



TRABAJO FINAL DE CURSO (TFC)

Por Javier Camarero López

2º Desarrollo de Aplicaciones Multiplataforma (DAM) 2024/2025
NDT New Digital Talent
Tutor: Juan Diego Mariscal

Contenido

| | |
|--|----|
| Resumen | 3 |
| Introducción | 3 |
| Estado del arte | 4 |
| Servidor web | 4 |
| Air Quality Monitor (medidor de la calidad del aire) | 4 |
| Sistema de riego para tus plantas de interior..... | 4 |
| Iluminación de ambiente para el televisor..... | 5 |
| Sistema de seguridad Raspberry Pi | 5 |
| Superficie LED interactiva..... | 5 |
| Mundo de Minecraft programable..... | 5 |
| Servidor VPN | 5 |
| Reloj binario | 6 |
| Ted, la tostadora que habla | 6 |
| AirPi: aplicación para el tiempo y estación de medición del aire | 6 |
| Repetidor de wifi..... | 6 |
| Marco de fotos digital | 6 |
| Advice Machine: máquina expendedora de consejos..... | 6 |
| Servidor doméstico y centro multimedia..... | 7 |
| Impresora 3D..... | 7 |
| Máquinas recreativas o videoconsolas | 7 |
| MagicMirror (espejo inteligente) | 7 |
| Zelda Home Automation | 7 |
| Metodología | 8 |
| Raspberry Pi | 8 |
| Sensor DHT11 | 10 |
| Socket..... | 12 |
| HTML | 13 |
| JavaScript | 14 |
| Python | 14 |
| Flask | 15 |
| Express y Node | 16 |
| Resultados | 17 |
| Simulación de Raspberry Pi en caso de no disponer de ella | 17 |
| Experimento físico | 18 |
| Discusión..... | 21 |

| | |
|--------------------|----|
| Conclusiones | 21 |
| Bibliografía | 22 |
| Anexos..... | 23 |

Resumen

Este trabajo está enfocado a la creación de un sistema de domotización de una vivienda mediante el uso de un servidor web que recopila la información de los sensores conectados a una Raspberry Pi 3 Modelo B+. La información captada por los sensores es enviada al servidor mediante la herramienta socket.io que nos permite tener la página web actualizada con cualquier cambio que le llegue a esta sin preocupación de saturar la red. Cabe recalcar que este servidor está levantado de forma local, es decir, está levantado en la red individual y se accede a este solo si se está conectado desde otro dispositivo a la misma red. Esto nos permitirá saber en todo momento los valores de la vivienda de los sensores de temperatura, presión, etc, que tenemos conectados. Sin más demora, procedo a explicar con más detenimiento el funcionamiento de este trabajo de investigación y sus respectivas tecnologías utilizadas en el siguiente punto.

Introducción

La idea de este trabajo surgió en el contexto de usar tanto tecnología usada en clase, en este caso Sockets, y aplicarlo en un ámbito nuevo desde cero usando una tecnología que no tiene nada que ver con lo visto en clase como es la Raspberry Pi. Uniendo ambas cosas surge un resultado capaz de levantar un servidor web que se actualiza constantemente para obtener los valores más actuales de los sensores conectados a dicha placa. El objetivo principal de este trabajo es probar la capacidad de los estudios aprendidos en el grado para ser integrados con tecnología nueva y demostrar la importancia de la actualización de los servidores para darle la mejor información al cliente. Aunque ya existen dispositivos para domotizar casas conectados a un servidor general, este trabajo comprobará que no es necesario usar dichos dispositivos si se tienen los conocimientos necesarios, incluso podría ser aplicable para llevar el seguimiento de los diferentes niveles del agua para un acuario y automatizar el vertido de los productos necesarios o el control de la temperatura del agua. Para llevar a cabo este experimento haré uso de una Raspberry Pi 3 modelo B+ junto con la instalación de su Sistema Operativo propio de Raspberry y conectada por HDMI a un monitor para visualizar en todo momento el servidor web. A ella irán conectados una serie de sensores que serán los encargados de tomar todos los valores necesarios para llevar un seguimiento correcto de lo que queramos saber. Crearé un servidor web que se programará con Python y se actualizará constantemente mediante Sockets, tecnología que permite actualizar los datos del servidor sin necesidad de estar subiéndolos manualmente. Este servidor web tendrá una página web sencilla con un estilo sencillo para embellecer mínimamente la página en la que aparecerán los datos. Por último, si queda tiempo tengo pensado crearle una carcasa a todo el conjunto de componentes a modo de casa en caso de usarse para medir los valores de temperatura, luz, presión, etc., o con forma de acuario o pecera en caso de querer medir los valores del agua.

