**UNIVERSIDAD AUTÓNOMA “GABRIEL RENÉ MORENO”**

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

CARRERA: INGENIERÍA ELECTROMECÁNICA



**“Monitorización de maquinarias mediante Raspberry para PyMes”**

IVAN MANUEL ROMERO GUTIERREZ

TRABAJO FINAL DEL CURSO DE CAPACITACIÓN PROFESIONAL MEDIANTE EXAMEN DE GRADO, PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTROMECÁNICO

**SANTA CRUZ – BOLIVIA**

**SEPTIEMBRE 2020**

**ÍNDICE**

**Introducción**

Resumen

Planteamiento del Problema

Justificación técnica

Técnica

Económica

**Objetivos**

Generales

Específicos

**Delimitación del trabajo**

Temporal

Geográfica

De contenido

**Metodología del Trabajo**

Descriptivo

Correlacional

**Hipótesis**

**Cronograma de trabajo**

1. **Marco Teórico**
   1. Monitoreo Continuo
   2. Competitividad en el mercado
   3. Rasberry
      1. Placas actuales en el mercado
      2. Hardware utilizado
      3. Sistema operativo
      4. Lenguaje de programación
   4. Arduino
      1. Placas actuales en el mercado
      2. Hardware utilizado
      3. Lenguaje de programación
   5. Labview
      1. Versiones
   6. Protocolos de comunicación
      1. Diferentes protocolos de programación
      2. Protocolo SSH
      3. Protocolo I2C
      4. Protocolo FTP
   7. Desarrollo del Hardware
      1. Sensores, actuadores y acondicionadores de Señal
         1. Sensor de Corriente no invasiva
            1. Acondicionador de Señal
         2. Sensor de Temperatura
         3. Sensor de Ruido
            1. Acondicionador de Señal
         4. Precio y Disponibilidad
      2. Conexiones entre los diferentes hardware
   8. Desarrollo del Software
      1. Sistema Operativo
         1. Instalación del SO
         2. Configuración del SO
         3. Instalación de dependencias
      2. Configuración del servidor
      3. Instalación de Base de datos
2. **Desarrollo final del proyecto**
   1. Cálculos
   2. Simulación
   3. Diseño
   4. Implementación
   5. Pruebas
   6. Análisis de Costo
3. **Conclusiones**
4. **Recomendaciones**
5. **Bibliografía**
6. **Anexos**

**INTRODUCCION**

Las diferentes empresas en Santa Cruz manejan múltiples tipos de maquinarias para sus trabajos y en muchos casos de manera continua, estos tienden a la obsolescencia ya sea de manera programada o natural, el tiempo de vida en este último caso se puede extender si tenemos un “*historial médico”* de la maquinaria.

Utilizando Tecnología *IoT*, el cual conecta el máximo de objetos posibles en un entorno programado, se monitorizará la maquinaria en cuestión y crear un historial para futuros inconveniente.

“*La prevención de accidentes no solo es una cuestión de legislación, sino un deber de los seres humanos, una razón de sentido económico (E. Werner von Slemens, 1816-1892)*”

El presente proyecto tiene como objetivo recolectar información a través de un sistema de monitorización en el cual utilizaremos un Raspberry Pi como servidor y un Arduino como transductor de diferentes señales representadas en múltiples variables sea corriente, voltaje, ruido, temperatura, vibraciones, etc.

Los principales objetivos son las pymes ya que estas cuentan muchas veces con pocas maquinarias pequeñas o medianas y dependen muchas veces exclusivamente de las mismas para realizar su trabajo y obtener sus beneficios, tener un técnico 24/7 suele ser costoso y en algunos casos hay que parar las maquinas por días mientras la reparan, tener un historial de los diferentes “*signos vitales*” puede ser muy ventajoso ya que un día perdido en una empresa se suele perder bastantes ingresos.

Aun en etapa de desarrollo esta versión constituye la primera de posibles futuras versiones del proyecto ( v 1.0.0), ya que se prevé contar con una placa más elaborada y su propio interfaz gráfica y no usar LabView en un futuro.

**Objetivos.**

**Objetivo General.**

Monitorizar de manera remota diferentes variables que suelen ser las más pertinentes en las maquinarias en relación al tiempo, utilizando una interfaz gráfica (LabView) para su respectiva monitorización remota de manera LAN, utilizando un Arduino con una placa amplificadora y un sensor sct013 para el manejo de corrientes no invasivas, Lm393 para un monitoreo del ruido y DHT11 para un monitoreo simple de temperatura y humedad, una placa Rasberry para un servidor remoto desde el cual podemos consular la base de datos.

**Objetivos Específicos.**

* Realizar un estudio exhaustivo sobre las diferentes equipos y maquinarias.
* Implementar censado automático y autónomo.
* Recopilar cierta cantidad de datos e interpretar.
* Construir una base de datos.

**Delimitación del trabajo**.

**Delimitación temporal**

El presente proyecto se desarrollará durante cuatro meses a partir de la aprobación de las misma por parte de la jefatura de carrera de Ing. Electromecánica, sin incluir los datos teóricos recolectados.

**Delimitación espacial o geográfica**.

El presente proyecto se realizará en la ciudad de Santa Cruz en la provincia Andrés Ibáñez.

**Delimitación de contenido**.

El presente proyecto se realizará tomando en cuenta lo aprendido en el diplomado, gran parte de la carrera como instrumentación, señales y sistemas, componentes de control y sistemas de comunicación, también se tomará cursos adicionales sobre programación en Python manejo de bases de datos, programación web, manejo de Linux y manejo de servidores.

**Metodología del trabajo**.

**Tipo de estudio.**

**Descriptivo**

El proyecto está dirigido principalmente al análisis de Datos y la recopilación de los mismos, para un futuro análisis y prevención.

**Correlacional**

Se puede apreciar un enfoque al sector PyMes, correlacionando sus gastos e identificando un patrón para todas las demás partes del sector.

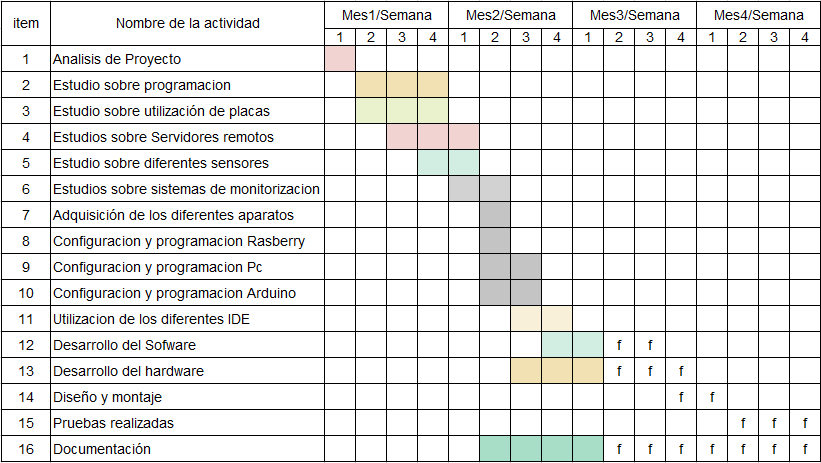
**Analítico**

Un enfoque hacia la toma de muestras y un futuro análisis de datos

**Hipótesis.**

Las empresas generalmente tienen entre una o varias maquinarias trabajando y esta representa su mayor inversión, muchas veces estas durante su reparación tienden a parar por días o incluso semanas hasta su reparación si llega a pasar, una de las finalidades es acortar ese transcurso de tiempo, por ello se recopilan los datos, para obtener un antecedente, histórico sobre la maquina en cuestión, con una pequeña inversión puede tener un beneficio como un *“historial médico de la maquina”*

**Cronograma de trabajo.**



*Tabla I.1 Cronograma de trabajo*

1. **Marco Teórico**
   1. **Sistema de monitoreo continuo**

Un sistema de monitoreo continuo no es un concepto nuevo ni mucho menos novedoso, en esencia es una técnica de mantenimiento preventivo.

*“El monitoreo de condición es el proceso de medir de forma periódica uno o más parámetros en la maquinaria, con el fin de identificar cambios significativos que usualmente indican fallas en el proceso, El monitoreo de condición es una parte imprescindible del mantenimiento predictivo y permite la planeación de acciones de mantenimiento enfocadas en evitar fallas y sus consecuencias”.*

Con el avance de la tecnología estas se vuelven mucho más accesibles lo módulos, placas y sensores, por lo que hoy en día podemos contar con las piezas para realizar nuestro propio sistema de monitoreo.

La adquisición de datos para tomar técnicas preventivas y predictivas cada vez se hace más importante, pero las pocas empresas que se dedican al monitoreo de máquinas actualmente son muy pocas en relación a la demanda del mercado, generalmente las empresas contratan sus propios técnicos para tomar acciones en el asunto y dedicarle tiempo para un mantenimiento temporal, aún con técnicos las probabilidades de fallas son mucho más altas que teniendo un historial de datos, ya sea por falta de personal (no todas las empresas pueden costear sueldo de técnicos) o por falta de preparación del mismo.

* 1. **Competitividad en el mercado**

Actualmente existen algunas empresas muy dedicadas al realizar maquinarias para un monitoreo continuo y generalmente se especializan en una o un par de tareas en específico.

Entre ellas tenemos *Erbessd instruments* que realiza máquinas de monitoreo específicos para mantenimiento predictivo.

De origen americano, se especializa en equipos de instrumentación y trabaja principalmente en maquinar rotativas, por lo que el sensor de temperatura, vibración y corriente son bastantes comunes.

También existe *National Instruments* que al igual que el anterior, se centra en la Instrumentación y es conocido por su principal producto, el *LabView*, que es un software potente y de sencillo uso para instrumentación en general, también hace máquinas de censado para el mantenimiento predictivo y tienes diversos tipos de sensores.

En Bolivia tenemos algunas empresas dedicadas al mantenimiento predictivo a través del censado continuo como lo es *Vibrobal*, empresa con centro en Cochabamba que trabaja con muchas otras marcas como ALL-TEST, ATTEN2,Eindrock, Des-Case, etc..  
Muchas de estas empresas están dedicadas a muchos ámbito, pero también destacan en el censado, para ello tienen diferentes artículos y productos que resuelven algún inconveniente en específico.



*Figura1.1 Pag.Principal https://www.erbessd-instruments.com/es/digivibemx/m30/*

* + 1. **ERBESSD INSTRUMENTS**

Con un precio variable de acuerdo a las funcionalidades que este contenga, el articulo con forma de Tablet, es especialmente diseñada para analizar vibraciones de diferentes maquinarias y guardar en su base de datos el valor y análisis de la recopilación de los mismos.

Figura 1.2 Información https://www.erbessd-instruments.com/es/digivibemx/m30/

El precio inicial es de 4800$ sin contar el envío ni los demás artículos relacionados a este.

Como ser los sensores, que se requieren de acuerdo a la funcionalidad del aparato.

