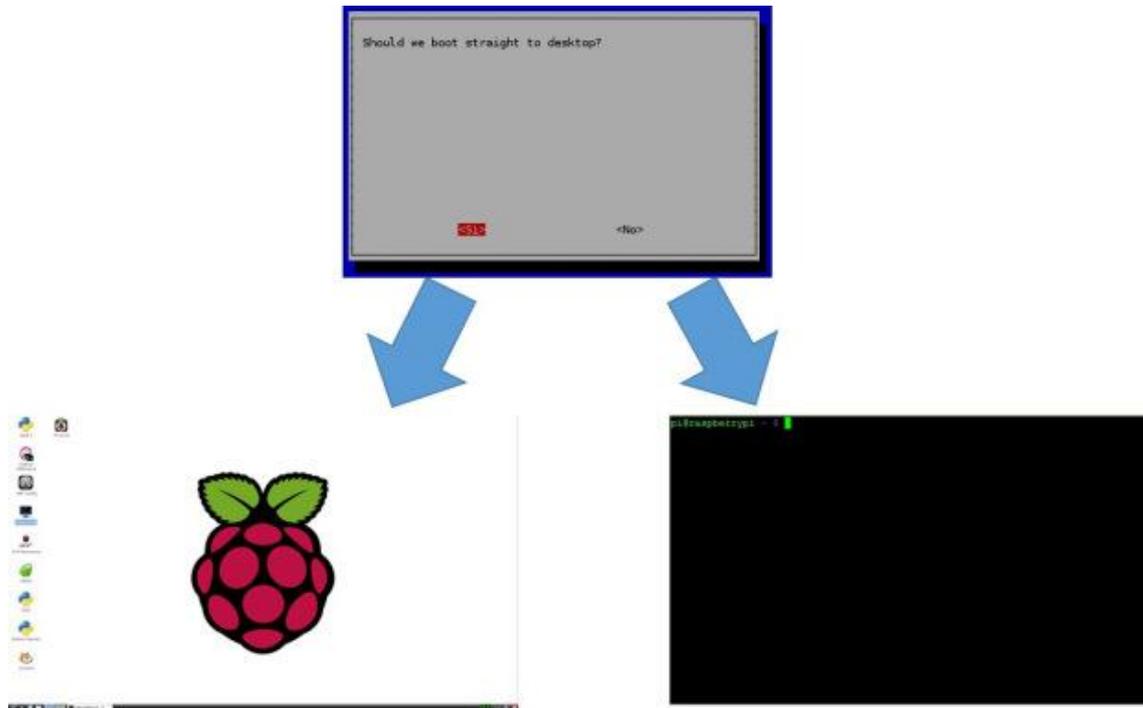
El usuario “pi” viene predeterminado con la contraseña “Raspberry” por lo tanto cualquier persona podría acceder al sistema. Por eso, es recomendable cambiar la contraseña en esta opción. El sistema le solicitará que ingrese la nueva contraseña y que la repita nuevamente, al finalizar aparecerá el siguiente mensaje:



Opción 3. Activar el escritorio al iniciar (Enable Boot to Desktop)

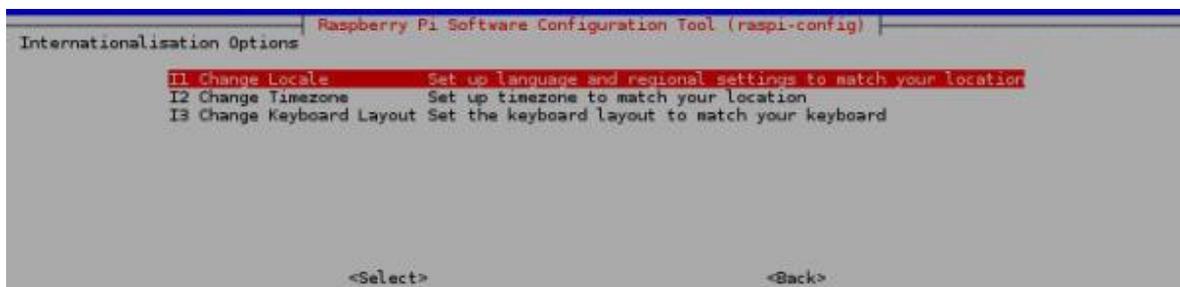
Esta opción permite que el Raspberry Pi comience inmediatamente el escritorio modo gráfico o en línea de comando. En caso de que inicie en modo de línea de comando y desee ingresar al modo gráfico solo ingrese el siguiente código:

```
startx
```



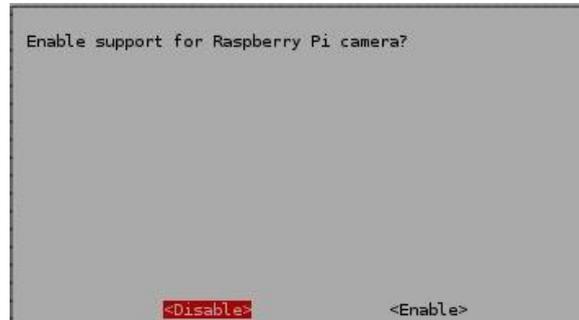
Opción 4. Opciones de internacionalización (Internationalisation Options)

Esta opción permite modificar el lenguaje del sistema operativo, la zona horaria y la distribución de su teclado.



Opción 5 – Activar la cámara (Enable camera)

Esta opción sirve para dar soporte a la cámara de Raspberry Pi, permite activar el puerto de comunicación entre la CPU y el controlador de la cámara.



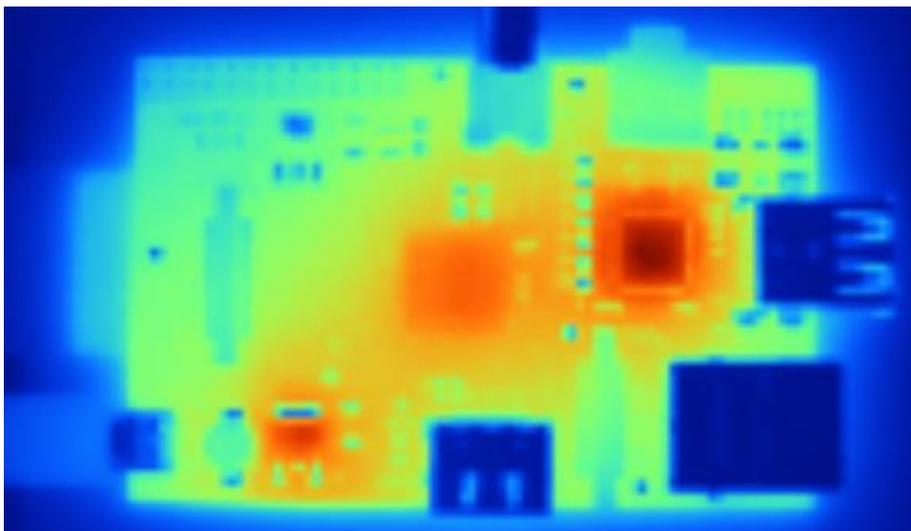
Opción 6 – Adicionar a Rastrack (Add to Rastrack)

Esta opción permite que su Raspberry Pi sea rastreado por el sitio web Rastrack. Es una herramienta para tener la estadística de donde se encuentran los Raspberry Pi en el mundo.

Fuente: <http://rastrack.co.uk>.

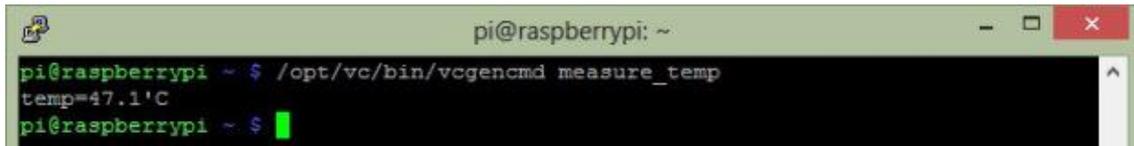
Opción 7 – Overclocking

Esta opción permite aumentar la velocidad del procesador. Se recomienda dejar la predeterminada, en el caso de elegir esta opción deberá considerar aumentar la energía y agregar disipadores de calor en la placa Raspberry Pi.



Para verificar la temperatura de la CPU del Raspberry Pi utilice el siguiente comando:

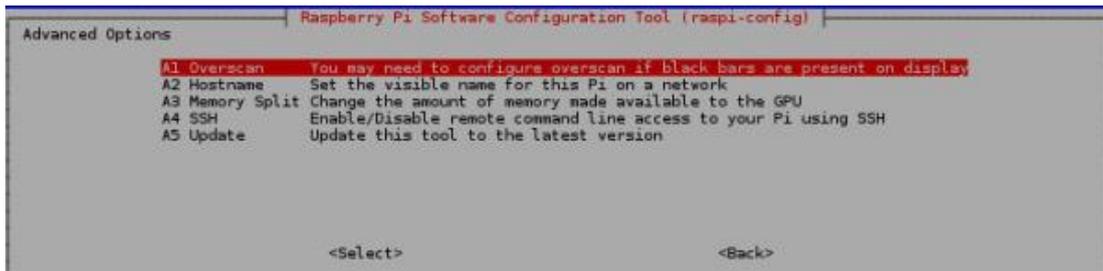
```
/opt/vc/bin/vcgencmd measure_temp
```



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ /opt/vc/bin/vcgencmd measure_temp  
temp=47.1'C  
pi@raspberrypi ~ $
```

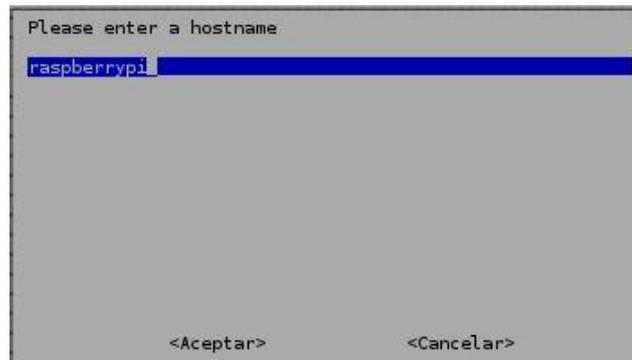
Opción 8. Opciones avanzadas (Advanced Options)

Esta opción presenta un submenú con las siguientes opciones.



La opción A1 overscan sirve para borrar las líneas negras en algunos monitores o televisores.

La opción A2 Hostname, sirve para identificar el Raspberry Pi en la red local.



La opción A3 – Distribución de la memoria (Memory Split) permite seleccionar la cantidad de memoria compartida entre la CPU y la unidad de gráficos (GPU), el modelo B versión 2.0 cuenta con 512 MB en total. El predeterminado es 64 MB para la memoria de vídeo. Al aumentar la memoria de vídeo y disminuir la del procesador, este se vuelve un poco más ‘lento’ para ejecutar otras tareas.



La opción A4 – Activar SSH (Enable SSH) se utiliza para acceder el Raspberry Pi remotamente desde un cliente SSH. Es necesario activar esta opción para conectarse remotamente al dispositivo.

La opción A5 – Actualizar (update) se utiliza para actualizar el sistema. O lo puede hacer con el siguiente comando:

```
sudo apt-get update
```

ANEXO B Instalación de MJPG Streamer

Adicionamos el código en /home/pi/mjpg-streamer.sh

```
#!/bin/bash
# chmod +x mjpg-streamer.sh
# Crontab: @reboot /home/pi/mjpg-streamer/mjpg-streamer.sh start
# Crontab: @reboot /home/pi/mjpg-streamer/mjpg-streamer-experimental/mjpg-streamer.sh
start
```

```
MJPEG_STREAMER_BIN="/usr/local/bin/mjpg_streamer" #
"${dirname$0}/mjpg_streamer"
MJPEG_STREAMER_WWW="/usr/local/share/mjpg-streamer/www"
MJPEG_STREAMER_LOG_FILE="${0%.*}.log" # "${dirname $0}/mjpg-streamer.log"
RUNNING_CHECK_INTERVAL="2" # how often to check to make sure the server is
running (in seconds)
HANGING_CHECK_INTERVAL="3" # how often to check to make sure the server is not
hanging (in seconds)
VIDEO_DEV="/dev/video0"
FRAME_RATE="5"
QUALITY="80"
RESOLUTION="1280x720" # 160x120 176x144 320x240 352x288 424x240 432x240
640x360 640x480 800x448 800x600 960x544 1280x720 1920x1080 (QVGA, VGA, SVGA,
WXGA) # lsusb -s 001:006 -v | egrep "Width|Height" #
https://www.textfixer.com/tools/alphabetical-order.php # v4l2-ctl --list-formats-ext #
Show Supported Video Formates
PORT="8081"
YUV="yes"
#####INPUT_OPTIONS="-r ${RESOLUTION} -d ${VIDEO_DEV} -f
${FRAME_RATE} -q ${QUALITY} -pl 60hz"
INPUT_OPTIONS="-r ${RESOLUTION} -d ${VIDEO_DEV} -q ${QUALITY} -pl 60hz
--every_frame 2" # Limit Framerate with "--every_frame ", ( mjpg_streamer --input
"input_uvc.so --help" )
```

```

if [ "${YUV}" == "true" ]; then
    INPUT_OPTIONS+=" -y"
fi

OUTPUT_OPTIONS="-p ${PORT} -w ${MJPEG_STREAMER_WWW}"

# =====
function running() {
    if ps aux | grep ${MJPEG_STREAMER_BIN} | grep ${VIDEO_DEV} >/dev/null 2>&1;
then
    return 0
    else
    return 1
    fi
}

function start() {
    if running; then
        echo "[${VIDEO_DEV}] already started"
        return 1
    fi

    export LD_LIBRARY_PATH="$(dirname $MJPEG_STREAMER_BIN):."

    echo "Starting: [${VIDEO_DEV}] ${MJPEG_STREAMER_BIN} -i \"input_uvc.so
${INPUT_OPTIONS}\" -o \"output_http.so ${OUTPUT_OPTIONS}\""
    ${MJPEG_STREAMER_BIN} -i "input_uvc.so ${INPUT_OPTIONS}" -o "output_http.so
${OUTPUT_OPTIONS}" >> ${MJPEG_STREAMER_LOG_FILE} 2>&1 &

    sleep 1

    if running; then
        if [ "$1" != "nocheck" ]; then

```

```

        check_running & > /dev/null 2>&1 # start the running checking task
        check_hanging & > /dev/null 2>&1 # start the hanging checking task
    fi
    echo "[$VIDEO_DEV] started"
    return 0

else
    echo "[$VIDEO_DEV] failed to start"
    return 1

fi
}

function stop() {
    if ! running; then
        echo "[$VIDEO_DEV] not running"
        return 1
    fi

    own_pid=$$

    if [ "$1" != "nocheck" ]; then
        # stop the script running check task
        ps aux | grep $0 | grep start | tr -s ' ' | cut -d ' ' -f 2 | grep -v ${own_pid} | xargs -r kill
        sleep 0.5
    fi

    # stop the server
    ps aux | grep ${MJPEG_STREAMER_BIN} | grep ${VIDEO_DEV} | tr -s ' ' | cut -d ' ' -f
2 | grep -v ${own_pid} | xargs -r kill

    echo "[$VIDEO_DEV] stopped"
    return 0
}

```

```

}
function check_running() {
    echo    "[${VIDEO_DEV}    starting    running    check    task"    >>
    ${MJPEG_STREAMER_LOG_FILE}

    while true; do
        sleep ${RUNNING_CHECK_INTERVAL}

        if ! running; then
            echo    "[${VIDEO_DEV}    server    stopped,    starting"    >>
            ${MJPEG_STREAMER_LOG_FILE}
            start nocheck
        fi
    done
}

function check_hanging() {
    echo    "[${VIDEO_DEV}    starting    hanging    check    task"    >>
    ${MJPEG_STREAMER_LOG_FILE}

    while true; do
        sleep ${HANGING_CHECK_INTERVAL}
        # treat the "error grabbing frames" case
        if tail -n2 ${MJPEG_STREAMER_LOG_FILE} | grep -i "error grabbing frames" >
/dev/null; then
            echo    "[${VIDEO_DEV}    server    is    hanging,    killing"    >>
            ${MJPEG_STREAMER_LOG_FILE}
            stop nocheck
        fi
    done
}

function help() {
    echo "Usage: $0 [start|stop|restart|status]"
}

```

```

    return 0
}

if [ "$1" == "start" ]; then
    start && exit 0 || exit -1

elif [ "$1" == "stop" ]; then
    stop && exit 0 || exit -1

elif [ "$1" == "restart" ]; then
    stop && sleep 1
    start && exit 0 || exit -1

elif [ "$1" == "status" ]; then
    if running; then
        echo "[$VIDEO_DEV] running"
        exit 0

    else
        echo "[$VIDEO_DEV] stopped"
        exit 1
    fi

else
    help

fi

```

Reiniciamos

Make Executable

```
chmod +x /home/pi/mjpg-streamer.sh
```

Open Cron Job

```
crontab -e
# Add line
@reboot /home/pi/mjpg-streamer.sh start
```

Instalación FMPEG

```
# Install FFMpeg from Package Manager
sudo apt-get update
sudo apt-get install ffmpeg -y
```

---- OR -----

COMPILE & INSTALL FFMPEG

```
# Run as Sudo
```

```
sudo -i
```

```
# INSTALL H264 SUPPORT
```

```
cd /usr/src
```

```
git clone git://git.videolan.org/x264
```

```
cd x264
```

```
./configure --host=arm-unknown-linux-gnueabi --enable-static --disable-openc1
```

```
make
```

```
sudo make install
```

```
# COMPILE & INSTALL FFMPEG (This may take a REALLY long time, so be patient.)
```