Estado del arte

Servidor web

La Raspberry Pi puede configurarse en tan solo unos pocos pasos de tal manera que se puede definir como tu propio servidor web Raspberry pi. Aun cuando su rendimiento es inferior al de las variantes comerciales que se ofrecen en el mercado, usar Raspberry Pi como web server privado es muy útil en algunos ámbitos, como cuando quieres analizar y alojar tu página web en un entorno de prueba o gestionar tus archivos por cuenta propia y hacer que estos estén disponibles en la nube.

Si se quiere utilizar Raspberry Pi como un web server, se debe ser consciente de que el hardware tiene ciertas limitaciones. El ordenador ofrece peor rendimiento en comparación con el que puedas obtener en un proveedor web al uso. Para el alojamiento de páginas web complejas (por ejemplo, tiendas online o páginas con contenidos dinámicos), Raspberry Pi también presenta ciertas limitaciones, al igual que para las páginas web que tienen muchos visitantes. El escaso ancho de banda de la conexión a Internet supone otro de los obstáculos en el uso de Raspberry Pi como servidor web.

Sin embargo, el Raspberry Pi web server sí es apto en algunos campos de aplicación: este se puede emplear como servidor local con conexión a Internet o utilizarse en una red corporativa. Con él se pueden testear páginas web que estén online o alojar páginas web completas de pequeña envergadura que tengan un número reducido de visitas. Asimismo, también se puede gestionar una nube propia o configurar programas para la domótica (control de luces, calefacción, etc.).

Otra de las ventajas de un Raspberry Pi como web server es la disminución de gastos corrientes que conlleva, ya que este necesita tanto una conexión a Internet como a la electricidad, pero esto no sobrepasa, en general, los 5 W. De este modo, el gasto por un funcionamiento continuo del servidor web será, prácticamente, insignificante. Además, los gastos de adquisición de los componentes requeridos son mínimos.

Air Quality Monitor (medidor de la calidad del aire)

La calidad del aire, sobre todo en las grandes ciudades, se puede mejorar. Medirla, por ejemplo, teniendo en cuenta la humedad o el nivel de concentración de partículas, es posible con un medidor de la calidad del aire. Se puede crear un medidor de interior y exterior propio con Raspberry Pi y un sensor de partículas SPS30. Gracias al lenguaje Rust los datos son leídos y procesados se suben a Microsoft Azure Cloud, donde se van creando valiosos análisis gráficos en tiempo real con Azure Stream Analytics.

Sistema de riego para tus plantas de interior

Se puede usar la Raspberry Pi como sistema de riego para plantas de interior, ya que, aunque algunas especies soportan bien un riego más irregular, hay otras, como los bonsáis, que resultan muy exigentes. Por ello, existen tutoriales enteros para construir sistemas de riego para bonsáis con Raspberry Pi, que aseguran automáticamente que las plantas reciban la cantidad de agua necesaria en el momento necesario, para ello se utiliza un sensor de humedad en conexión con una bomba.

Iluminación de ambiente para el televisor

La retroiluminación LED en los televisores es cada vez más popular. Sin embargo, los dispositivos que vienen de fábrica suelen estar en un rango de precio alto. Con un poco de habilidad y una Raspberry Pi, no es difícil construir una iluminación ambiental propia. Incluso puede adaptarse en color a las imágenes que se muestran en el televisor, lo que la hace aún más potente que la iluminación LED convencional. Se necesita una bombilla RGB de color regulable, una Raspberry Pi y una cámara web USB. Este último escanea los colores de la imagen mostrada.