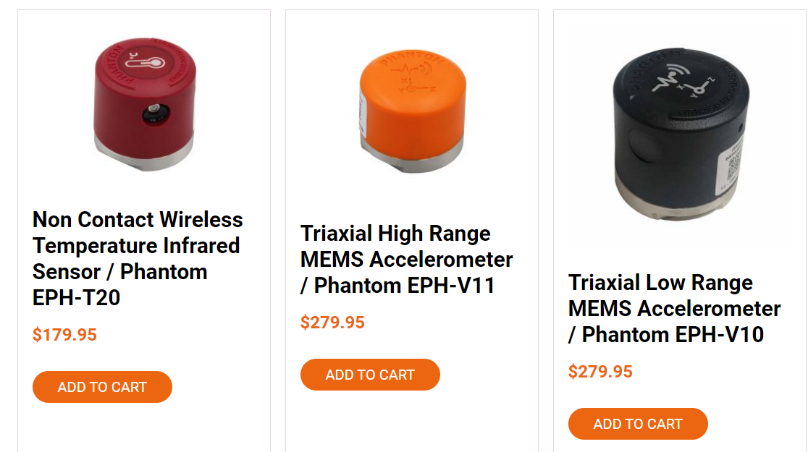


Figura 1.3 Sensores https://store.erbessd-instruments.com/products/monitoring/

* + 1. **National Instruments**

La marca americana especialista en software de instrumentación y su mayor insignia “*LabView*”

Figura 1.4 LabView https://www.ni.com/es-cr/shop/labview/select-edition.html

Siendo este un programa que se puede utilizar para realizar cálculos avanzados, simulación e interfaz gráfica.

Se le dará un pequeño uso como interfaz gráfica



Figura 1.5 Sensores NI https://www.ni.com/es-cr/shop/hardware/temperature-category.html#

Vendiéndose así los dispositivos por separados, su principal contribución es la solución en software.

* + 1. **Vibrobal**

Es una empresa boliviana más dedicada al mantenimiento, control y prevención, también siendo esta representante de empresas como ALL-TEST. ATTEN2 , etc.

Figura 1.6 Vibrobal https://www.vibrobal.com/index.php/productos

* 1. **Rasberry**

Es una mini computadora de bajo presupuesto desarrollado en Inglaterra por la fundación Rasberry Pi con el objetivo principal de estimular la enseñanza de las ciencias de la computación.

Se utiliza para muchos proyectos y es bastante popular en la actualidad, sus terminales GPIO logran hacer de estas muy utilizadas para la electrónica y domótica, el puerto ethernet para conexiones remotas y estaciones como mini-servidores

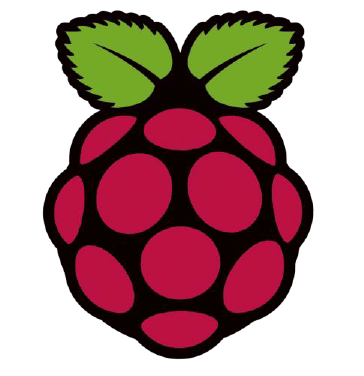


Figura 1.7 Logo Rasberry https://www.raspberrypi.org/

* + 1. **Placas actuales en el mercado**

Realizando una recopilación de datos sobre el mercado principal de electrónicos en Santa Cruz, las placas Rasberry que más comúnmente se encuentran son:

**Rasberry pi 3B**

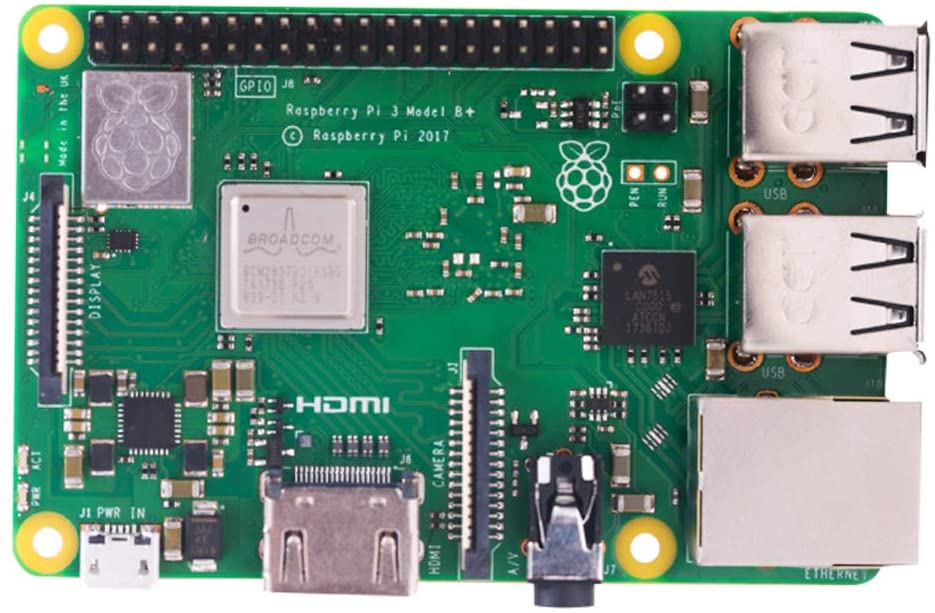
****

Figura 1.8 Rasberry Fotografía

Sus principales características son:

1GB RAM

Puerto Hdmi

Procesador BCM 2837

Entrada para cámara

4 USB 2.0

Puerto Ethernet

Dimension: 56mm x 85mm

Precio 400Bs

**Rasberry pi 4B**



Figura 1.9 Rasberry 4B

Sus Principales características son:

4GB de Ram

Puerto Hdmi micro

Procesador BCM 2711

2 Usb 3.0 y 2Usb 2.0

Puerto Ethernet

Dimensión: 56mm x 85 mm

Precio 700Bs

Utilizando el rasberry Modelo 2 o 3 el proyecto funciona con normalidad, he utilizado el modelo 4 pensando en actualizaciones y proyectos a futuro.

* + 1. **Hardware utilizado**

Aparte se comprará una fuente de Poder y una memoria para su debido uso, un monitor y teclado en caso no se tuviera, aunque este se controlará de manera remota, ya sea de modo LAN y WAN.

En cuestión a las memorias de 8GB es suficiente, el costo de las mismas en el mercado actual es de 35Bs. Dependiendo en el mercado de la localidad.

Si se piensa guardar imágenes o diferentes archivos como un servidor se puede ampliar a una de mayor capacidad.



Figura 1.10 Precio Memoria 4GB PCell (Mercado Local)

Las fuentes de poder se compran por separado en la misma tienda electrónica y tienen un valor de aproximadamente 70bs.

Los teclados y mouses más sencillos tienen un valor aproximado de 100bs ambos. Se recuerda que estas placas solo tienen entrada USB y Hdmi, solamente se comprará con este tipo de entrada para no gastar en adaptadores.

En caso de no tener pantalla alguna, monitor, tv, Tablet, etc. Es necesario comprar una con entrada Hdmi.

* + 1. **Sistema operativo**

El rasberry tiene diferentes SO (Sistemas Operativos) para poder instalarle, siendo sus principales representantes distribuciones de Linux, tales como, Pidora, Ubuntu, Arch, Debian etc.

Hay algunas que destacan para hacer proyectos electrónicos, tales como Rasbian y RetroPie.

**Rasbian**

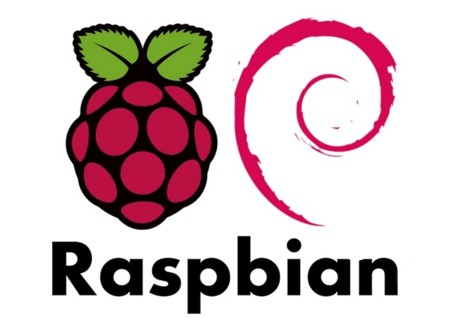
Es un sistema operativo libre basado en una distribución de Linux llamado Debian y Optimizado para el hardware Rasberry PI.

Figura 1.11 Rasbian https://www.raspbian.org/

**Ubuntu**

Es desde bastante tiempo una de las distribuciones más populares, Apto tanto como sistema operativo para el ordenador personal como para el servidor, este también se utiliza en las estaciones internacionales espaciales.

Figura 1.12 Ubuntu https://ubuntu.com/



**RetroPi**

Esta distribución está desarrollado por defecto sobre Rasbian y otros componentes de software, transforma el miniordenador en una consola con la que jugar a juegos clásicos.

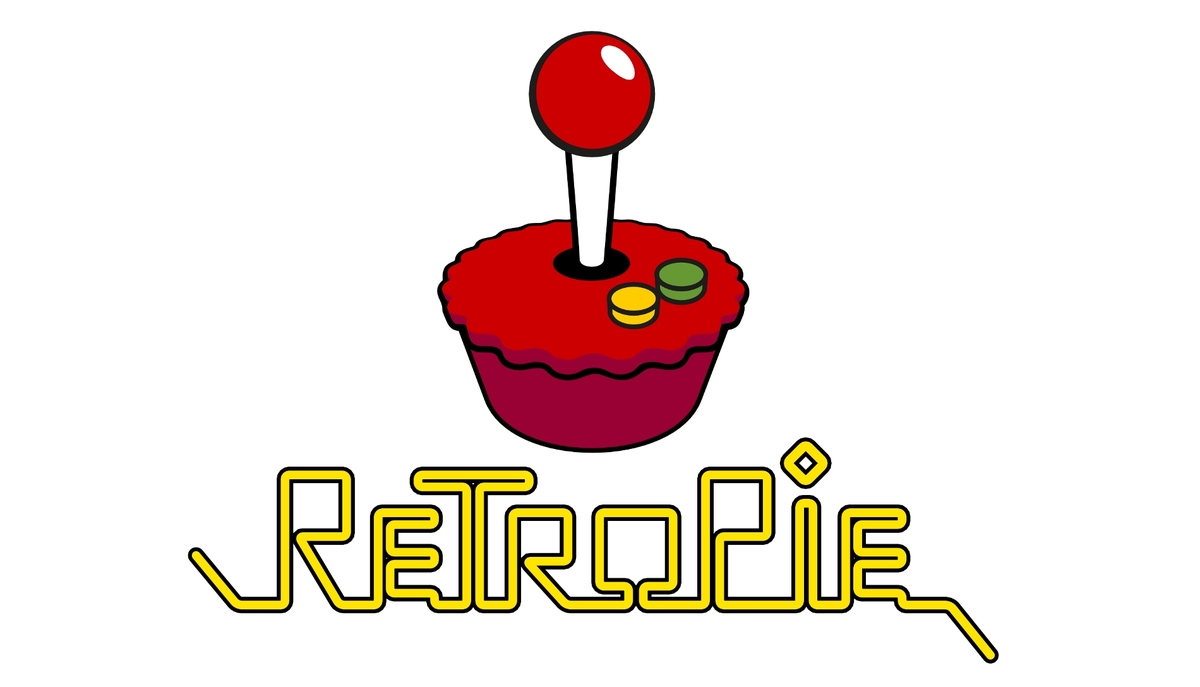


Figura 1.13 RetroPi https://retropie.org.uk/

Con estas menciones sobre Sistemas Operativos se optó por la que mejor quedará en relación al proyecto, *Rasbian* por su bajo peso y fácil uso para la electrónica en general.