```
cd /usr/src
```

```
git clone https://github.com/FFmpeg/FFmpeg.git
```

```
cd FFmpeg
```

```
sudo ./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-  
nonfree
```

```
make
```

```
sudo make install
```

```
Instalación Receded Stream w/ffmpeg
```

```
# [Variables]
```

```
source_stram="http://localhost:8080/?action=stream"
```

```
destination_directory="/home/pi/Videos"
```

```
destination_file="cncjs-recording_$(date +%Y%m%d_%H%M%S').mpeg"
```

```
# Recored Stream w/ ffmpeg
```

```
ffmpeg -f mjpeg -re -i "${source_stram}" -q:v 10
```

```
"${destination_directory}/${destination_file}"
```

ANEXO C Instalación de Twilio

Lo primero que se tiene que hacer es instalar la librería de Twilio en la Raspberry Pi desde el terminal con el comando: **sudo pip install**.

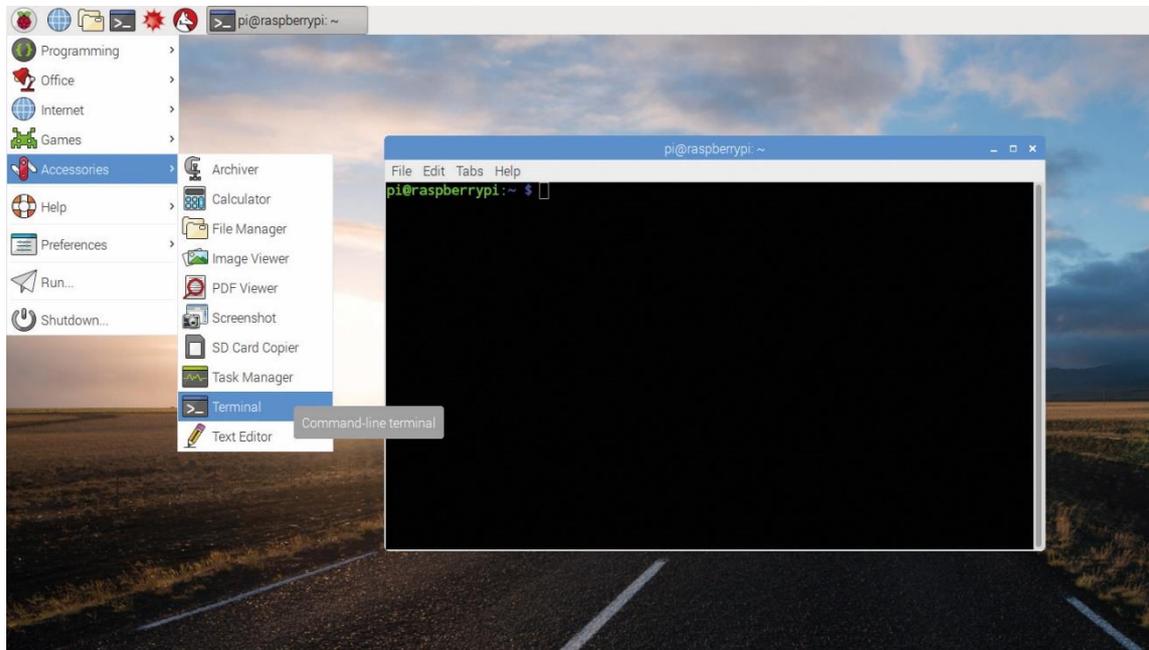


Figura 42. Consola de Terminal de Raspberry PI

Elaborado: José Huaipatin y Freddy Peñafiel

Una vez instalada las librerías, hay que dirigirse a la página oficial de Twilio en donde debe crearse una cuenta que proporcionará una cuenta SID que es como un código que se proporciona a las diferentes cuentas que se creen en la plataforma y además proporcionan un token de autenticación (ACCOUNT SID y AUTH TOKEN) los cuales serán habilitados en el código respectivo de whatsapp.py

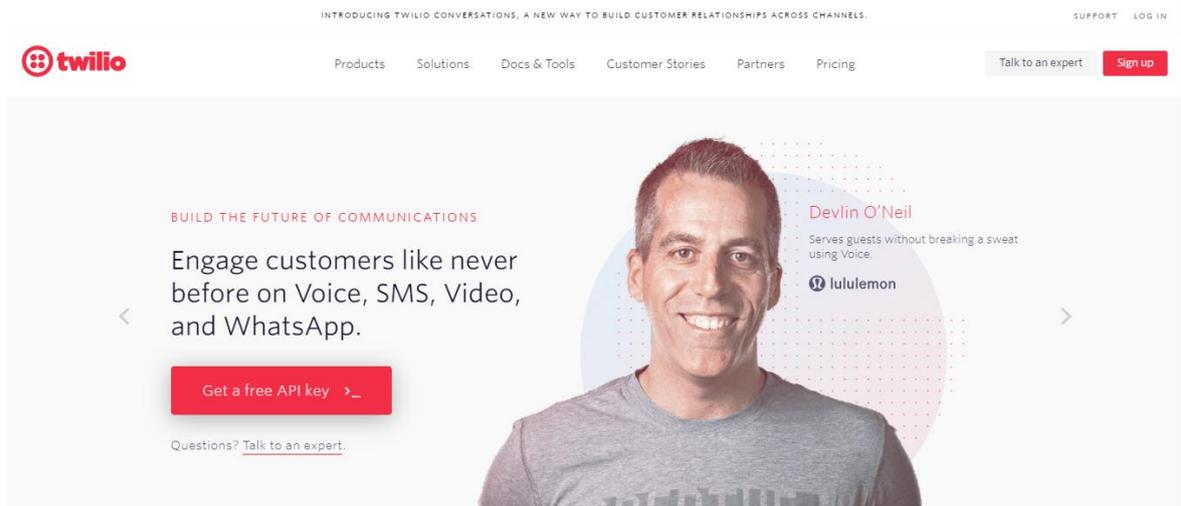


Figura 43. Página de Twilio

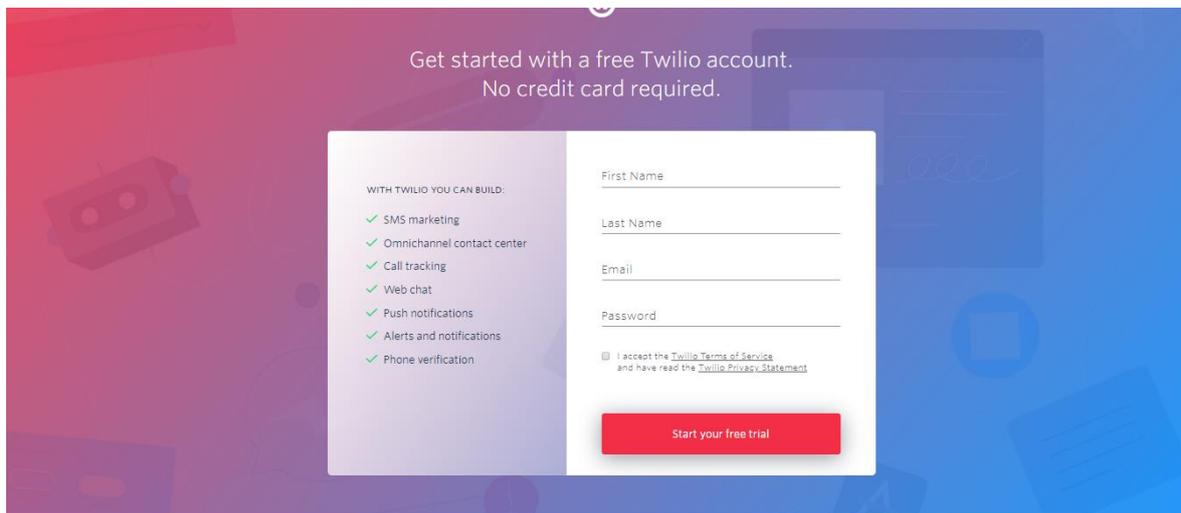


Figura 44. Formulario de registro para Twilio

En la siguiente figura se observa la interfaz principal de Twilio con la cuenta ya creada y con su saldo respectivo y las claves de autenticación que han sido ocultas en la imagen por seguridad de la cuenta.

Prototipo Raspberry Pi José- Freddy Dashboard

Project Info

TRIAL BALANCE	TRIAL NUMBER	Here's how your Twilio Trial account works: <ul style="list-style-type: none">You can send messages and make calls to verified numbers.Messages and calls include a note about this coming from a "T account." Learn more about your trial or upgrade to remove restrictions.
\$14.50	+12057079992 Need more numbers?	
ACCOUNT SID	ACb04318baa202bf1e716fb1f3502ff31b	
AUTH TOKEN	Show	PROJECT NAME: Prototipo Raspberry P... PROGRAMMING LANGUAGE: Python

Después se configura el código de **whatsapp.py** que está en la carpeta de **NOTIFICACIÓN**, en la primera línea de codificación se importa la librería de Twilio seguidamente se coloca el “account SID” de la cuenta creada en Twilio con el token o clave de seguridad. La siguiente línea: **client = Client(account_sid, auth_token)** enlaza los primeros parámetros mencionados anteriormente y su método cliente hace la autenticación. Luego se define la clase para el primer mensaje en donde se coloca en “from” el número de teléfono que proporciona la plataforma y en “to” el número al cual se quiere recibir las notificaciones (del autor del proyecto) además se coloca el mensaje a enviar para el rango de temperatura del ambiente donde se encuentra el infante: **'La temperatura es mayor a 37.5 grados, por favor revisar el ambiente del niño '**

```
8 class whatsapp_mensaje_1:
9     #aquí definimos una función con un método llamado __init__ con un parámetro self que guarda la referencia de lo que se haya crea
10    # en nuestro caso, nos envía el mensaje de whatsapp
11    def __init__(self):
12        # si todo es correcto mediante el método cliente.messages se crea el mensaje a enviar
13        whatsapp1 = client.messages.create(
14            body='La temperatura es mayor a 37.5 grados, se considera FIEBRE. Por favor revisar al niño',
15            from_='whatsapp:+', # el número otorgado por twilio
16            to='whatsapp:+' # el número del destinatario a enviar el mensaje
17        )
```

Para definir la segunda clase se procede a hacer lo mismo que la clase anterior, pero el mensaje a enviar cambia de acuerdo al rango de temperatura alta del ambiente **'La temperatura es mayor a 39.5 grados, se considera TEMPERATURA ALTA. Por favor revisar al niño**