Sistema de seguridad Raspberry Pi

Un Raspberry Pi no solo puede aportar confort, sino también incrementar la seguridad doméstica. Este miniordenador puede usarse como base para un modesto pero refinado sistema de seguridad que, una vez encendido, escanea el entorno en directo y notifica cualquier movimiento registrado, incluyendo una foto que envía por mensaje en Telegram. El sistema también puede enviar un mensaje si el aparato se conecta o desconecta.

Superficie LED interactiva

Raspberry Pi no solo es interesante para proyectos con una utilidad práctica. Por alrededor de 130 euros, se puede ensamblar una mesa estándar de IKEA con una pantalla interactiva que reacciona a los objetos que se colocan sobre ella con animaciones de colores y que incluso permite jugar al Tetris. El núcleo del proyecto es un Raspberry Pi que procesa todas las interacciones por medio de microcontroladores Arduino y las transforma en animaciones con ayuda del software Glediator.

Mundo de Minecraft programable

Minecraft ha sido durante años uno de los juegos preferidos sobre todo para los jugadores más jóvenes. La gran variedad de opciones que ofrecen los bloques personalizables permite dar rienda suelta a la imaginación. Minecraft: Pi Edition va incluso un paso más allá. Se puede instalar en una Raspberry Pi y ofrece la posibilidad de ajustar el código de Minecraft para poder manejar los objetos en el mundo virtual como desees. Con este objetivo, esta edición de Pi ofrece soporte para diferentes lenguajes de programación, como Python.

Servidor VPN

Con un servidor VPN (Virtual Private Network) se puede cifrar cualquier tráfico de datos en una red, lo que es recomendable sobre todo si se utiliza una red inalámbrica pública. Sin dicho cifrado, los datos sensibles y personales pueden ser interceptados en cualquier momento. Aquí es donde entra en juego el servidor VPN, que se puede gestionar fácilmente por medio de un dispositivo como Raspberry Pi. Como autoridad central de autenticación y de mediación para cada uno de los clientes VPN, el miniordenador disfruta de una gran aceptación tanto en el ámbito privado, como en el comercial.

Reloj binario

Se puede conectar al miniordenador una Unicorn HAT, una placa de expansión con 64 RGB LED que muestra la hora actual en código binario. De arriba hacia abajo, este reloj especial ofrece el año (las últimas dos cifras), el mes, el día, la hora (en formato de 24 horas), los minutos, los segundos e incluso las centésimas de segundo.

Ted, la tostadora que habla

El control por voz es uno de los temas más importantes en la historia más reciente de la tecnología. Esto es lo que pensó el dúo de desarrolladores “8 Bits and a Byte” lanzando al mercado a Ted. Aun siendo un proyecto Raspberry Pi principalmente lúdico, el entretenido invento muestra a la perfección las posibilidades y la flexibilidad del miniordenador. Las funciones locutoras de la tostadora se basan en el Voice Kit de Google AIY y la capacidad de cómputo viene de la mano de un Raspberry Pi 3 Model B con un módulo de cámara incluido.

AirPi: aplicación para el tiempo y estación de medición del aire

AirPi es el kit formado por una Raspberry Pi y diversos sensores que permite medir valores como la temperatura, la humedad del aire, la presión atmosférica, el nivel de radiación ultravioleta (UV) o el contenido de monóxido de carbono o dióxido de nitrógeno en el aire. Además de la información sobre la calidad del aire, esta aplicación para Raspberry Pi también proporciona información sobre la meteorología. Gracias a la capacidad del miniordenador de conectarse a Internet, pueden obtenerse los valores en una interfaz web.

Repetidor de wifi

Las conexiones a Internet sin cable son prácticas, si bien dependen en gran medida de la intensidad de señal que emite el router. Es habitual que la conexión sea mejor, por ejemplo, en el salón que, en el dormitorio, más alejado del router. La solución perfecta a tus problemas es un repetidor wifi. Este recibe la señal de router y la distribuye. En la página PiMyLifeUp, el usuario Gus ha publicado los pasos para que puedas crear este repetidor con un Raspberry Pi. Dependiendo del modelo necesitarás también hasta dos adaptadores de wifi (WiFi dongles), aunque primero comprueba si el modelo que estás utilizando tiene su propio adaptador.