* + 1. **Lenguaje de Programación**

Para realizar un buen programa necesitamos las bases de programación de un lenguaje preferible flexible, ya que el dispositivo lo adaptaremos para uso por LAN como servidor y utilizará el programa para contacto con el arduino, el leguaje principalmente utilizado para los programas es:

**Java**: Lenguaje Multiplataforma, aunque no se le da mucho uso en la placa Rasberry

**Ensamblador**: Bastante difícil de manejar, aunque al ser de bajo nivel el control sobre el dispositivo es prácticamente total.

**JavaScript**: Es un lenguaje bastante utilizado en distintas ramas de la informática, aunque principal uso es la programación WEB.

**Python**: Es el lenguaje por defecto de la placa, por lo cual es muy probable que la mayor parte de la programación se haga a través de este lenguaje

Aunque el proyecto también está enfocado a la Web y la IoT, también se pretende mejorar con métodos de Programación de aprendizaje, utilizando tecnologías libres y programas como *tensorFlow* este puede ser más a futuro, cuando el tiempo no sea tanta prioridad.



Figura 1.14 Python https://www.python.org/

* 1. **Arduino**

Arduino es una plataforma de Uso Libre, basado en “De fácil uso”, su principal función es para la enseñanza y para personas que no están familiarizados mucho en el mundo de la informática y electrónica.

* + 1. **Placas actuales en el mercado**

En la actualidad hay infinidad de placas y modelos en el mercado, el principal uso que se le dará es de transductor de señales análogas, ya que la placa Rasberry no tiene esta particularidad y al ser de bajo presupuesto y de fácil uso, las personas que quieran realizar su propio sistema de monitoreo continuo van a usar de manera más sencilla el sistema.

**Arduino Uno**

Existen de diferentes precios dependiendo de donde provenga el ensamblado, desde 55bs hasta 100bs.

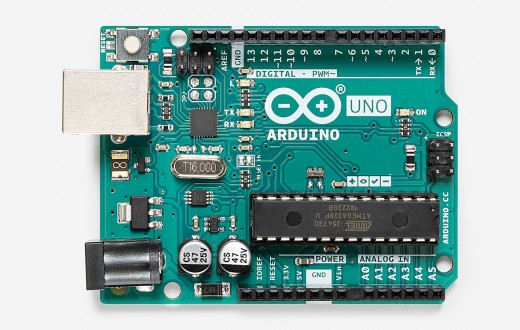


Figura 1.15 Arduino https://store.arduino.cc/usa/arduino-uno-rev3

**Arduino Leonardo**

**Características**:

Microcontrolador Armega32u4

Voltaje de entrada 7-12v

Voltaje Límite 20v

Pines digitales 20

Pines Analogicos 12

Canales PWM 7

**Precio 65 – 120bs**



Figura 1.16 https://store.arduino.cc/usa/leonardo

**Características**:

Microcontrolador Armega

Voltaje de entrada 7-12v

Voltaje Límite 20v

Pines digitales 20

Pines Analogicos 12

Canales PWM 7

**Precio 65 – 120bs**

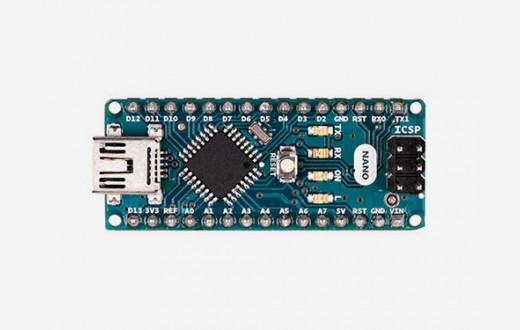


Figura 1.17 https://store.arduino.cc/usa/arduino-nano

**Características**:

Microcontrolador Armega328

Voltaje de entrada 6-10v

Voltaje Límite 5v

Pines digitales 22

Pines Analógicos 8

Canales PWM 6

**Precio 30 - 40bs**

Se escogió el arduino nano por su precio accesible y por sus características, se puede usar cualquier otro arduino similar tal que el uno, mini, etc.

El mínimo de canales analógicos para el proyecto es de 3, es la única observación en la tableta arduino a escoger.

* + 1. **Lenguaje de Programación**

El único lenguaje de programación que corre en su IDE (Entorno de desarrollo Integrado) de manera nativa es c++, *No es un C++ puro, sino es una adaptación que proviene de avr-lib que provee una librería de C e alta calidad para usar con GCC (Compilador c y C++) en los compiladores AVR de Atmel.*

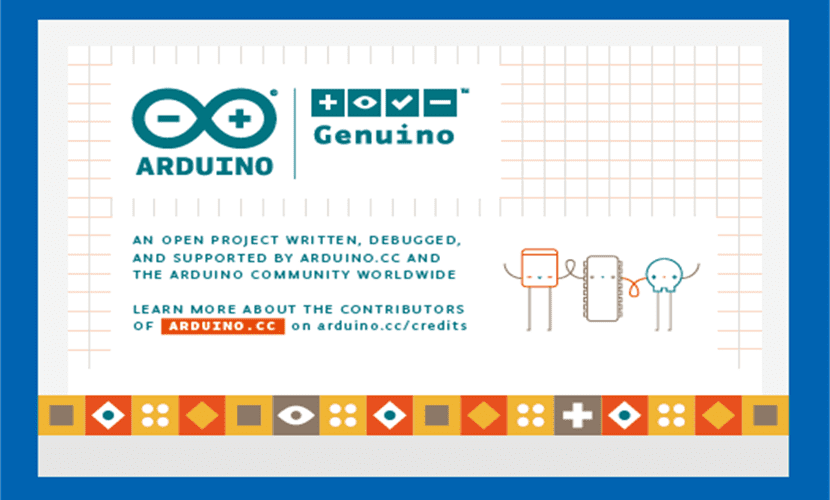


Figura 1.18 IDE Arduino https://www.arduino.cc/

* 1. **LabView**

Es una Plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual grafico pensado para sistemas de hardware y software de pruebas de control y diseño, simulado o real y embebido.



Figura 1.19 LabView https://www.ni.com/es-cr/shop/labview.html

Esta plataforma se utilizará para solamente hacer una demostración gráfica, no es necesario comprar la licencia, con la prueba gratis es suficiente para la demostración.

(Si se tiene certificación del programa se puede utilizar de manera continua)

* 1. **Protocolos de comunicación**

En informática y telecomunicaciones son una serie de normas, pasos e instrucciones que se deben dar para que se efectúe la comunicación entre 2 o más dispositivos, los programas que intervienen en la comunicación ponen orden, sin ellas se volvería caóticas y no podría dar como tal.

* + 1. **Diferentes Protocolos de Comunicación**

Existen muchísimos protocolos de comunicación entre ellas tenemos los que destacan en el proyecto.

* + - * 1. **I2C (Inter-integrated circuit)**
        2. **SSH (Secure Shell)**
        3. **SMB (Samba)**
        4. **FTP (Transmission Control Protocol)**
    1. **Protocolo SSH**

Su principal función es el acceso remoto a un servidor de un canal seguro en el que toda la información está cifrada.



Figura 1.20 SSH https://www.ssh.org/

El sistema trata de encriptar un archivo de tal manera que cada usuario tenga su propia clave de acceso, esta es enviada al receptor y esta vuelve a encriptar el archivo, retorna al primer usuario así des-encriptando el archivo reenviando el documento, este vuelve al receptor y des-encripta.

Es bastante seguro, aunque tiene ciertas vulnerabilidades, se utiliza en servidores Linux y existen 2 versiones de la misma.

La primera, por el hecho de que algunas patentes han expirado es menos segura pero más extendida, y la segunda específicamente para la distribución Red-Hat en el cual ya viene instalado por defecto, esta última es más segura que la primera versión

* + 1. **Protocolo I2C**

Se utiliza principalmente para la comunicación entre partes de circuitos electrónicos (generalmente placas), por ejemplo, entre un controlador y circuitos periféricos integrados.

Utiliza un sistema Maestro-Esclavo en el cual la transferencia de datos será iniciada por un maestro, el esclavo reacciona ante el impulso.

Puede ser una conexión muchos-a-muchos en el cual se puede comunicar varios maestros con un esclavo, un maestro con un esclavo, un maestro con muchos esclavos y muchos maestros con muchos esclavos.

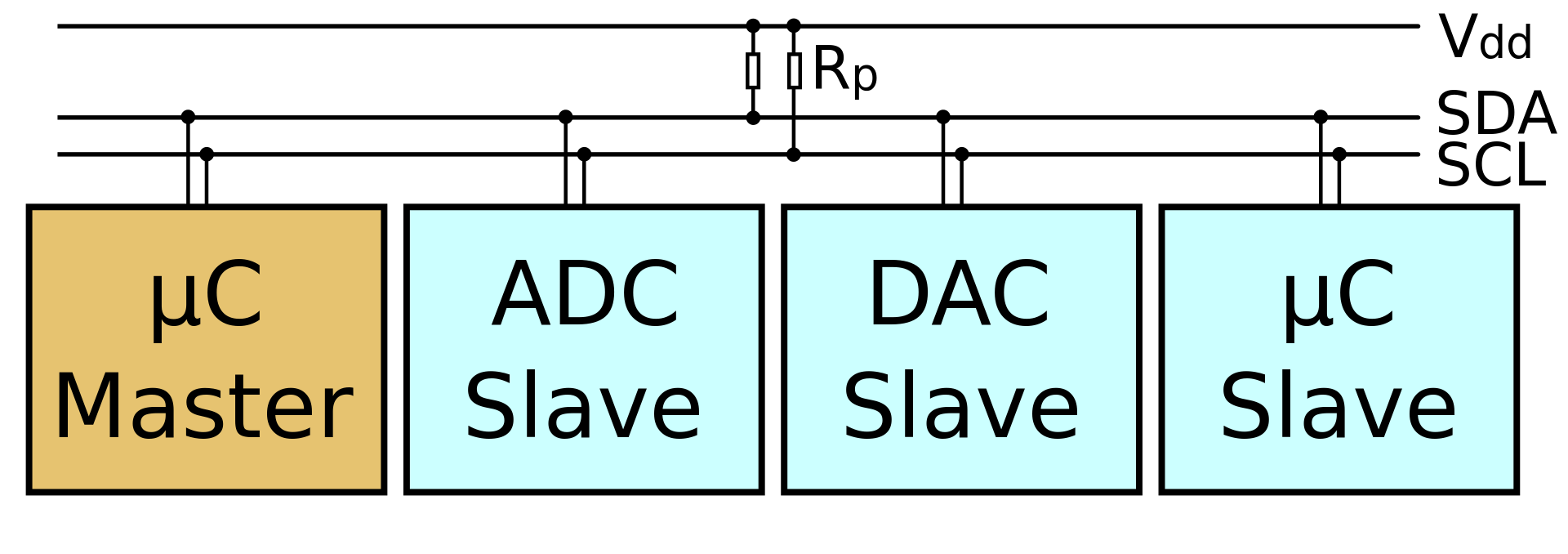


Figura 1.21 esquema I2C https://es.wikipedia.org/wiki/I2C

El protocolo del I2C es por definición bastante simple, pero muy susceptible a interferencias. Este hecho limita su uso únicamente a entornos de poca interferencia.