```

24 class whatsapp_mensaje_2:
25     def __init__(self):
26
27         whatsapp1 = client.messages.create(
28
29             body='La temperatura es mayor a 39.5 grados, se considera FIEBRE ALTA. Por favor revisar al niño',
30             from_='whatsapp:          i',
31             to='whatsapp:+'
32         )

```

Una vez que haya tenido el código de WhatsApp con las dos condiciones o mensajes a tomar en cuenta, hay que llamar la función creada en whatsapp.py en el código principal del sensor. Entonces se modifica nuevamente el código principal para ello en las primeras líneas se hace una importación de la librería de Twilio: `from twilio.rest import Client`.

Y luego en la parte de las condicionales establecidas en el código principal es en donde se tiene que llamar a la función del primer y segundo mensaje de whatsapp.py.

Para llamar a la función lo que se tiene que hacer es poner el nombre del archivo.py en este caso “WhatsApp” seguido se coloca un punto y el nombre de la clase que se creó en dicho archivo así: `whatsapp.whatsapp_mensaje_1()`

Se llama al mensaje 1 después de la condicional que está en el rango de fiebre, es decir después de: `elif (temp > 37.5 and temp <39.5)`.

Luego para llamar a la función del segundo mensaje simplemente cambia la clase: `whatsapp.whatsapp_mensaje_2()` que será colocado luego de la condición `else` que indique el rango de fiebre alta. La modificación del archivo principal (`mlxtodo.py`) con la notificación de WhatsApp y Gmail.

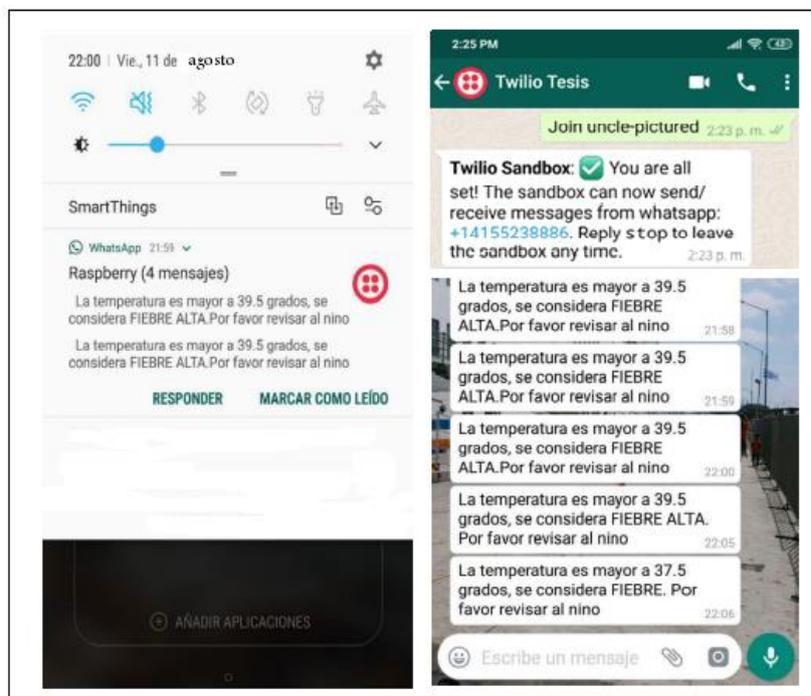
Ahora se requiere entrar a la plataforma en la opción “SMS programables” se selecciona “WhatsApp Beta” para realizar una configuración que permita el envío de mensaje desde la plataforma a su teléfono móvil, es necesario que para comenzar: mandar un mensaje desde su WhatsApp al número proporcionado por la plataforma con la frase **join uncle-pictured**, como se visualiza en la imagen



Antes agregue el número proporcionado a sus contactos, en este caso se lo agregó con el nombre de “**Twilio Tesis**” y envíe el mensaje solicitado.



Ahora se ejecuta el programa desde el terminal: **Python** `/home/pi/NOTIFICACION/mlxtodo.py` y la medición que detecte en fiebre o fiebre alta la enviará automáticamente como notificación un mensaje de WhatsApp al teléfono. En la figura 88 se muestran imágenes de las notificaciones que llegaron al teléfono móvil del destinatario (en este caso el autor del proyecto) inmediatamente luego de haber una variación de temperatura, se visualiza las notificaciones ya configuradas.



Programación de Tareas con CRON

El comando **crontab** se utiliza en sistemas **UNIX** para programar la ejecución de otros comandos, es decir, para automatizar tareas. Podemos ver los crontab que se están programados y también editarlos, lógicamente.

Para verlos, utilizamos este comando:

sudo crontab -l

Para editarlos:

sudo crontab -e

Las tareas *cron* siguen una determinada sintaxis. Tienen 5 asteriscos seguidos del comando a ejecutar. De izquierda a derecha, los asteriscos representan:

Minutos: de 0 a 59.

Horas: de 0 a 23.

Día del mes: de 1 a 31.

Mes: de 1 a 12.

Día de la semana: de 0 a 6, siendo 0 el domingo.

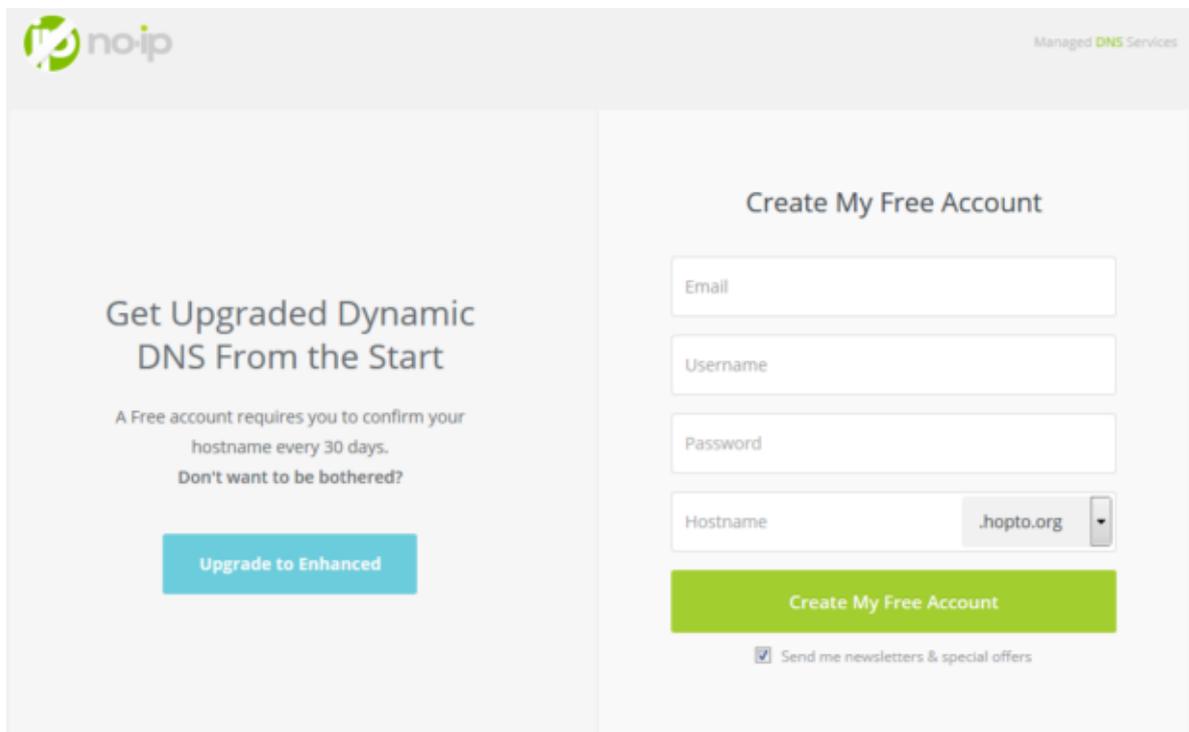
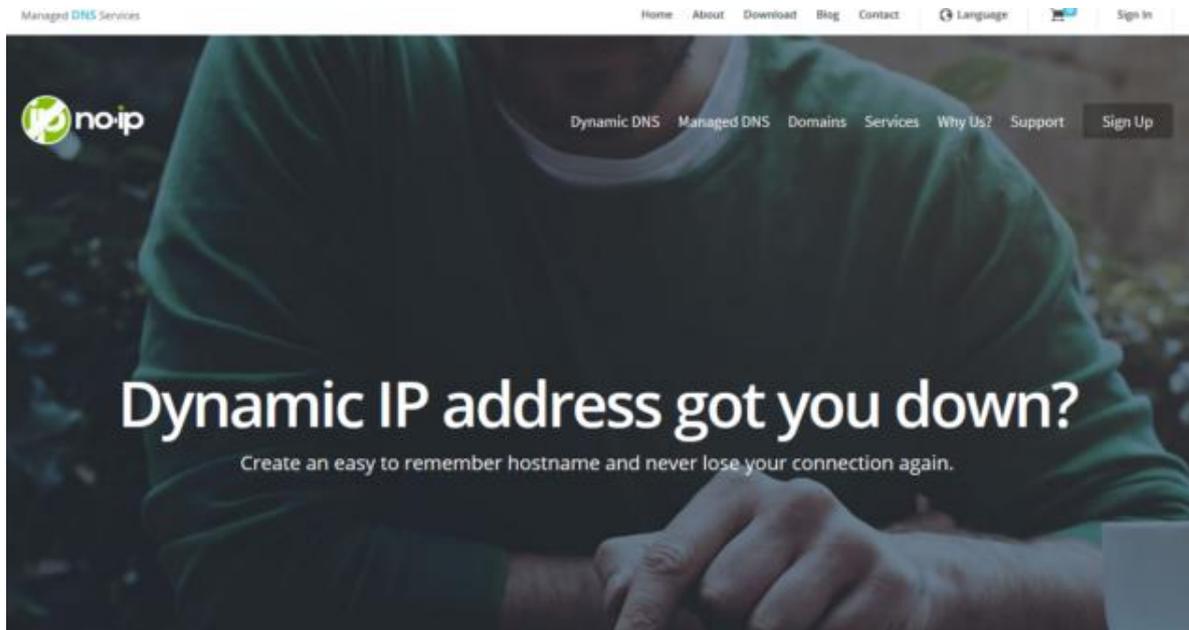
Por ejemplo:

```
30 2 * * 1 /bin/ejecutar/script.sh
```

Este script se ejecuta todos los viernes a las 2:30 horas.

ANEXO D Instalación y Configuración de NoIP.

Ingresamos al sitio <https://www.noip.com/> y nos registramos gratuitamente



Luego de haber registrado la cuenta, procedemos con la configuración de no-ip en el Raspberry Pi:

Accedemos a la Raspi mediante **ssh**, y antes que nada tenemos que actualizarla. Lo hacemos tecleando:

```
sudo apt-get update && sudo apt-get upgrade
```

Esto actualizará primero los repositorios del sistema y posteriormente lo actualizará. Dependiendo de cuánto haga que no se haya actualizado, o si es la primera vez, puede tardar un rato.

Una vez actualizada, es hora de instalar no-IP. Para ello:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

Extraemos el archivo con:

```
tar -zxvf noip-duc-linux.tar.gz
```

Y accedemos a la carpeta usando:

```
cd noip-2.1.9-1
```

Una vez dentro, procedemos a la instalación:

```
sudo make
```

```
sudo make install
```

Ahora nos va a preguntar el nombre de usuario y contraseña de no-IP y, una vez introducido, ya **cogerá automáticamente en dominio que hemos creado antes**. Luego nos quedará elegir el tiempo de refresco, que lo podemos dejar por defecto, y a la siguiente pregunta **respondemos «NO» (n)**. Ya tenemos no-IP configurado.

Creamos un nuevo archivo:

```
sudo nano /etc/init.d/noip2
```

Y copiamos dentro lo siguiente:

```
sudo /usr/local/bin/noip2
```

Ahora añadimos permisos de ejecución con:

```
sudo chmod +x /etc/init.d/noip2
```

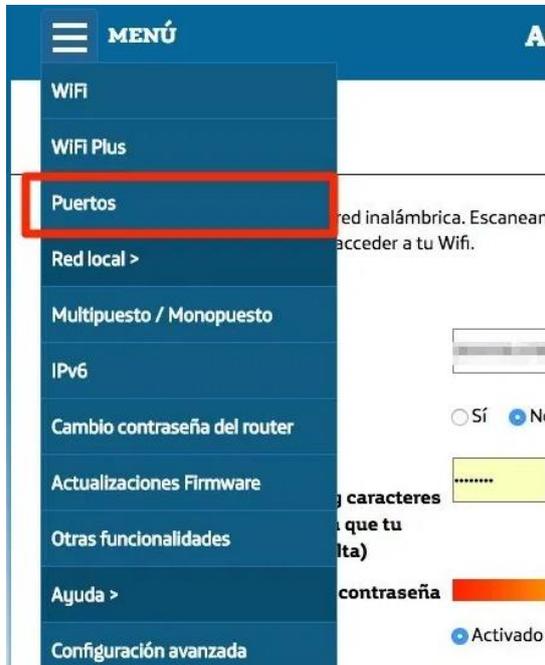
Modificamos el fichero update-rc:

```
sudo update-rc.d noip2 defaults
```

Y lo ponemos en marcha:

```
sudo /usr/local/bin/noip2
```

Luego abrir los puertos del Router



En la siguiente pantalla rellenamos los campos como sigue:

Puertos

Rellena los siguientes campos y pulsa el botón Añadir. Ten en cuenta que para abrir un rango de puertos debes usar el siguiente formato : 5001:5010

Nombre regla de puertos	<input type="text" value="SSH Raspi"/>
Dirección IP	<input type="text" value="192.168.1.14"/>
Protocolo	<input type="text" value="TCP-UDP"/>
Abrir Puerto/Rango Externo (WAN)	<input type="text" value="22"/>
Abrir Puerto/Rango Interno (LAN)	<input type="text" value="22"/>

El nombre es puramente informativo, pero la dirección IP debe ser la que le hemos asignado a nuestra Raspi (de ahí la importancia de que la pusiéramos estática), y los

campos de puerto ponemos el 22 (el puerto por defecto de las conexiones ssh). Añadir y listo.

Lo único que nos quedaría es probar desde fuera de casa si podemos hacer ssh, usando:

```
ssh pi@midominio.ddns.net
```

Y con esto ya tenemos nuestra Raspi más cerca de interactuar con el exterior de una forma mucho más madura.

Tabla 29. Historia de usuario 1

Historia de Usuario	
Número	1
Usuario:	Encargado del cuidado del bebe
Nombre Historia:	Monitorear la temperatura y humedad de la habitación del bebe
Prioridad en negocio:	1
Riesgo en desarrollo:	Medio
Puntos estimados:	5
Iteración	3
Asignada:	
Programador	Freddy Peñafiel
Descripción	El cliente desea poder supervisar la temperatura y grados de humedad de la habitación.
Observaciones	Para abaratar costos se utilizará hardware libre compatible con la placa ARDUINO R3

Tabla 30. Historia de usuario 2

Historia de Usuario	
Número	2
Usuario:	Encargado del cuidado del bebe
Nombre Historia:	Streaming en vivo de la habitación del bebe
Prioridad en negocio:	2
Riesgo en desarrollo:	Medio
Puntos estimados:	5
Iteración Asignada:	5 a 6
Programador	Freddy Peñafiel
Descripción	El usuario desea incluir en la solución el Streaming de lo que sucede en la habitación el mismo que será captado a través de una cámara.
Observaciones	Es muy difícil que una persona se encuentre todo el tiempo supervisando al bebe para ello utilizaremos una cámara PiNoir compatible con la placa Raspberry PI

Tabla 31. Historia de usuario 3

Historia de Usuario	
Número	3
Usuario:	Encargado del cuidado del bebe
Nombre Historia:	Monitorear la presencia de intrusos en la habitación del bebe.
Prioridad en negocio:	3
Riesgo en desarrollo:	Alto
Puntos estimados:	5
Iteración Asignada:	2
Programador	Freddy Peñafiel
Descripción	El usuario desea detectar el ingreso de individuos en la habitación del bebe.
Observaciones	Se precisa la utilización de sensores infrarrojos compatibles con la placa Arduino UNO R3 para la detección de personas.

Tabla 32. Historia de usuario 4

Historia de Usuario	
Número	4
Usuario:	Encargado del cuidado del bebe
Nombre Historia:	Recibir notificaciones de las alertas por mensaje de texto
Prioridad en negocio:	0
Riesgo en desarrollo:	Medio
Puntos estimados:	1
Iteración Asignada:	1
Programador	Freddy Peñafiel
Descripción	El cliente desea recibir las notificaciones de los eventos catalogados como negocios vía mensaje de texto.
Observaciones	Para realizar esto se explica al cliente que necesitará un modem y un chip que cuente con un paquete ilimitado de mensajes.

Tabla 33. Historia de usuario 5

Historia de Usuario	
Número	5
Usuario:	Encargado del cuidado del bebe.
Nombre Historia:	Recibir notificaciones de las alertas por correo electrónico.
Prioridad en negocio:	0
Riesgo en desarrollo:	Medio
Puntos estimados:	1
Iteración Asignada:	1
Programador	Freddy Peñafiel
Descripción	El cliente desea recibir las notificaciones de los eventos catalogados como riesgosos vía correo electrónico.
Observaciones	El correo electrónico no es un medio de comunicación rápida sobre todo cuando se trata de los correos personales.

Tabla 34. Historia de usuario 6

Historia de Usuario	
Número	6
Usuario:	Encargado del cuidado del bebe.
Nombre Historia:	Enviar notificaciones de las alertas por WhatsApp
Prioridad en negocio:	4
Riesgo en desarrollo:	Medio
Puntos estimados:	4
Iteración Asignada:	9
Programador	Freddy Peñafiel
Descripción	El cliente desea recibir las notificaciones de los eventos catalogados como riesgosos vía WhatsApp
Observaciones	WhatsApp, es un sistema de mensajería instantánea utilizado por gran parte de la población mundial, su implementación no conlleva inversión económica.

Tabla 35. Historia de usuario 7

Historia de Usuario	
Número	7
Usuario:	Encargado del cuidado del bebe
Nombre Historia:	Emisión de reportes de la información recopilada
Prioridad en negocio:	5
Riesgo en desarrollo:	Bajo
Puntos estimados:	5
Iteración Asignada:	8
Programador	Freddy Peñafiel
Descripción	Al cliente le gustaría poder generar reportes de las alertas presentadas.
Observaciones	Para lograr este objetivo se necesitaría contar con una base de datos para almacenar hechos históricos.

Tabla 36. Historia de usuario 8

Historia de Usuario	
Número	8
Usuario:	Encargado del cuidado del bebe.
Nombre Historia:	Monitorear la presencia de gases tóxicos en la habitación
Prioridad en negocio:	6
Riesgo en desarrollo:	Alto
Puntos estimados:	5
Iteración Asignada:	4
Programador	Freddy Peñafiel
Descripción	El usuario desea detectar la presencia de gases tóxicos en la habitación del bebe.
Observaciones	Se precisa la utilización de sensores MQ compatibles con la placa Arduino UNO R3.

Tabla 37. Historia de usuario 9

Historia de Usuario	
Número	9
Usuario:	Encargado del cuidado del bebe.
Nombre Historia:	Control de acceso de usuarios
Prioridad en negocio:	0
Riesgo en desarrollo:	Medio
Puntos estimados:	1
Iteración Asignada:	1
Programador	Freddy Peñafiel
Descripción	El usuario solicita el ingreso a la interfaz de monitoreo a través de código y clave para evitar el acceso externo.
Observaciones	Se determina que el ruido de una alarma podría hacer levantar al bebe y molestar a vecinos de alrededor

ANEXO F Modelo Entidad Relación

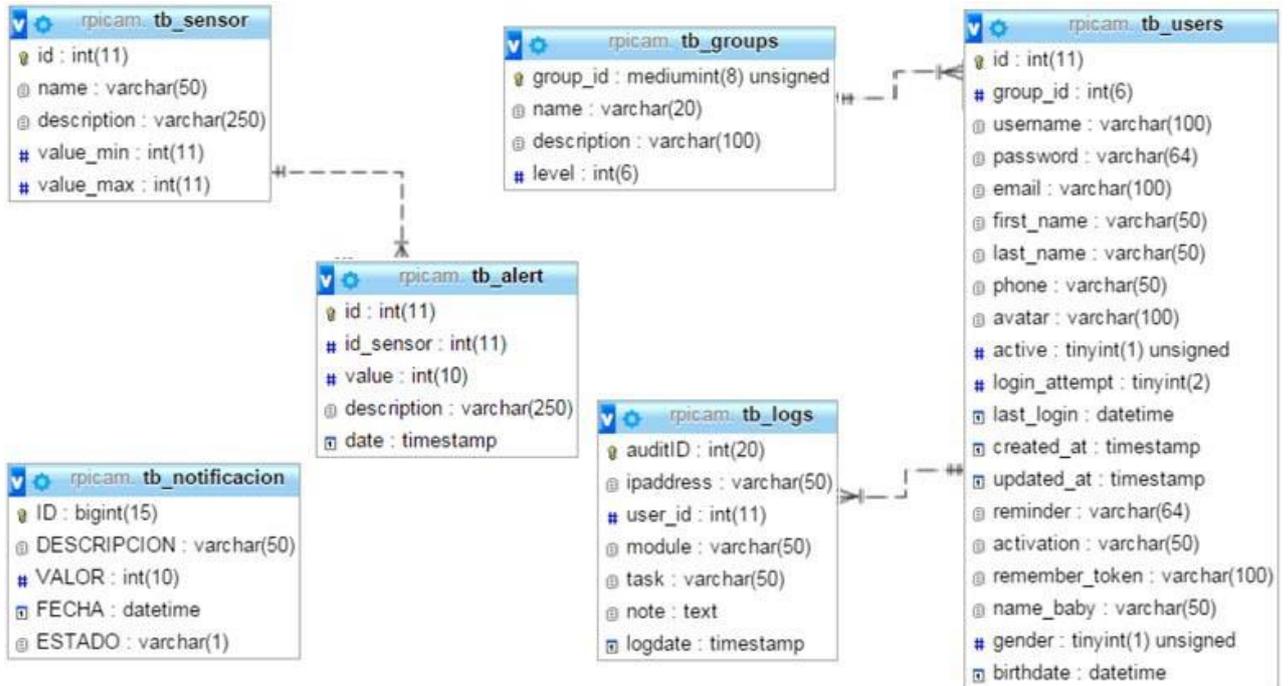


Figura 45. Modelo Entidad Relación

Anexo G Manual del Usuario

El entorno BEBECAM