Marco de fotos digital

Un proyecto ideal para los principiantes de Raspberry Pi es el marco de fotos digital. Además de la Raspberry Pi 4, necesitas una pantalla táctil de 7 pulgadas y una tarjeta de memoria para las fotos. El usuario es quien decide qué imágenes se muestran. Podrás filtrar por determinadas palabras clave en motores de búsqueda como Google Images o mostrar, por ejemplo, tus fotos de las vacaciones.

Advice Machine: máquina expendedora de consejos

Otro proyecto muy interesante es la máquina casera expendedora de dichos que, gestionada con ayuda de un Raspberry Pi, ofrece consejos a cambio de una remuneración. La calidad del resultado consiste en una pequeña nota elaborada con una impresora

térmica dependiendo de la cuantía de las monedas utilizadas. Las ingeniosas respuestas en forma de consejos, dichos y chistes no muy serios se obtienen de la “base de datos de la fortuna” que se usa también en sistemas Unix y Linux, donde constituyen una fuente popular de entretenimiento.

Servidor doméstico y centro multimedia

También puede usarse como servidor doméstico de archivos en el que se deposita todo tipo de datos (documentos, imágenes, videos, música, etc.), a los que se puede acceder con cualquier dispositivo conectado a dicho servidor (PC, portátil, smartphone, tablet, etc.). Esto se puede llevar a cabo mediante una conexión por medio de wifi o de cable, pero también se puede ir más lejos y utilizar una Raspberry Pi como un centro multimedia completo. Así, el miniordenador no solo sirve como medio de almacenamiento de contenidos multimedia, sino también como plataforma central para reproducir las películas, la música y las imágenes del disco duro o, asimismo, para usar servicios de streaming y diversas mediatecas online como YouTube o Spotify.

Impresora 3D

Un proyecto con Raspberry Pi muy caro es el Pi 3D Scanner, que consta de 100 miniordenadores Raspberry Pi, cada uno dotado con una tarjeta SD y módulo de cámara propio, que conforman la estructura básica del escáner de cuerpo completo de dos metros de altura. Gracias a un software de gestión de escaneo 3D de programación propia, los valores obtenidos pueden optimizarse y utilizarse posteriormente para imprimir modelos en 3D.

Máquinas recreativas o videoconsolas

El rendimiento del Raspberry Pi es suficiente para reproducir videojuegos antiguos de máquinas recreativas o de antiguas videoconsolas (por emuladores). Los más “manitas” han creado miniaturas de dichas máquinas, pero también versiones de un tamaño similar al original (algunas incluso con ranura para monedas) para obtener la sensación de estar en un auténtico salón recreativo. El software más popular para este tipo de propuestas es una combinación de Raspbian y del emulador basado en él llamado RetroPie.

MagicMirror (espejo inteligente)

El espejo MagicMirror se trata de un espejo unidireccional tras el que se coloca un monitor y el pequeño ordenador. En el espejo se muestran la hora, el tiempo, las próximas citas y mucho más. También existe una segunda versión optimizada del MagicMirror que puede ampliarse casi de forma infinita gracias a su estructura modular. Debido a que el código es abierto, este espejo mágico se ha convertido en los últimos años en un amplio proyecto comunitario.

Zelda Home Automation

Un proyecto con Raspberry Pi muy particular es el Sufficiently Advanced, este consiste en una autoridad domótica central para controlar más cómodamente los dispositivos técnicos en su propia casa que funciona con un Raspberry Pi. El manejo de este sistema no se lleva a cabo por control de voz, comandos de texto o una interfaz web, sino mediante la reproducción de melodías del clásico de Nintendo The Legend of Zelda: Ocarina of

Time. Al igual que Link, el héroe del videojuego que rompió moldes en su época, se utiliza una ocarina (flauta vascular) para tocar, por ejemplo, “Zelda’s lullaby” para abrir la puerta principal del castillo.