* + 1. **Protocolo FTP**

Se utiliza para la transferencia de archivos de unos sistemas conectados a través de la red WAN, al ser bidireccional se pueden hacer transferencia de archivos Cliente – Servidor o viceversa, este protocolo es parte de la familia de protocolo TCP-IP del cual no se hará especial mención.

El Cliente envía una orden, el servidor lo recibe y ejecuta ordenes, esta contiene 2 canales, un canal de órdenes y un canal de datos.

El Pi o interprete de protocolo controla mediante los comandos recibidos del cliente o enviados por el mismo.

El DTP ejecutara la petición sea enviar o recibir archivos.

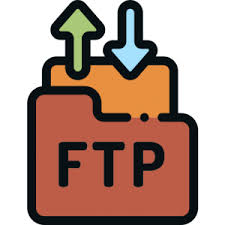


Figura 1.22 https://rubrika.es/marketing-digital/glosario/que-es-ftp/

Principalmente se utiliza el protocolo I2C para la conexión de distintos aparatos sea el arduino con Rasberry, Arduino con las placas alternas a este, Rasberry con pantalla 6” u 8” entre otras.

Muchos los protocolos que tienen que ver con la WEB.

* + - * 1. **ARP**
        2. **FTP**
        3. **HTTP**
        4. POP
        5. Telnet
        6. **DNS**
        7. RIP

Varias veces son automáticas cabe mencionarlas ya que se utilizan por si se quiere un mejor control y acceso al sistema.

También existe comunicación sin protocolo ya que simplemente es la lectura de una terminal o de un sensor sencillo de tal manera que se pueda realizar el trabajo de la mejor manera y con menor tiempo posible.

* 1. **Desarrollo del Hardware**

Principalmente accionado por Sensores y placas el Hardware está desarrollado para su facilidad y practicidad.

El proyecto se enfoca a un público objetivo del cual no ha desarrollado su capacidad o estudios sobre la electrónica o programación, se utiliza circuitos pre-diseñado y sensores que se encuentran en el mercado actual, para que estas sean lo más sencillas de conseguir y ejecutar el sistema para así poder tener un propio sistema de monitoreo continuo.

Con el tiempo se espera poder desarrollar más el sistema contando con un mayor número de entradas y otras cualidades.

* + 1. **Sensores, actuadores y acondicionadores de Señal**

Los sensores son uno de los pilares principales del proyecto, estos se encargan como se indica de censar datos y transformarlo en señales eléctricas que puedan ser manejadas por una terminal del Arduino.

Si alguna señal no llega al estándar que maneja estas terminales, esta se acondiciona de acuerdo a las exigencias de la tableta de programación.

Los actuadores es un añadido de control extra que se puede omitir, consta de relé, motores, switches con el cual podemos controlar de manera remota, este se puede condicionar a un programa automático o manual dependiendo del usuario final.

* + - 1. **Sensor de Corriente no invasiva (AUMENTAR)**

La familia SCT-013 es una agrupación de sensores de corrientes no invasivas, el cual consta de una bobina, resistencia interna y su capsula, la principal función de este tipo de sensor es convertir una corriente eléctrica o voltaje en una señal proporcional en esta.

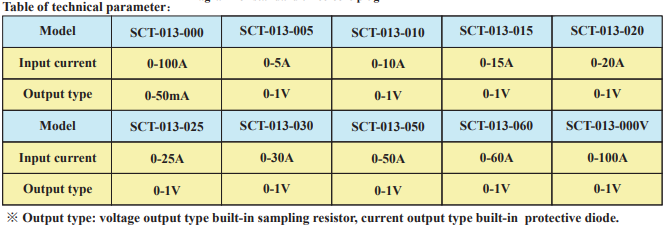
A continuación, se verá la relación entre tensión o corriente del cable comparado con la señal:

Tabla 1.00 Sensor A https://www.mcielectronics.cl/website\_MCI/static/documents

El sensor que se utiliza en el prototipo es específicamente SCT-013-050. El cual maneja valores entre 0 y 50A de entrada y arroja valores de 0 a 1V volteo de salida correspondientemente a los valores máximos y mínimos del sistema.

Para escoger el sensor a utilizar se utilizó como referencia un grupo de máquinas que puede consumir como máximo 30A en un anterior proyecto.

* + - * 1. **Acondicionador de Señal**

Se puede intuir que la señal entregada por el sensor al tener un valor entre 0 y 1V necesita amplificarse, además hay que tomar en cuenta que el resultado que arroja no es lineal con un valor de dispersión de ±3%, lo cual significa que, para valores muy pequeños de corriente, que estén rondando el doble de ese porcentaje es el mínimo aceptado para intentar hacer alguna medición.

El rango que acepta la entrada análoga del arduino ronda entre los 0 – 5v y una resolución de 10 Bits lo que proporciona 1024 niveles digitales el cual supone una precisión de la medición de ±2,44 mV.

Siento este el valor mínimo que soporta arduino para que la entrada tenga un valor diferente de 0.

Si la razón de cambio de los valores es de un 3% significaría que habría una dispersión de 0.75A estadísticamente trabajando con 2N para una mejor fiabilidad mínima aceptada, nos daría un resultado de 0.03V el cual nos indica que 30mV es el valor mínimo censado para el cual la dispersión no le afectaría demasiado su fiabilidad.

30mV implica 1.5A de corriente que se necesita como mínimo aceptable del cable que se intentará buscar valores.

El valor máximo con arduíno es de 5V para 1024 valores mientras que del sensor es de 1V, siendo esta un quintil de la población a evaluar, entonces el valor obtenido y computado oscilará entre 0 y 205 bits.

Para tener mejor precisión sobre lo datos se mejora los valores máximos hasta 1024 bits de información, para ello amplificar el valor máximo a un mínimo de 3.3v aunque lo mejor sería manejar un valor igual o lo más cercano a 5V para intentar utilizar los 1024 bits de información que utiliza el procesador de la placa para mejorar la precisión de manera digital.

Para ello utilizar una placa puede ser la mejor opción, aunque también se puede manejar AOP, pero este es un poco complicado de usar para la gente común y el objetivo del proyecto es que se puedan armar su propio sistema de monitoreo continuo de una forma económica y eficaz.

**PLACA ADS1115 (CAMBIAR)**

Esta placa de amplificador simplemente se conecta la alimentación, retorno y señal como indica.

Figura 1.23 conexión placa ADS1115

Esto nos ayudará como se mencionó anterior mente con el acondicionamiento de señal.

* + - 1. **Sensor de Temperatura**

Utilizando casi cualquier sensor bajo la lógica siguiente.

Arduino = 10bits = 1024 valores

Sensor (en volteos) = x V

Resolución

Eso significa que cada bit de información almacena un valor de x V.

En el caso del lm35 sacando la resolución no dará:

En conclusión, tenemos una razón de cambio de 3 bits por cada grado Celsius registrado.

Un valor de 1024 (0011 1111 1111) = 10 bits = 5v = 150ºC

Un valor de 312(0001 0100 0001) = 1.52V = 45,6ºC

Estos valores pueden oscilar por diferentes motivos, ya sea por ruido de la misma fuente de alimentación o por voltaje de entrada diferente de 5V.



Figura 1.24 Sensor Lm35 Datasheet

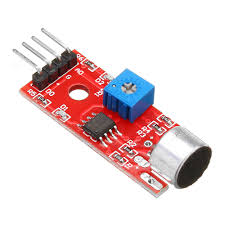
* + - 1. **Sensor de Ruido**

Típicamente llamado también micrófono, es uno de los sensores más importantes e interesantes para colocar en un proyecto de estas índoles.

las maquinas tienden a sonar de una forma armónica, este sonido cambia con el tiempo y podemos registrar su cambio, muchos mecánicos a lo largo de la historia han reparado maquinar guiándose por este parámetro.

Se puede usar un micrófono y un sistema de amplificación con AOP mas estas darían muchas veces problemas y confusión a la hora de hacer el proyecto para las personas que no entienden bien sobre electrónica y programación, entonces utilizando una placa o modulo pre-fabricado nos resultará más sencillo.

figura 1.26 Sensor KY – 037 Datasheet



El sensor de sonido KY-037 tiene un costo de 30bs. Y con este podremos registrar los niveles sonoros que necesitemos.

* + - * 1. **Acondicionador de Señal**

La señal emitida por un micrófono generalmente es muy pequeña, este necesita e un pre-amplificador y amplificador, también dependiendo de donde se coloque necesitamos elevar el nivel de la señal al semi-ciclo positivo, en el caso del arduíno por ejemplo ya que las señales analógicas que este recibe por sus compuertas no admite valores negativos de señal, muchas veces se necesita un filtro en serie para filtrar las señales continuas.

En algunos circuitos el preamplificador y el amplificador se encuentran en un mismo chip o en otro caso el AOP puede admitir señales débiles y no es necesario un pre amplificado.



Figura 1.27 Amplificador y Preamplificador de sonido.

* + - 1. **Precio y Disponibilidad**

En el mercado existen muchos sensores y dispositivos de censado junto con transductores.

En el circuito usaremos 3 de ejemplos de distintas mediciones, más se puede utilizar casi cualquier sensor siempre y cuando estén dentro de los parámetros aceptados, en el sensor de temperatura se pensaba usar un pt100 ya que es el más usual pero no estaba disponible en el mercado en ese momento, y no tenía tiempo para buscar ya que las defensas y tutorías solo duraban un par de semanas, usaremos el más barato y común para estudiantes lm35. El precio oscila entre 5 - 7bs.

El sensor de sonido o micrófono usaremos una placa KY- 37, es un micrófono con su etapa de amplificado, su valor esta entre 30 y 35bs.

El sensor de corriente no invasiva usaremos el SCT 013 050 ya que se quedó de un anterior proyecto fallido y con el motivo de reciclar electrónica usaremos este sensor.

|  |  |
| --- | --- |
| Sensor | Precio |
| Temperatura LM35 | 8bs |
| Sonido | 35bs |
| Vibración | 7bs |
| Total | 50bs |

Tabla 1.2 Sensores y Precios

* + 1. **Conexiones entre los diferentes hardware.**

Sobre las conexiones se usarán distintos puertos y métodos de comunicación, para la comunicación entre arduino y rasberry se hará por medio de cable USB, los sensores se pensaba en un principio que se utilizaría puertos Jack macho para los sensores y hembras para el arduino, creando una placa de conexión para los sensores o los acondicionadores de señal en defecto.