Ha sido consecuentemente desarrollado en un entorno web intuitivo, amigable y fácil uso. La ventana principal es el lugar donde el usuario tiene la posibilidad de acceder a todas las opciones, alcanzando a realizar el monitoreo de la habitación en tiempo real y visualizar los reportes.

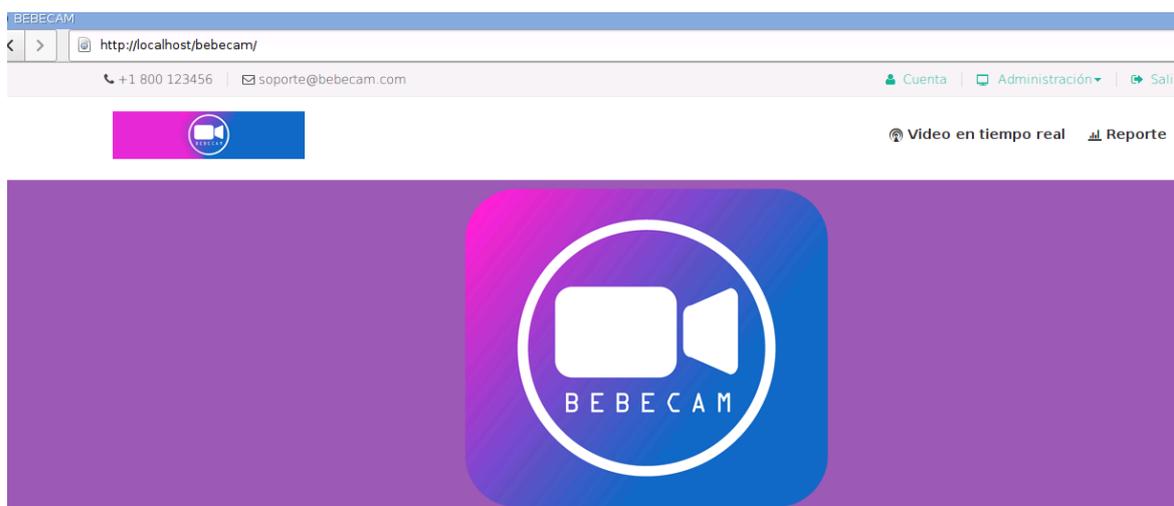


Figura 46. Pantalla Principal

En esta pantalla puede observarse:

1. Número de soporte a usuarios.
2. Correo de soporte a usuarios.
3. Menú **Cuenta**: Este menú contiene las opciones de perfil de usuario.
4. Menú **Administración**: Este menú contiene las opciones de mantenimiento, administración y configuración.
5. **Salir**: Terminar sesión.
6. Logo.
7. Wallpaper.
8. **Video en tiempo real**: Esta opción permite monitorear todos los eventos que se presentan en la habitación a través de una cámara.
9. **Reporte**: Esta opción permite consultar toda la información recopilada por los sensores.

El usuario podrá cerrar su sesión en el momento que desee, simplemente haciendo clic en “Salir”. El sistema cerrará la sesión del usuario y se mostrará la pantalla inicial de Login del sistema.

Interfaz de Inicio de Sesión

Es la pantalla inicial y se muestra al ingresar a la página web, utilizada para autenticarse con el ingreso de una cuenta de usuario y contraseña



BEBECAM

Usuario

superadmin

Contraseña

.....|

Ingresar

Crear cuenta

[Volver al sitio](#)

Figura 47. Interfaz de inicio de sesión

Pasos para seguir:

1. Abrir el navegador web de su preferencia.
2. Ingresar a la siguiente dirección web: <http://<<localhost:8080>>/bebecam>, Reemplazar “localhost” por la dirección IP asignada al Raspberry PI.
3. Ingresar usuario y contraseña.
 - El sistema verificará el usuario y contraseña ingresados, de ser correctos, el usuario ingresará al sistema y podrá acceder a las opciones según su perfil de acceso. En el caso de que los datos sean incorrectos mostrará un mensaje de error.

Interfaz de Registro de Usuarios

En esta pantalla se realiza el registro de usuarios nuevos, los cuales serán activados por el Administrador del sistema quién además se encargará de asignarles un perfil de accesos



La imagen muestra una interfaz de usuario para el registro de usuarios en un sistema llamado BEBECAM. El formulario está organizado en secciones con encabezados en negrita: 'Usuario', 'Nombre', 'Apellidos', 'Celular', 'Contraseña' y 'Confirmar contraseña'. Cada sección tiene un campo de entrada de texto con un ícono de validación a la derecha (un 'X' en un cuadrado para el usuario, un ícono de persona para nombre, apellidos y celular, y un candado para las contraseñas). Al final del formulario hay un botón verde con el texto 'Crear cuenta'.

Figura 48. Interfaz de ingreso de usuario

Pasos para seguir:

1. Ingresar a la opción “Crear cuenta” de la interfaz de inicio de sesión.
2. Registrar los datos principales de la cuenta:
 - a. Usuario: nombre de usuario.
 - b. Nombre: dos nombres.
 - c. Apellidos: dos apellidos.
 - d. Celular: numero de 10 dígitos.
 - e. Contraseña: máximo 10 caracteres.
 - f. Confirmar contraseña.

Menú de Administración

Encargado de registrar la información correspondiente a los sensores y grupos Además se cuenta con opciones de configuración, mantenimiento de usuarios y log de actividades. El usuario administrador es el que participa en este menú.

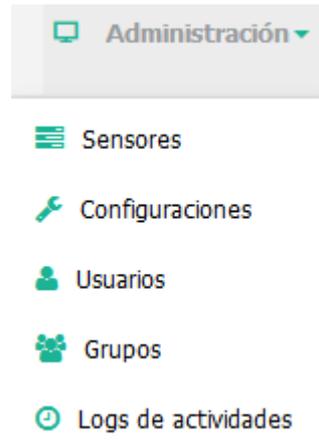


Figura 49. Menú de administración

En este menú se observa lo siguiente:

- Sensores. - esta opción permite definir los rangos de lectura de los sensores.
- Configuraciones. - Esta opción permite definir las configuraciones generales del sistema y la limpieza de la cache.
- Usuarios. - Esta opción permite definir los permisos de los usuarios y realizar el mantenimiento de estos.
- Grupos. - Esta opción permite definir los grupos de usuario y sus niveles de acceso.
- Log de actividades. - esta opción nos permite obtener un reporte de todas las actividades realizadas por los usuarios del sistema.

Menú Cuenta

Permite visualizar y modificar la información el usuario logueado. Además del ingreso de información adicional y cambio de contraseña.

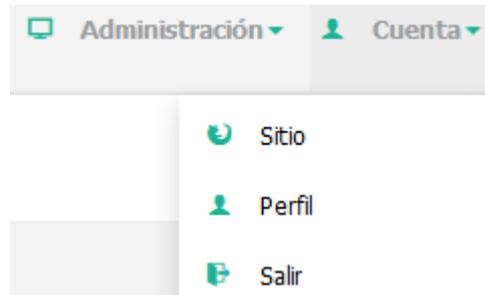


Figura 50. Menú cuenta

En este menú puede observarse:

- Sitio. - Esta opción que permite regresar al menú principal del sistema.
- Perfil. - Esta opción permite ingresar información adicional y cambiar la clave de usuario logueado
- Salir. - cerrar sesión.

Información personal

Esta opción permite modificar la información personal del usuario logueado.

A screenshot of a web application's 'Account' page. The page title is 'Account' with a subtitle 'Ver detalles de mi información'. Below the title is a breadcrumb 'Inicio / Cuenta'. There are three tabs: 'Información personal' (highlighted with a red dashed border), 'Datos del bebe', and 'Cambiar clave'. The 'Información personal' section contains three input fields: 'Usuario' with the value 'superadmin', 'Nombre' with the value 'Root', and 'Apellidos' with the value 'Admin'. A blue 'Guardar' button is located at the bottom right of the form.

Figura 51. Figura Información personal

Pasos para seguir:

1. Ingresar a la opción **“Perfil”** del menú **“Cuenta”**.
2. Ingresar a la etiqueta **“Información personal”**.
3. Modificar los campos de Nombre o Apellidos.
4. Dar clic en **“Guardar”**.

Datos del bebe

Esta opción permite ingresar o modificar la información del bebe.



The screenshot shows a user profile interface with three tabs: 'Información personal', 'Datos del bebe' (which is active and highlighted in green), and 'Cambiar clave'. The 'Datos del bebe' section contains the following fields:

- Nombre:** A text input field containing the value 'pepito'.
- Sexo:** A dropdown menu with 'Masculino' selected.
- Fecha de nacimiento:** A date and time input field showing '0000-00-00 00:'.
- Foto:** A button labeled 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'.
- Below the photo field, the text 'Imagen 80x80px' is displayed.
- A small square image placeholder showing a person's face.
- A blue button labeled 'Guardar' is positioned at the bottom of the form.

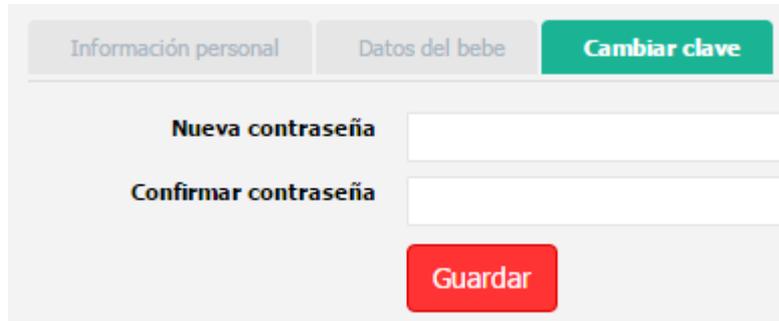
Figura 52. Figura Datos del bebe