Metodología

Raspberry Pi

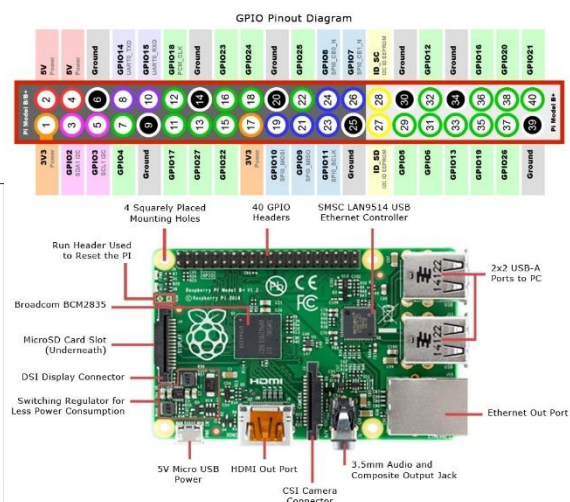
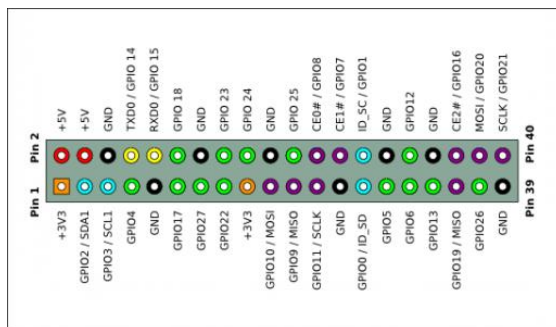
Este sistema informático, de casi el tamaño de una tarjeta de crédito que consiste en un ordenador monoplaca no solo compuesto por el hardware básico de un ordenador (procesador, unidad de memoria, etc.), sino también por diversos tipos de conexiones (USB, HDMI, vídeo, sonido, etc.). Dispone de una ranura para insertar una tarjeta microSD, que hace de disco duro y contiene el sistema operativo. Se recomienda usar el sistema Raspbian, que está basado en Debian, aunque también se pueden usar otras distribuciones de Linux o una versión especial de Windows. La alimentación la proporciona un cargador USB (micro o C) y la conexión a Internet puede establecerse mediante un cable de red a través de una interfaz Ethernet o mediante WiFi. Otros elementos como el ratón, el teclado o el disco duro externo pueden conectarse por medio de las ranuras USB, y el puerto HDMI representa la manera más sencilla de conectar una pantalla a la Raspberry Pi. También se cuenta con diversos pines (pernos de contacto) que pueden programarse con otras funciones.

A lo largo de los años hasta ahora han ido creando diferentes modelos de este miniordenador que son los siguientes:

- Raspberry Pi Modelo A: Este fue el primer modelo de Raspberry Pi que llegó al mercado en 2012 a un precio inicial de unos 40 euros. Era muy básico y de hecho no tenía ni siquiera puerto RJ-45 de red, pero sí que permitía conectarse a Internet si se conectaba a un dongle WiFi por USB. Tenía un único USB-A 2.0, micro USB para alimentación y salida de vídeo HDMI pero también RCA, por lo que permitía utilizarla con monitores y TV muy antiguos. También tenía 26 pines GPIO. Este modelo contaba con un procesador Broadcom BCM2835 de un solo núcleo a 700 MHz, con gráfica Broadcom VideoCore IV y 256 MB de memoria RAM.
- Raspberry Pi Modelo B y B+: Salió al mercado como variante del modelo A también en 2012, e incluía diversas mejoras como por ejemplo el doble de memoria RAM (512 MB), un puerto USB adicional (dos en total) y en este caso sí que se incluyó el puerto RJ-45 para Ethernet cableado. El procesador y demás especificaciones técnicas eran idénticas a la del modelo A, pero un tiempo después se lanzó al mercado la variante B+ que dobló la cantidad de puertos USB (cuatro en total) y pasó de utilizar una tarjeta SD para almacenamiento a una micro SD.
- Raspberry Pi 2 Modelo B: Dos años más tarde (2014) se lanzó la segunda versión definitiva de la Raspberry Pi. El procesador se mejoró a un Broadcom BCM2836 de cuatro núcleos a 900 MHz, mejorando sustancialmente su capacidad de cómputo. También se dobló la cantidad de memoria RAM integrada que pasó a ser de 1 GB, aunque la gráfica seguía siendo la misma VideoCore IV de los modelos anteriores. Este modelo suprime la salida de vídeo mediante RCA, pero a cambio se integraron más pines GPIO para tener 40 en total.