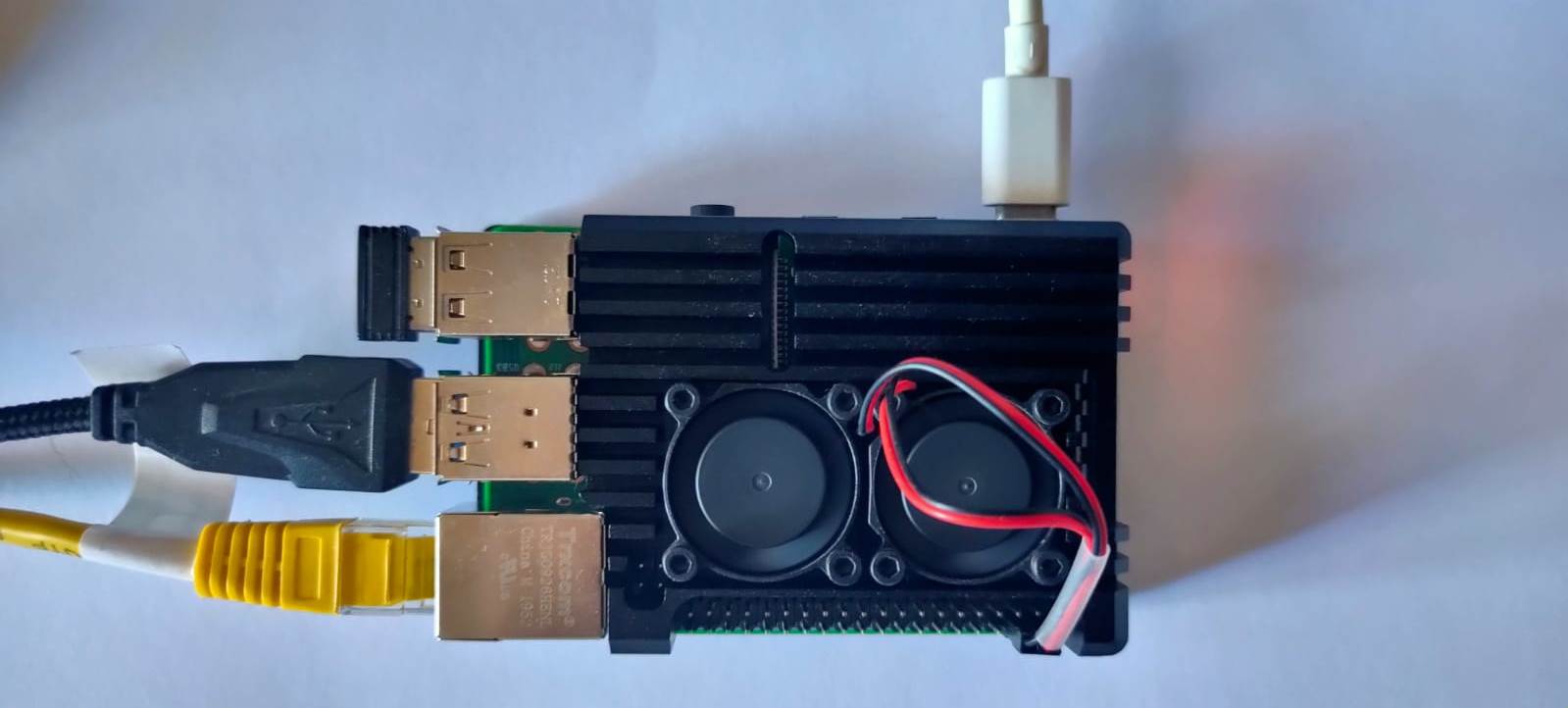


Figura 1.28 Rasberry pi4

La alimentación será extraída del Rasberry y una fuente de 5V 5A que estará conectada en el rasberry.

La red estará conectada al Rasberry mediante cable RJ45 o mediante conexión inalámbrica.



Figura 1.29 Rasberry conexiones laterales

La conexión de vídeo es mediante hdmi y micro hdmi dependiendo del modelo de rasberry que se use, la entrada de audio mediante Jack, estas entradas nos servirán al principio para las configuraciones iniciales, luego se realizará las demás mediante conexión remota.



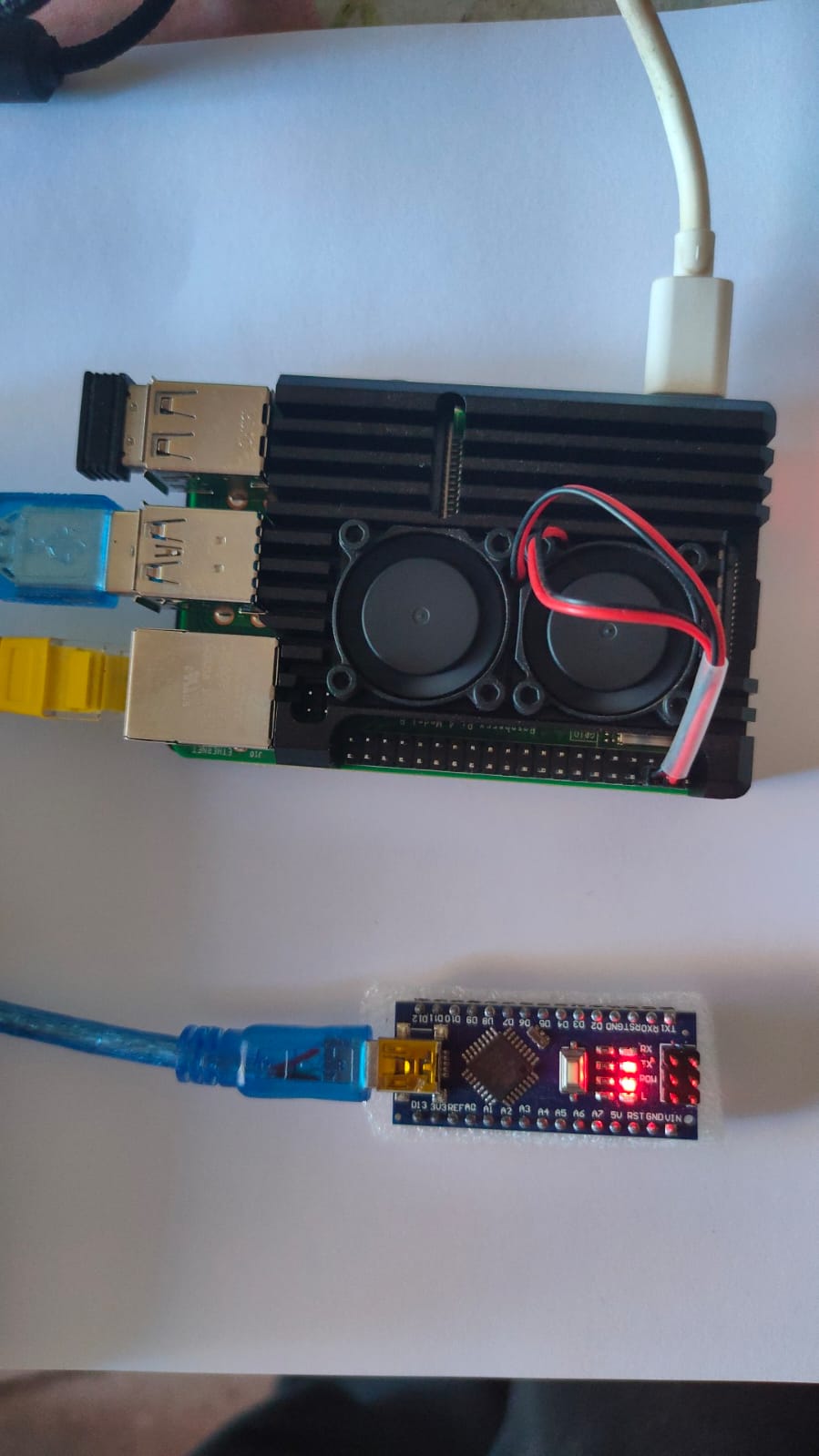
Figura 1.30 entrada frontales



Figura 1.31 entradas GPIO

Las entradas GPIO se pueden conectar tanto sensores como actuadores, más estas son digitales, asique necesitaremos un transductor para las señales analógicas.

Figura 1.32 Conexión Rasberry - Arduino



El arduino se utilizará como transductor de señales analógicas y estará conectado mediante cable USB.

* 1. **Desarrollo del software**

El software es tan importante como el hardware, para algunas placas que son análogas no necesitaremos instalar un SO o algún lenguaje de programación, pero para el rasberry es necesaria tanto la instalación de dependencias como la instalación de sistema operativos y todos los utilitarios para que el programa pueda funcionar por la red.

* + 1. **Sistema Operativo**

Para Programar en una placa como Rasberry es necesario indicarle el SO a usar, como se ha mencionado anteriormente este posee la capacidad de albergar uno o varios, cabe resaltar que, si no hubiera rasberry en casi cualquier pc, con esta distribución de Linux igual funcionaria.

* + - 1. **Instalación del SO**

Para la instalación del SO necesitaremos tener un disco duro que en el caso del rasberry se utilizar una tarjeta micro SD para la instalación del SO y para la guardar las configuraciones, un problema que conlleva es que al estas no estar hechas para que funcionen como tal todo el tiempo, es posible que las esperanzas de vida de estas unidades de memoria se vean mermadas en un plazo de 1 año a 2 después de ser utilizadas como SO.

Para la instalación del SO es necesario seguir estos pasos

Se busca en la página oficial la descarga del SO.

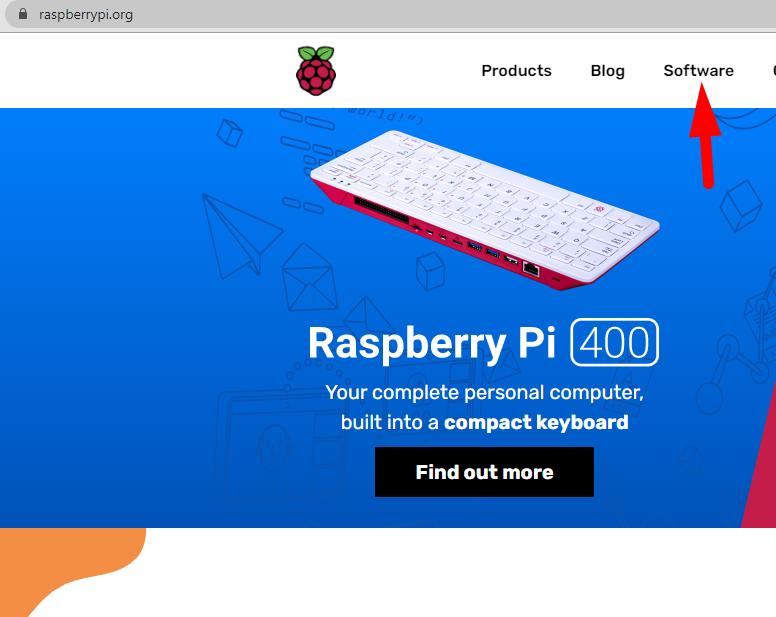
Buscar en la página oficial el Software para la instalación

Figura 1.33 Pag. Oficial Rasberrypi.org

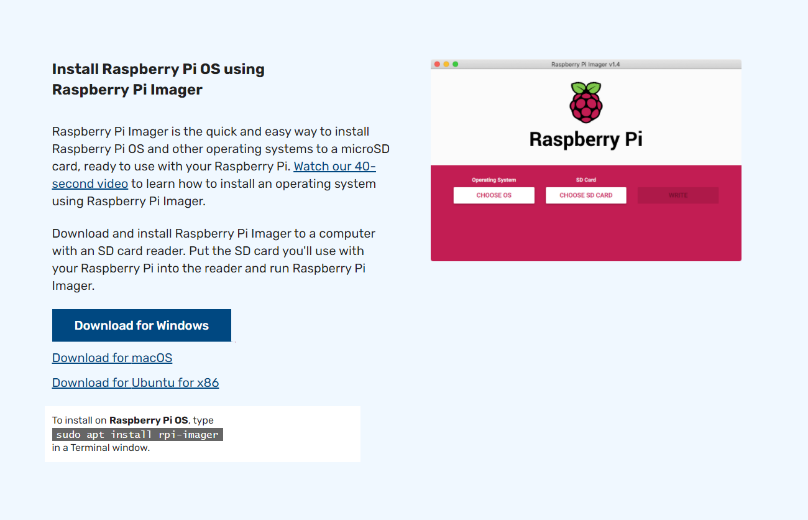
Descargar la imagen para la instalación del SO

Figura 1.34 Imagen Rasberrypi.org/Software

Elegir el SO

Figura 1.35 Programa para Instalación del SO

Elegir el dispositivo donde se instalará

Figura 1.36 Programa para Instalación del SO

Esperar a la instalación



Figura 1.37 Programa para Instalación del SO

Esperar a que el programa instale una imagen del SO, una vez terminada la instalación solo falta quitar el SD del pc y colocarlo en nuestro rasberry (Se recomienda hacer la configuración del wifi antes).