Pasos para seguir:

1. Ingresar a la opción **“Perfil”** del menú **“Cuenta”**.
2. Ingresar a la etiqueta **“Datos del bebe”**.
3. Ingresar o modificar la información de los campos de Nombre, Sexo, Fecha de nacimiento y Foto.
4. Dar clic en **“Guardar”**.

Cambiar clave

Esta opción permite modificar la contraseña del usuario logueado.



El formulario muestra tres pestañas: 'Información personal', 'Datos del bebe' y 'Cambiar clave'. La pestaña 'Cambiar clave' está activa y contiene dos campos de texto etiquetados como 'Nueva contraseña' y 'Confirmar contraseña'. Debajo de los campos hay un botón rojo con el texto 'Guardar'.

Figura 53. Figura Cambio de clave

Pasos para seguir:

1. Ingresar a la opción **“Perfil”** del menú **“Cuenta”**.
2. Ingresar a la etiqueta **“Cambiar clave”**.
3. Ingresar y confirmar la nueva contraseña.
4. Dar clic en **“Guardar”**.

Sensor

Esta opción permite modificar los rangos de lectura de los sensores.

Sensor Cambiar límites de los sensores
Inicio / Sensor

+ Nuevo Copiar Eliminar Descargar Imprimir Configuración

#	<input type="checkbox"/>	Name	Description	Value Min	Value Max	Acción
	<input type="checkbox"/>					IR
1	<input type="checkbox"/>	PIR	SENSOR DE PRESENCIA	0	1	
2	<input type="checkbox"/>	MQ2	SENSOR DE GASES	0	1000	
3	<input type="checkbox"/>	DHT11	SENSOR DE TEMPERATURA	19	35	
4	<input type="checkbox"/>	DHT11	SENSOR DE HUMEDAD	29	51	

Mostrar Ordenar por Orden IR Mostrando : 1 a 4 de 4

Figura 54. Figura Lecturas de los sensores

Pasos para seguir:

1. Ingresar a la opción “**Sensores**” del menú “**Administración**”.
2. Ingresar a la opción “**Editar**”  del sensor que se desea modificar.
3. Modificar los valores de “**Value Min**” o “**Value Max**”.
4. Dar clic en la opción “**Guardar**”  del sensor modificado.

Ingreso y seguridad

Esta opción permite modificar los parámetros de seguridad.

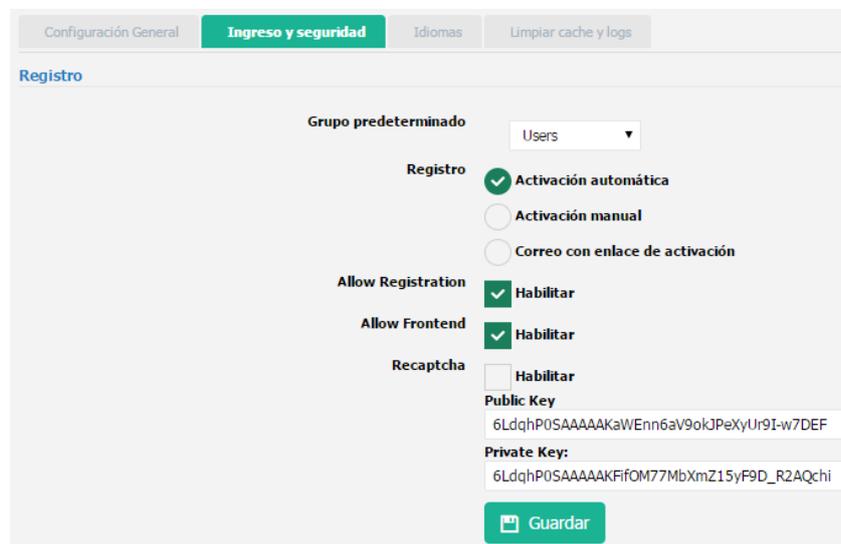


Figura 55. Figura Ingreso y seguridad

Pasos para seguir:

1. Ingresar a la opción “**Configuraciones**” del menú “**Administración**”.
2. Ingresar a la etiqueta “**Ingreso y seguridad**”.
3. Modificar los parámetros de los campos Registro, Allow Registration, Allow Frontend y Recaptcha.
4. Dar clic en la opción “**Guardar**”.

Limpiar cache y logs

Esta opción permite limpiar la cache del sistema.

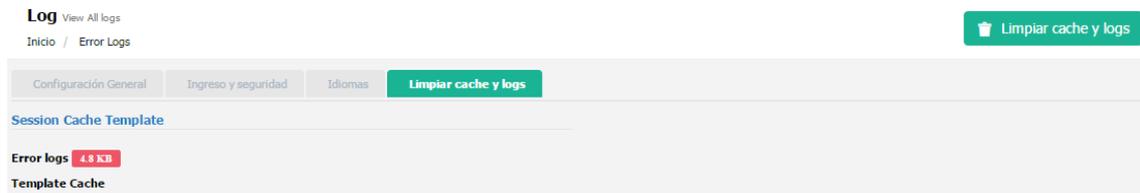


Figura 56. Figura limpiar cache y logs

Pasos para seguir:

1. Ingresar a la opción “**Configuraciones**” del menú “**Administración**”.
2. Ingresar a la etiqueta “**Limpiar cache y logs**”.
3. Dar clic en la opción “**Limpiar cache y logs**”.

Sistema de usuarios Users

Esta opción permite ingresar, modificar y eliminar usuarios. Además de agregar sus perfiles de acceso y activación.

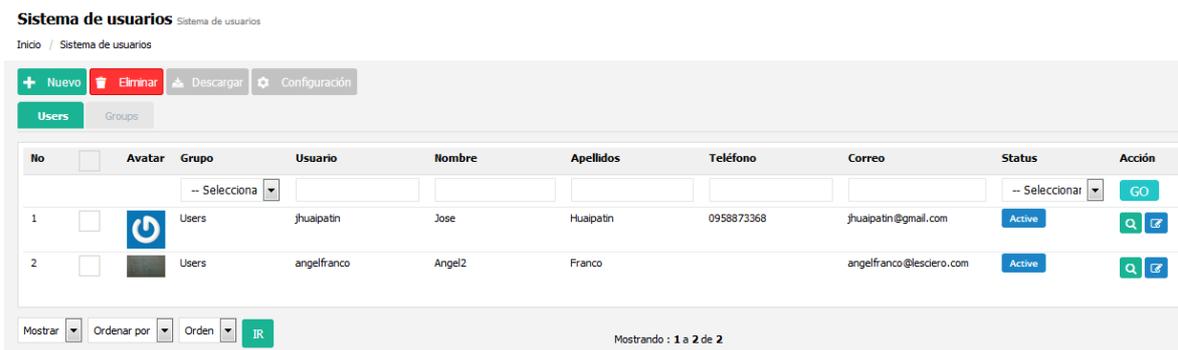


Figura 57. Figura Sistema de Usuarios

Pasos para seguir:

1. Ingresar a la opción “**Usuarios**” del menú “**Administración**”.
2. Ingresar a la opción “**Editar**” del usuario que se desea modificar.

3. Modificar el “**status**” para activar al usuario.
4. Dar clic en la opción “**Guardar**”  del sensor modificado.

Sistema de Usuarios Group

Esta opción permite ingresar, modificar y eliminar los grupos de usuarios.

Users Group View All
Inicio / Users Group

+ Nuevo Eliminar Descargar Configuración

Users **Groups**

No	<input type="checkbox"/>	ID	Name	Description	Level	Acción
1	<input type="checkbox"/>	1	Superadmin	Root Superadmin , should be as top level group	1	 
2	<input type="checkbox"/>	2	Administrator	Administrator level, level No 2	2	 
3	<input type="checkbox"/>	3	Users	Users as registered / member	3	 

Figura 58. Figura Sistema de Usuarios Group

Pasos para seguir:

1. Ingresar a la opción “**Usuarios**” del menú “**Administración**”.
2. Ingresar a la etiqueta “**Group**”.
3. Ingresar a la opción “**Editar**”  del grupo que se desea modificar.
4. Seleccionar el “**Grupo**” con los perfiles de acceso que se proporcionará al usuario.
5. Dar clic en la opción “**Guardar**”  del sensor modificado.

Logs

Esta opción permite auditar cada uno de los procesos realizados en el sistema.

Además, esta información puede ser descargada a Excel con la “**Descargar**”.

Logs Users Activity Log
Inicio / Logs

Eliminar Descargar Configuración

No	IP	Users	Module	Task	Note	Logdate	Acción
#	<input type="text"/>	-- Seleccionar -	<input type="text"/>	<input type="text"/>			GO
1	127.0.0.1	Root	logs	destroy	ID : 3 , Has Been Removed Successful	2014-11-27 20:49:22	
2	10.50.0.223	Root	sensor	save	New Entry row with ID : 5 , Has Been Save Successful	2016-03-17 20:23:00	
3	192.168.8.100	Root	sensor	save	ID : 2 , Has Been Changed Successful	2016-03-22 21:26:46	

Mostrar Ordenar por Orden IR

Mostrando : 1 a 3 de 3

Figura 59. Figura Logs

Video en Tiempo Real

Esta opción permite monitorear el entorno de la habitación. Además, cuenta con opciones de visualización que mejoran los ángulos de la imagen mostrada.

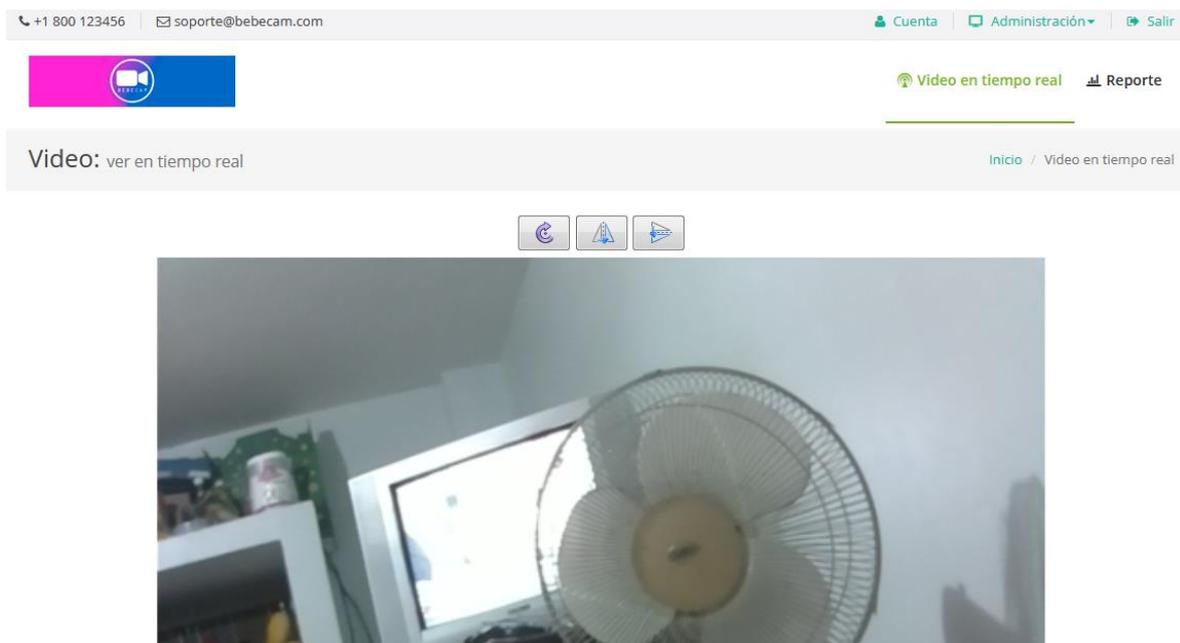


Figura 60. Figura Video tiempo real

Pasos para seguir:

1. Ingresar a la opción “**Video en Tiempo Real**” de la ventana principal.
2. Dar clic en la opción “**Rotación 90°**” , hasta obtener la imagen deseada.
3. Dar clic en la opción “**Ángulo Vertical**” , hasta obtener la imagen deseada.
4. Dar clic en la opción “**Ángulo Horizontal**” , hasta obtener la imagen deseada.

Reporte

Esta opción nos permite obtener un reporte detallado de la información recopilada por los sensores.

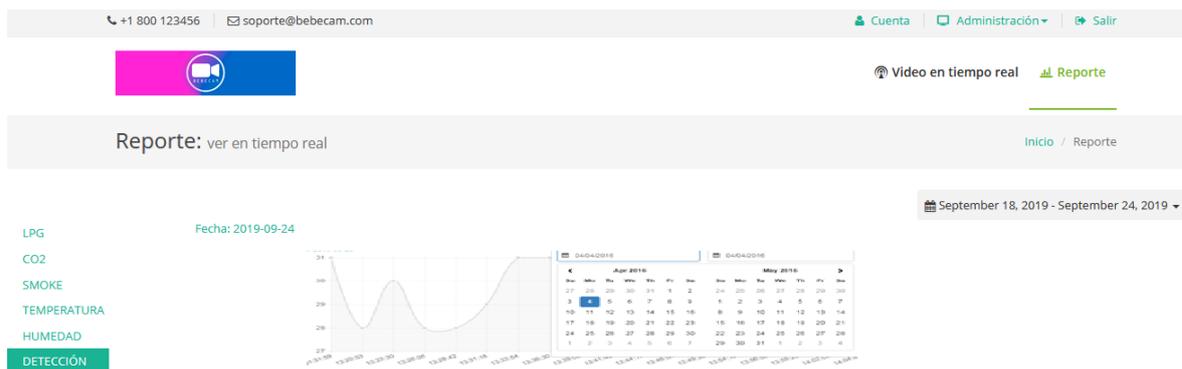


Figura 61. Figura: Reporte

Pasos para seguir:

1. Ingresar a la opción “**Reporte**” de la ventana principal.
2. Seleccionar el sensor que se desea consultar.
3. Seleccionar la opción de Rangos de fecha que se ajuste a sus necesidades:
 - a) Hoy: Información del día.
 - b) Ayer: Información del día anterior.
 - c) Últimos 7 días: Información de los últimos 7 días.
 - d) Este mes: Información del mes vigente.
 - e) El mes pasado: Información del mes pasado.

- f) Rango personalizado: Es necesario ingresar fecha de inicio y fin.
4. Dar clic en “**Aplicar**”.

REFERENCIAS BIBLIOGRÁFICAS

(s.f.). Obtenido de <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>