- Raspberry Pi 3 Modelo B: Llegó al mercado en el año 2016 con un procesador mejorado (también un quad core de Broadcom pero a 1.2 GHz) y su gran novedad fue la inclusión de conectividad WiFi y Bluetooth sin necesidad de adaptadores o añadidos. Sin embargo, la memoria RAM se mantuvo en 1 GB y la conectividad no varió en absoluto, así que no contaba con grandes novedades respecto al modelo anterior.
- Raspberry Pi 3 Modelo B+: En marzo de 2018 llegó esta variante que suponía una actualización de la anterior. El procesador volvió a ser mejorado y ahora era de cuatro núcleos a 1.4 GHz, y también se mejoró su conectividad inalámbrica que pasó de 2.4 GHz a 2.4 y 5 GHz. Se mejoró también la versión del Bluetooth integrado (que pasó de 4.1 a 4.2) y el puerto Ethernet pasó a ser Gigabit, aunque limitado a 300 Mbps.
- Raspberry Pi 3 Modelo A+: Llegó a las tiendas en noviembre de 2018, y realmente fueron una decisión comercial de la compañía porque suponían una reducción de las prestaciones del modelo anterior, pero con un precio bastante inferior. Se redujo su cantidad de memoria RAM a 512 MB, solo tenían un puerto USB y se eliminó el conector RJ-45 para tener Ethernet cableado. Sin embargo, fueron un éxito en ventas.
- Raspberry Pi 4 Modelo B: Anunciada en junio de 2019, fue un éxito total de ventas hasta tal punto que las tiendas se quedaban sin stock en cuanto lo reponían y se produjo incluso cierta especulación, tanto en el mercado de segunda mano como por parte de las tiendas minoristas que las vendían a más del doble de su precio original. Este modelo fue el primero capaz de manejar resolución 4K a 60 Hz (incluso dos pantallas 4K a 30 Hz). Además, se incluyó por primera vez un puerto USB 3.0 y su puerto RJ-45 Gigabit ya no estaba limitado (es decir, ahora sí ofrecía hasta 1000 Mbps), ofreciendo un aumento de prestaciones más que notable. Se comercializó en tres variantes dependiendo de su memoria RAM: 2, 4 y 8 GB.
- Raspberry Pi 400: Vio la luz en noviembre de 2020, y consiste en una placa de desarrollo personalizada basada en la Raspberry Pi 4 especialmente diseñada para incluirla en un teclado basado en el Raspberry Pi Keyboard, uno de los accesorios más vendidos de este dispositivo. La idea de este modelo es ser una solución muy compacta para educación, permitiendo tener un sistema de desarrollo que ocupe poco espacio. Mantiene los puertos GPIO, los cuales son accesibles de manera sencilla, ya que están en el borde superior del teclado.
- Raspberry Pi Pico: La primera versión fue presentada en 2021. Actualmente, existen dos variantes de esta misma placa, aunque las dos utilizan el mismo micro controlador (RP2040). Este ha sido diseñado por la propia compañía y nos ofrece un procesador ARM de doble núcleo a 133 MHz acompañado de apenas 264 KB de memoria RAM y 2 MB de almacenamiento interno.
- Raspberry Pi 5 Modelo B: Este es el último modelo que ha aterrizado en el mercado, lanzado en octubre de 2023. Según el fabricante, ofrece un rendimiento de entre 2 y 3 veces el de su predecesora a nivel global (pero especialmente en cuanto a la GPU), gracias a su procesador BCM2712 de Broadcom que cuenta con 4 núcleos ARM Cortex A76 a 2.4 GHz, que integra una GPU VideoCore VII V3D a 800 MHz capaz de decodificar vídeo a 4K y 60 FPS.