* + - 1. **Configuración del SO**

Antes de encender el Rasberry hay que hacer una configuración del wifi en la tarjeta que se encuentra dentro del directorio boot en la tarjeta de memoria.

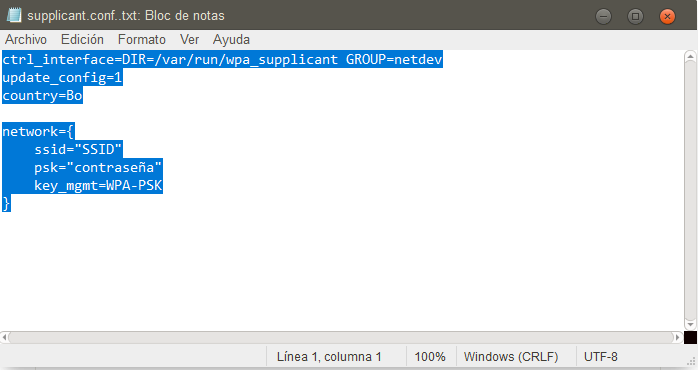
Se tiene que crear un nuevo archivo llamado wpa.supplicant.conf

Figura 1.38 Archivo de configuración del wifi

Si el usuario es de otro país donde dice country se tiene que cambiar por la clave del país correspondiente, donde dice ssid por el id del wifi, donde dice psk la contraseña del wifi, recordando que estas comillas dobles no son necesarias escribirlas, solo son para denotar el ejemplo.

Las entradas del Rasberry se conectan como indica en la sección “Conexión entre los diferentes hardware” específicamente figura 1.28.

Procedemos a conectar la entrada de video, alimentación, mouse, teclado y encendemos, encontraremos que nos pide una configuración inicial, simplemente colocamos lo datos que nos pide, en caso del proyecto usaremos Ciudad como Bolivia, Lenguaje como español y la zona horaria, LaPaz, para la contraseña es disposición del usuario y el usuario por defecto rabian lo nombra como “pi”.

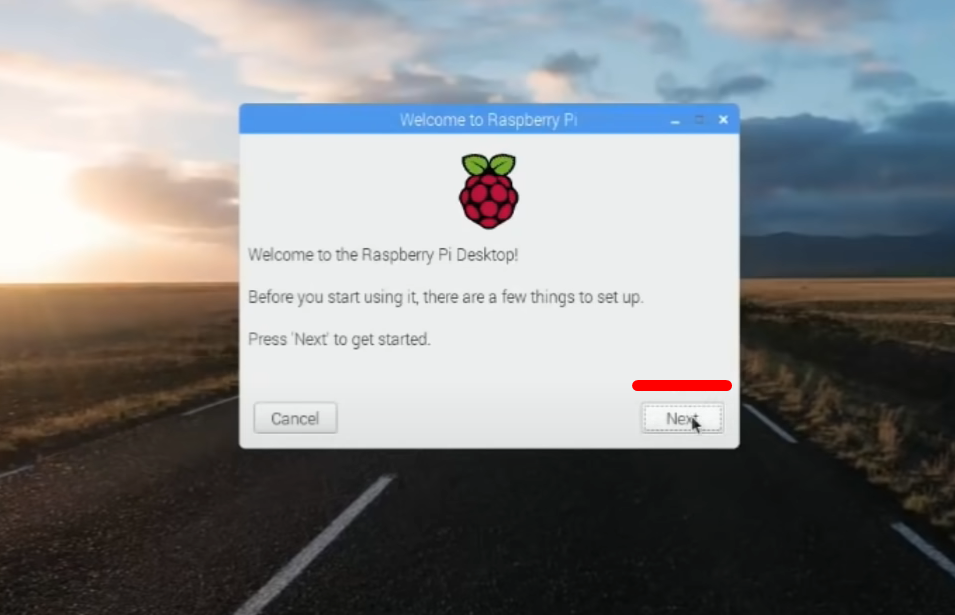
Figura 1.39 Configuración inicial Rasbian



Figura 1.40 Configuración inicial 2 Rasbian



Figura 1.41 Configuración inicial 3 Rasbian

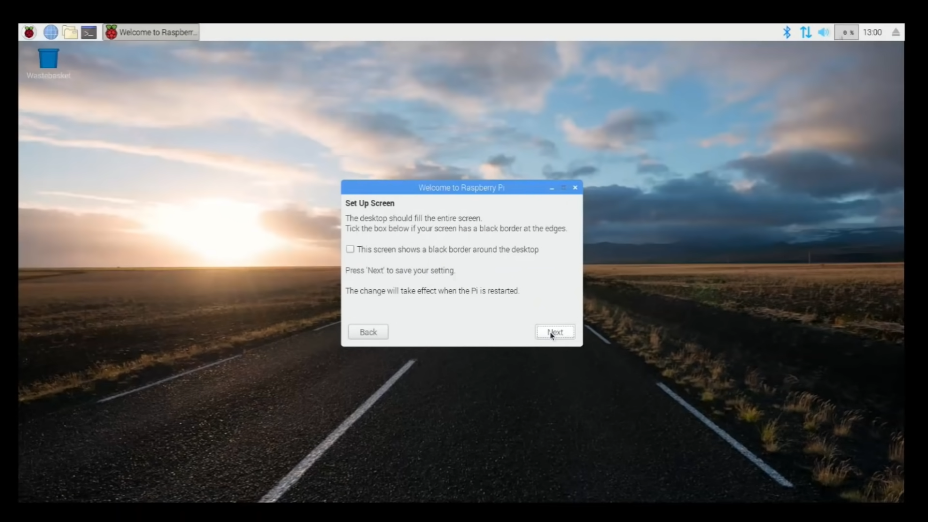


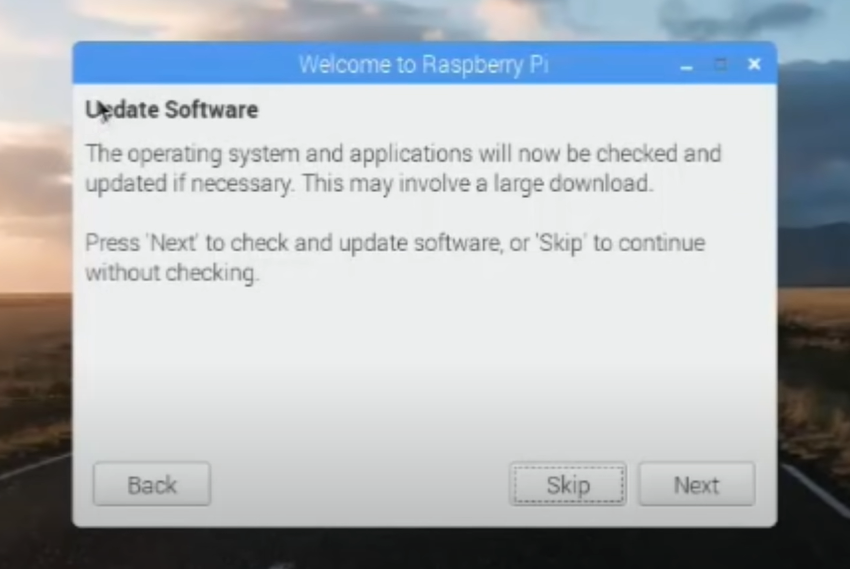
Figura 1.42 Configuración inicial Rasbian

Por último, si la pantalla contiene bordes negros pinchar el check para que al reiniciar este tome la pantalla completa, luego el sistema muestra una pantalla de “Update” que corresponde a la actualización del software, en caso que no lo mostrara igual se puede actualizar el software mediante los códigos:

*Sudo apt-get update*

*Sudo apt-get upgrade*

Como cualquier otro SO Linux llamamos al súper usuario para que configure o actualice el sistema.



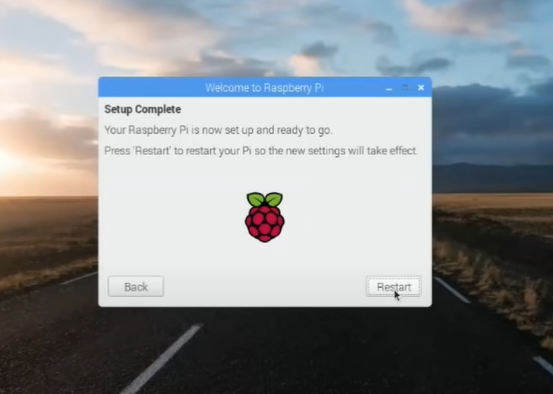
Figura 1.43 Configuración inicial Update

Figura 1.44 Pantalla de configuración completa.

Una vez completada la configuración el procedemos a reiniciar el sistema y cuando vuelva a encender ya no pedirá más configuraciones.

Para podes manejarlo de forma remota tenemos que activar el VNC, para ello instalamos el VNC en los dispositivos que no lo tengas por defecto, abrimos la consola e introducimos los siguientes comandos en la terminal:

*sudo apt-get update*

*sudo apt-get install realvnc-vnc-server*

*sudo apt-get install realvnc-vnc-viewer*

Luego rasbian actualizará tanto la aplicación del servidor como del cliente, una vez actualizadas estas dependencias, insertamos el

comando:

*Sudo raspi-config*

Con esto se iniciará una terminal de configuración del rasberry, entramos a la configuración de interfaces, nos desplazamos hasta donde dice VNC, y aceptamos la solicitud de activar VNC.