(s.f.). Obtenido de <https://es.wikipedia.org/wiki/WhatsApp>

(s.f.). Obtenido de <https://sg.com.mx/revista/open-hardware>

(s.f.). Obtenido de <https://www.biodic.net/palabra/prototipo/>

Aprendiendo Arduino. (s.f.). Obtenido de

<https://aprendiendoarduino.wordpress.com/2016/06/25/que-es-arduino-y-hardware-libre/>

Aprendiendo Arduino. (25 de Junio de 2016). Recuperado el 18 de Septiembre de 2018, de

<https://aprendiendoarduino.wordpress.com/2016/06/25/que-es-arduino-y-hardware-libre/>

Arduino, A. (s.f.). Obtenido de

<https://aprendiendoarduino.wordpress.com/2016/11/09/alimentacion-arduino/>

Arduino, A. (9 de Noviembre de 2016). *Aprendiendo Arduino*. Recuperado el 15 de Noviembre de

2018, de <https://aprendiendoarduino.wordpress.com/2016/11/09/alimentacion-arduino/>

Benito, I. (29 de Mayo de 2015). *Bebes Uno*. Recuperado el 17 de Septiembre de 2017, de

<https://bebes.uno/motorola-mbp11-opinion-y-analisis-de-este-vigilabebes/>

BioDic. (s.f.). Recuperado el 3 de Septiembre de 2018, de

<https://www.biodic.net/palabra/prototipo/>

concepto.de. (23 de 11 de 2018). Recuperado el 23 de 09 de 2019, de [https://concepto.de/software-](https://concepto.de/software-libre/)

[libre/](https://concepto.de/software-libre/)

David, M. (s.f.). *Definición de Open Source*. Recuperado el 29 de Mayo de 2018, de

<https://www.economiasimple.net/glosario/open-source>

DefiniciónABC tu diccionario hecho fácil. (s.f.). Recuperado el 18 de 09 de 2019, de

<https://www.definicionabc.com/tecnologia/mensajeria-instantanea.php>

Digital Guide IONOS. (2 de Agosto de 2019). Recuperado el 3 de Septiembre de 2019, de

<https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>

- DLink*. (s.f.). Obtenido de <http://www.dlink.com/es/es/products/dcs-8551-eyeontm-baby-monitor-hd-360>
- Eduardo, L. (23 de Marzo de 2015). *hetpro-store*. Recuperado el 12 de Abril de 2019, de <https://hetpro-store.com/TUTORIALES/sensor-dht11/>
- electronilab*. (s.f.). Obtenido de <https://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/>
- Electronilab*. (s.f.). Recuperado el 25 de Febrero de 2018, de <https://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/>
- Ernesto, L. (s.f.). Obtenido de https://www.sitiosargentina.com.ar/webmaster/cursos%20y%20tutoriales/que_es_la_mensaje_instantanea.htm
- Garcia, M., & Ramirez, S. (Julio de 2013). *ResearchGate*. Recuperado el 15 de Septiembre de 2017, de https://www.researchgate.net/publication/252627644_Integracion_de_OpenCv_en_una_Raspberry_Pi_Sistema_de_Deteccion_de_Rostro_via_WEB
- GNU*. (s.f.). Recuperado el 21 de Septiembre de 2017, de <https://www.gnu.org/philosophy/free-sw.es.html>
- González Godoy, C. A., & Salcedo Parra, O. J. (Mayo-Agosto de 2017). *UAEM redalyc*. Obtenido de <http://www.redalyc.org>: <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/851/1369>
- infortelecom*. (s.f.). Obtenido de <https://infortelecom.es/blog/que-es-un-servidor-y-para-que-sirve/>
- Infortelecom Te lanzamos al Cloud*. (29 de Septiembre de 2016). Recuperado el 27 de Enero de 2019, de <https://infortelecom.es/blog/que-es-un-servidor-y-para-que-sirve/>
- Juárez, G. (11 de Febrero de 2011). *Hipertextual*. Recuperado el 20 de Septiembre de 2017, de <https://hipertextual.com/2011/02/open-hardware>
- Juárez, G. (11 de Febrero de 2011). *Hipertextual*. Recuperado el 20 de Septiembre de 2017, de <https://hipertextual.com/2011/02/open-hardware>
- lenguajedeprogramacion*. (s.f.). Obtenido de <https://lenguajesdeprogramacion.net/python/>
- Luján, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante, España. Recuperado el Abril de 2018

maestrodeweb. (s.f.). Obtenido de <http://www.maestrodeweb.com/introspectiva-guido-van-rossum-python/>

Motorola Store. (s.f.). Recuperado el 17 de Septiembre de 2017, de <https://www.motorolastore.com/baby-monitors/video-monitoring/motorola-mbp33s.html>

Naylamp. (s.f.). Obtenido de https://naylampmechatronics.com/blog/40_Tutorial-sensor-de-temperatura-y-humedad-DHT1.html

Naylamp mechatronics. (s.f.). Recuperado el 19 de Enero de 2019, de https://naylampmechatronics.com/blog/40_Tutorial-sensor-de-temperatura-y-humedad-DHT1.html

neoattack. (s.f.). Obtenido de <https://neoattack.com/neowiki/lenguaje-de-programacion/>

NeoAttack. (5 de Abril de 2019). Recuperado el 17 de Marzo de 2019, de <https://neoattack.com/neowiki/lenguaje-de-programacion/>

Open Source. (s.f.). Recuperado el 20 de Septiembre de 2017, de <https://www.gpsos.es/soluciones-open-source/definicion-de-open-source/>

Pastor, J. (27 de Junio de 2016). *Xataka*. Recuperado el 4 de Septiembre de 2017, de <https://www.xataka.com/accesorios/las-13-mejores-ideas-que-hemos-encontrado-hechas-con-raspberry-pi>

Philips. (s.f.). *centralamerica*. Obtenido de https://www.centralamerica.philips.com/c-p/SCD603_00/avent-monitor-para-bebes-con-video-digital/caracteristicastecnicas

Repetí. (2007).

Robotec. (s.f.). Obtenido de <https://robotec.mx/products/sensor-gas-mq-2-butano-propano-hidrogeno>

Robotec. (s.f.). *Robotec MX*. Recuperado el 19 de Enero de 2018, de <https://robotec.mx/products/sensor-gas-mq-2-butano-propano-hidrogeno>

SG. (s.f.). Recuperado el 5 de Marzo de 2018, de <https://sg.com.mx/revista/open-hardware>

Tomalá Asunción, A. E. (7 de Marzo de 2019). Obtenido de <http://repositorio.ug.edu.ec/handle/redug/40641>

Torres Héctor, D. (s.f.). Obtenido de <https://hetpro-store.com/TUTORIALES/sensor-de-gas-mq2/>

Torres Héctor, D. (6 de Agosto de 2014). *HetPro*. Recuperado el 14 de Febrero de 2018, de <https://hetpro-store.com/TUTORIALES/sensor-de-gas-mq2/>

Unicef. (s.f.). *UNICEF*. Recuperado el 15 de 07 de 2017, de https://www.unicef.org/ecuador/spanish/children_5494.htm

Vicente, G. (s.f.). Obtenido de <https://www.diarioelectronicohoy.com/blog/sensor-hc-sr501-con-arduino>

Vicente, G. (9 de Noviembre de 2017). *EPA Electrónica Práctica Aplicada*. Recuperado el 12 de Diciembre de 2018, de <https://www.diarioelectronicohoy.com/blog/sensor-hc-sr501-con-arduino>

Vigila Bebés Ya. (s.f.). Recuperado el 15 de Septiembre de 2017, de <https://www.vigilabebesya.com/ghb/>

Wikipedia. (20 de Diciembre de 2018). Obtenido de <https://es.wikipedia.org/wiki/WhatsApp>