- Raspberry Pi Zero: El primer modelo, la Raspberry Pi Zero, fue lanzado en 2015 a un precio de apenas 5 dólares, hecho que propició un éxito abrumador de ventas. La Pi Zero contaba con un procesador de un núcleo trabajando a una frecuencia de 1 GHz. Se complementa este procesador con 512 MB de memoria RAM. Para el almacenamiento integra un lector de tarjetas micro SD. Sobre la conectividad, cuenta con un mini HDMI, un micro USB OTG y un micro USB para alimentación, así como un HAT de 40 pines y un conector CSI para cámaras. Posteriormente, Raspberry Pi lanzó el modelo Zero W, el cual es idéntico al modelo anterior, salvo por un aspecto. Esta tarjeta incluye compatibilidad con Wi-Fi 5, Bluetooth 4.1 y Bluetooth Low Energy. Lo que permite esta mejora en la conectividad es la posibilidad de desarrollar proyectos mucho más potentes.
- Raspberry Pi Zero 2: Esta es una revisión del modelo anterior, dotándolo de mayor potencia y capacidades. Para empezar, la Zero 2 W utiliza un procesador Cortex-A53 de 4 núcleos trabajando a una frecuencia de 1 GHz. Incluye esta placa un total de 512 MB de memoria SDRAM y un lector de tarjetas micro SD, para que podamos personalizar la capacidad de la unidad. Incluye esta placa un mini HDMI para video y un micro USB OTG para la conectividad, así como un micro USB para la alimentación. Por otro lado, incluye un conector CSI-2 para cámara y un HAT de 40 pines. Algo muy interesante es que ofrece decodificación de MPEG-4 y H.264 y codificación H.264, así como soporte para gráficos OpenGL 2.0. Finalmente, esta placa nos ofrece conectividad Wi-Fi 5 en la banda de los 2.4 GHz y conectividad Bluetooth 4.2 y Bluetooth Low Energy.



Sensor DHT11

El sensor DHT11 es un sensor pequeño, económico y sencillo de utilizar. Nos permite obtener información tanto de la temperatura como de la humedad del aire que lo rodea y todo ello utilizando un único pin de nuestro Arduino, Raspberry Pi o microcontrolador favorito.

En este caso, además, viene en un módulo que incluye la pull-up y saca únicamente las 3 patas que necesita el sensor para funcionar (dos para la alimentación y una para el dato) por lo que incluirlo en tu proyecto será tan sencillo como conectar y empezar a utilizar. Además, viene con tres cablecillos tipo jumper para que tengas todo lo necesario para montarlo.

Una de las ventajas que nos ofrece el DHT11, además de medir la temperatura y la humedad, es que es digital. A diferencia de sensores como el LM35, este sensor utiliza un pin digital para enviarnos la información y, por lo tanto, estaremos más protegidos frente al ruido.

Si el ejemplo más común es la estación meteorológica, también puede ser útil en proyectos donde otros sensores y componentes dependan de la temperatura o la humedad para tomar las medidas.