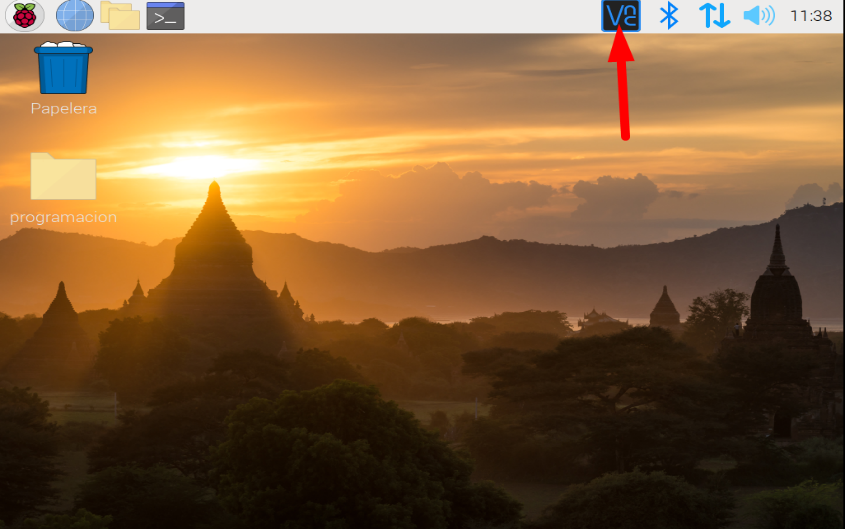
Nos aparecerá una pestaña con el logo de VNC.

Figura 1.45 VNC activado

Podremos acceder a nuestro rasberry de manera local con tan solo el ip del mismo, para ello necesitaremos instalar VNC en nuestro ordenador donde quisiéramos controlar nuestro mini-servidor, Para utilizarlo de manera WAN tendremos que asociar las direcciones del equipo a una dirección DNS y así poder acceder como un servidor WAN, o también memorizar cada una de las direcciones y máscaras para saber la dirección exacta del equipo.

* + - 1. **Instalación de dependencias**

Dentro de los programas que vamos a utilizar para la conexión entre arduino y rasberry necesitaremos un lenguaje para ejecutar un código que comunique ambos dispositivos y como ya se ha mensionado antes utilizaremos Python por lo que procedemos a instalar y actualizar las dependencias.

* sudo apt-get intall Python-serial

Para la utilización del proyecto es necesario contar con muchos programas, entre ellas el mismo arduino IDE, para ello necesitamos como costumbre de cualquier programador de Linux, actualizar el SO.

* sudo apt-get update
* sudo apt-get upgrade

Luego procedemos a instalar arduino en el rasberry.

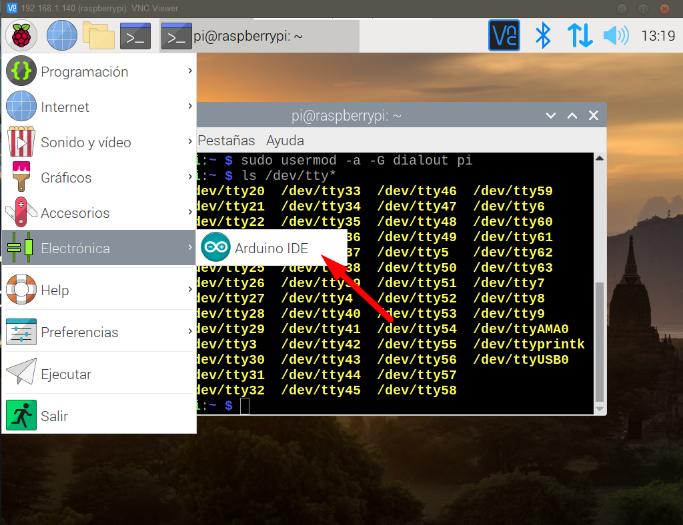
* sudo apt-get install arduino

Figura 1.46 Instalación Arduino IDE

Para verificar los puertos usb-serial y así determinar si se reconoció la tarjeta arduino.

* Dmesg | grep ttyACM

Entonces la consola nos da una respuesta USB AMC device anunciando que nuestro dispositivo está conectado.

Verificamos si reconoce el puerto serie del arduino y conectamos, podemos subir un archivo de prueba para haciendo que este pueda parpadear un led, con el ejemplo del archivo blink dentro del IDE.

Entonces el pin 13 del arduino tendría que parpadear

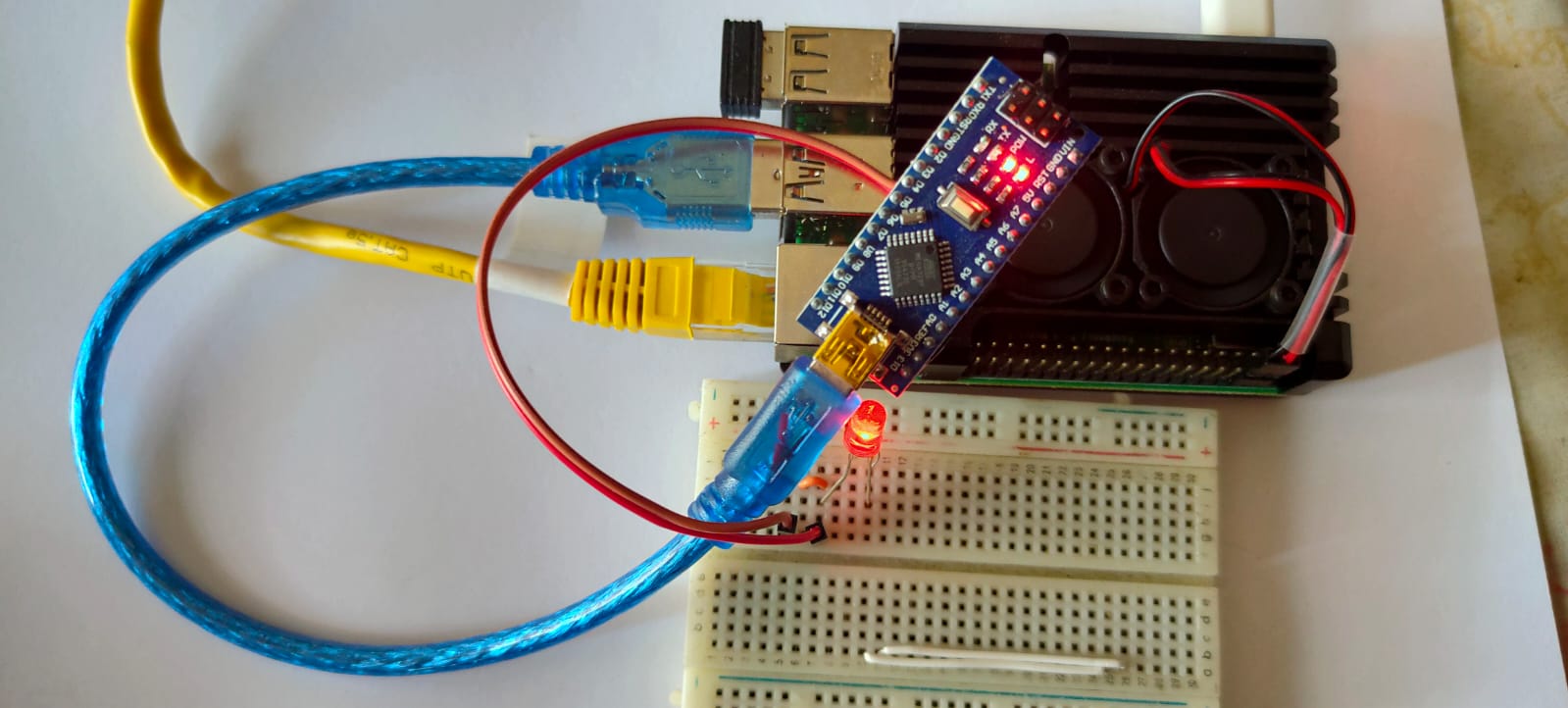


Figura 1.47 Control del arduino

Si somos un poco meticulosos nos damos cuenta que hemos controlado un arduino a través de un mini ordenador que controlamos de forma remota con otro ordenador a través de protocolo ssh, pero aún falta programar bastante para darle el cuerpo al proyecto.

El arduino puede controlarse de forma inalámbrica a través de bluetooth, etc. Pero por falta de tiempo no se implementará esa actualización, aún estará pendiente para versiones posteriores.

Para una versión futura se pretende implementar un sittema de inteligencia artificial usando una herramienta llamada tensor-Flow y robótica utilizando herramientas web como jonny Five y css, JavaScript, Html, nodeJs, etc.

Para estas versiones actualizaremos la versión de npm, en pocas palabras desinstalamos la versión por defecto de nodeJS e instalamos una moderna, para ello nos vamos a los repositorios de git de npm y realizamos la instalación por consola.

* curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash

Luego reseteamos el archivo source para que reconozca el archivo nvm.

* Source ~/.bashrc

Con esto reseteamos el archivo rc y con eso ya podemos instalar nodeJs y ejecutar comandos de node con npm, ejecutamos:

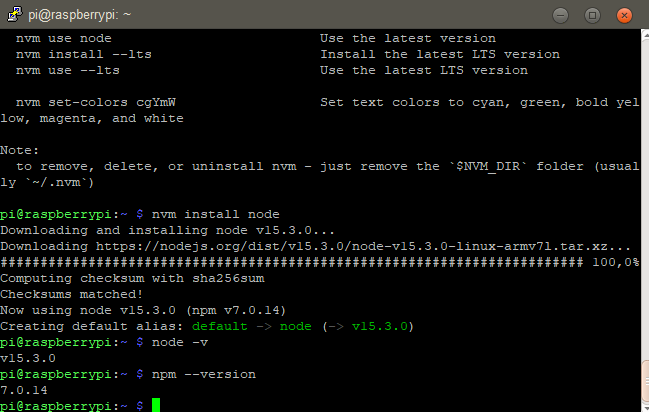


figura 1.48 instalando nodeJs

* nvm instal node

Con los comandos:

* Node –v
* npm –-version

Podemos verificar la versión instalada, si nos da tal respuesta como la versión entonces ya tenemos instalado las dependencias, para guardar nuestro avance para evitar fallos o para actualizar el software en un futuro necesitamos de un controlador de versiones, entonces utilizaremos git, lo instalamos con el siguiente comando.

* sudo apt-get install git

una vez tenemos el controlador de versiones podemos descargar repositorios a nuestra disposición.

* + 1. **Configuración del Servidor**
       1. **Crear Api Rest**

Api significa Interfaz de programación de Aplicaciones y esta es un conjunto de protocolos y definiciones (Normalización**)**.

Las apis permiten que sus protocolos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados, básicamente es **protocolizar el enrutamiento de las aplicaciones.**

Servicio o programa que está en el back-end que nos permite recibir peticiones HTTP ya sea por GET, POST, PUT O DELETE, por los diferentes métodos que acepta el protocolo HTTP e interactuar con una base de datos y devolvernos un resultado en formato JSON o SQL.

**Una forma de crear Apis con Nodejs es:**

1. Comando **npm init** para crear el package .Json
   1. Nombre del Paquete "api-<web>"
   2. que version "(1.0.0)"
   3. description "<Description del api>"
   4. punto de inicio del programa (la HOME) "index.js"
   5. text command " "
   6. git repository " "
   7. keyword " "
   8. autor "<Nombre>"
   9. licencia "<MIT>"

Para saber más sobre el tema (https://docs.npmjs.com/files/package.json)

1. **Instalar las librerías que se necesiten**
   1. **npm i** <> --save --> guardar el paquete en el **directorio actual** [ de otra forma se guarda de manera global en directorios generales ]
      1. express ( frameworks para trabajar con el protocolo http )
      2. body-parser ( paquete para convertir las peticiones a Json )
      3. connect-multiparty ( subir archivos al servidor .file)
      4. mongoose ( Libreria ORM para trabajar con Mongodb )
      5. nodemon -D ( refresca automaticamente los cambies en servidor local ) **[ importante!! NO INSTALAR EN EL SERVIDOR NO LOCAL]**
      6. johnny-five controlador de placas como arduíno
   2. **npm install**
      1. express
      2. mongoose
2. Crear el archivo index.js y **configurar el package.json =>  "scripts" : { ...//, ..//, "start" : "nodemon index.js" }** **( SOLO SERVIDOR LOCAL )**

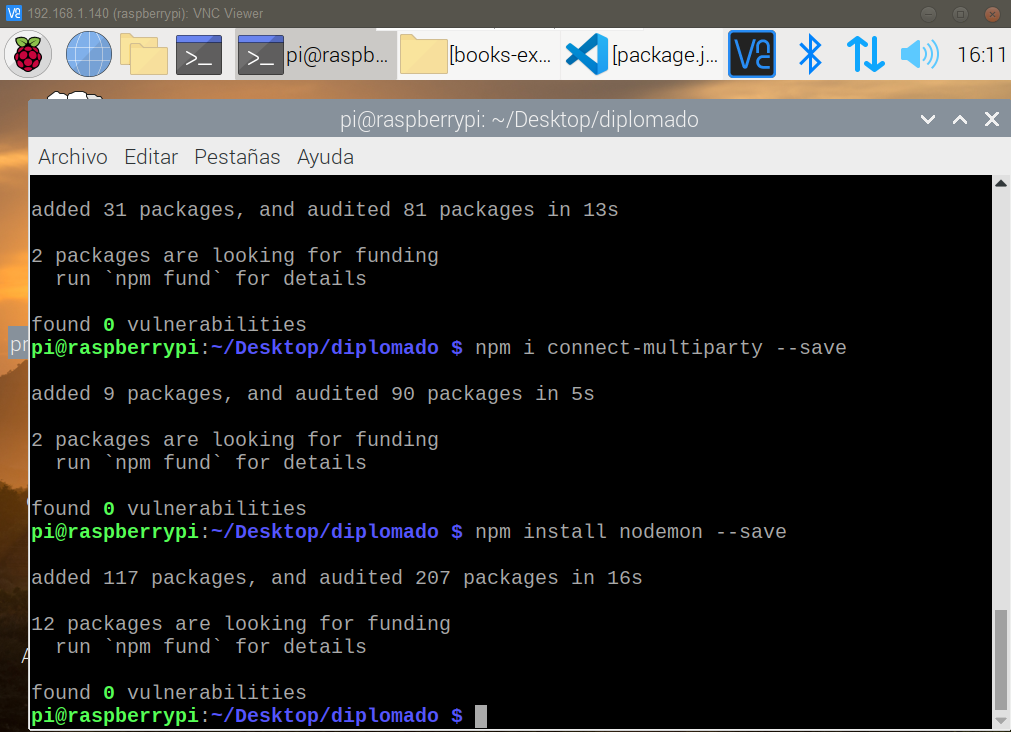


Figura 1.49 Creación de API REST

Entonces obtenemos una carpeta con un archivo package,json, que básicamente es un objeto con diferentes configuraciones, se puede abrir prácticamente con cualquier editor de código, los SO de distribuciones Linux llevan básicamente 2, nano y vim, en caso de no tenerlo con el comando.

* sudo apt-get install <nombre del programa>

Se puede instalar dependencias básicas para las necesidades, con otros programas como docker o git también podemos instalarnos programas de acuerdo a las necesidades, ahora que tenemos instalados estos programas podemos instalar casi cualquier programa, en todo caso si no está muy familiarizado con los códigos también se puede realizar con un asistente de instalación tal y como lo hace Windows, solo que como rasbian es una distro de Linux Debian, intentaremos descargarnos los archivos de instalación de esta distribución.

Para simplificar el uso de editores se utilizará uno de los más populares como es Visual Studio Code.

En el ejemplo anterior el objeto Json ser vería de la siguiente manera

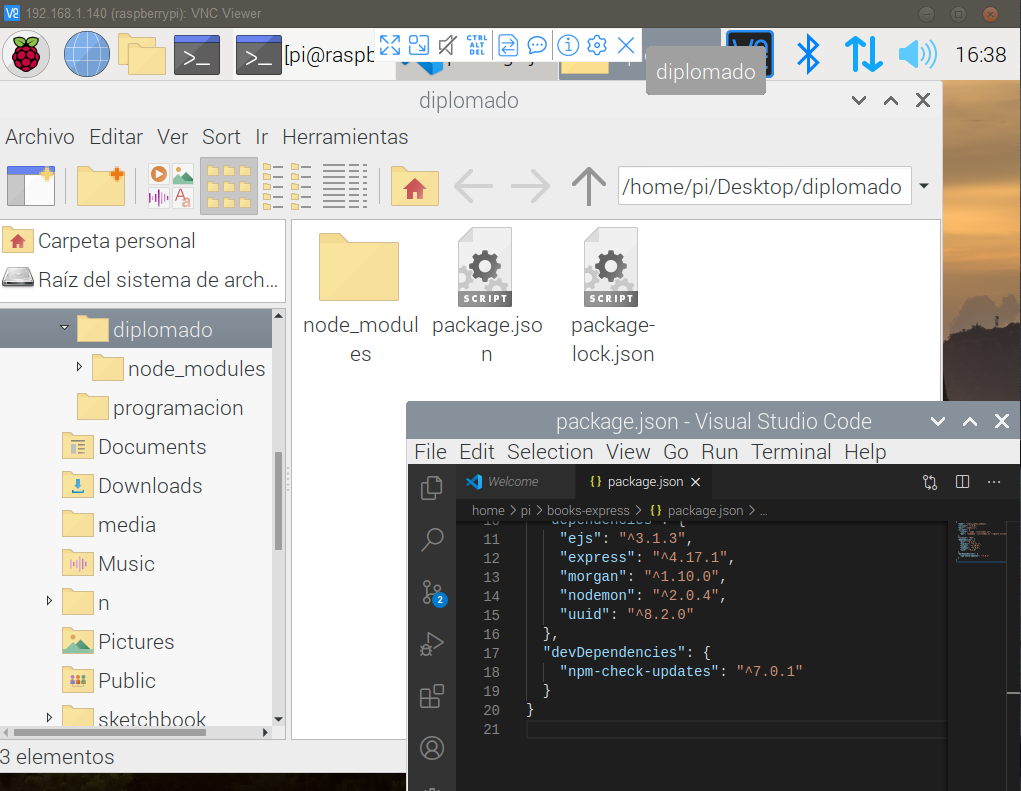


Figura 1.50 Visual Studio Packcage.Json

* + - 1. **Instalación de Base de datos**

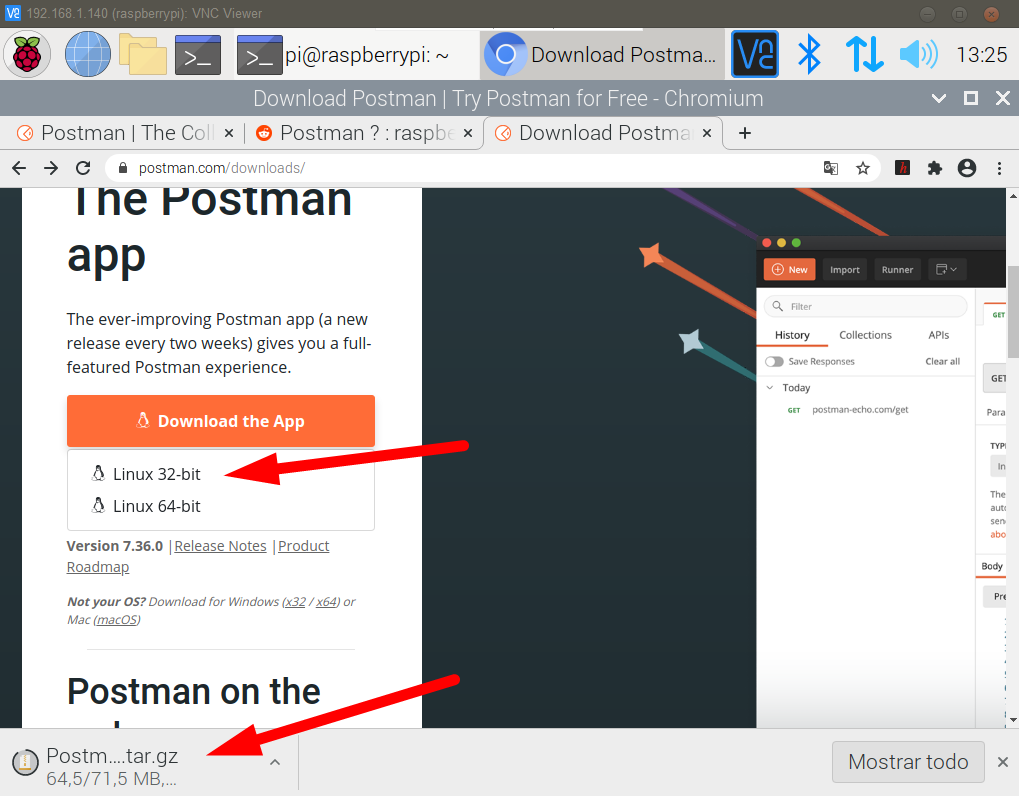
Para la creación de base de datos tenemos muchas opciones, entre ellas destacan dos tipos, las bases de datos relacionales y no relacionales, en base a experiencia personal se utilizará las bases de datos no relacionales, ya es estas suelen ser más rápidas y acumular más información en menor cantidad de memoria, por lo cual una base de datos para este tipo de trabajo donde se pretende almacenar mucha información suele ser más óptimo, las bases de datos relacionales en base a SQL también se pueden utilizar, resulta más fácil e intuitivo aunque son más óptimas para instituciones, escuelas, las no relacionales en base a los objetos Json, Bson, etc. Se pueden utilizar directamente desde JavaScript y son más complejas, para ello necesitaremos hacer las pruebas e instalar las dependencias necesarias para que la misma base funcione, existe también bases de datos en base a kobol, un lenguaje de bajo nivel con el que las bases de datos resultan muchas más óptima en cuestión a velocidad de búsqueda de datos y almacenamiento de grandes volúmenes de información, son un poco más complejas y se utilizan en bancos.

Para la instalación de las dependencias necesitaremos instalar MongoDb ya que puede almacenar datos de forma Bson, para ello utilizaremos el clásico:

* sudo apt-get install mongodb
* mongo –eval

con estos comandos nos instala mongodb y evaluamos si ya está funcionando en el sistema.

1. **Desarrollo final del proyecto**

En esta etapa veremos paso a paso la programación para poder realizar nuestro propio sistema de monitoreo continuo.

.

.

.

.

.

.

.

.

.

. es necesario la motivación. no se me ocurren muchas ideas

* 1. Cálculos
     1. Cálculos de Hardware
     2. Cálculos de Programación