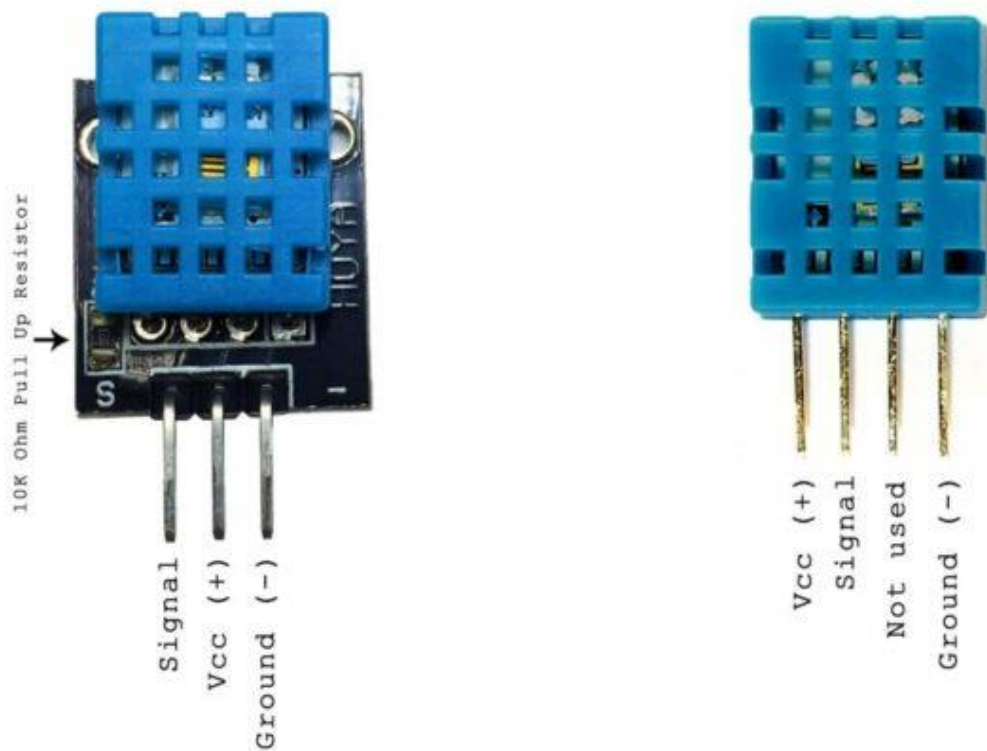
Lo podemos comprar de dos maneras, de forma individual donde solo tenemos el sensor DHT11, o insertado en una PCB.

La diferencia en precio no es excesiva y la versión con PCB aporta una resistencia pull-up de 5 k Ω y un LED que nos avisa de su funcionamiento. Otra diferencia entre estas dos versiones del DHT11 son los pines.

En la versión sin PCB tenemos 4 pines y en la versión con PCB tenemos 3 pines.

No tenemos que confundirnos entre analógico y digital. Aunque lo conectemos a un pin digital, se trata de un dispositivo analógico. Dentro del propio dispositivo se hace la conversión entre analógico y digital.

Por lo tanto, partimos de una señal analógica que luego es convertida en formato digital y se enviará al microcontrolador. La trama de datos es de 40 bits correspondiente a la información de humedad y temperatura del DHT11.



Socket

Un socket, en términos sencillos, es una interfaz de programación de aplicaciones (API) que proporciona un mecanismo para que los procesos de software se comuniquen entre sí a través de una red. Es la puerta de entrada a la transmisión y recepción de datos entre dispositivos, ya sea en una máquina local o a través de internet.

La comunicación a través de sockets sigue un modelo cliente-servidor, donde un extremo actúa como el servidor que espera conexiones entrantes y el otro como el cliente que inicia la conexión. Aquí hay una descripción paso a paso de cómo funcionan los sockets:

1. Creación: El proceso comienza con la creación de un socket en ambas extremidades, el cliente y el servidor. El servidor escucha en un puerto específico, esperando solicitudes entrantes, mientras que el cliente especifica la dirección IP y el puerto al que desea conectarse.
2. Conexión inicial: El cliente inicia una solicitud de conexión al servidor mediante la dirección IP y el puerto. Si el servidor está disponible y acepta la conexión, se establece una conexión entre los dos extremos.
3. Transferencia de datos: Una vez establecida la conexión, los sockets permiten la transferencia de datos en ambas direcciones. La información puede ser enviada y recibida de forma secuencial o simultánea, dependiendo de la implementación.
4. Cierre de la conexión: Cuando la transferencia de datos ha concluido, ambas partes pueden cerrar la conexión. Esto libera los recursos asociados con el socket y permite que otros procesos utilicen el mismo puerto en el futuro.



HTML

HTML es el lenguaje con el que se define el contenido de las páginas web. Corresponde a las siglas en inglés de Lenguaje de Marcado de Hipertexto, básicamente son un conjunto de etiquetas que el navegador interpreta y se emplean para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, tablas, vídeos, etc.

El lenguaje HTML sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces hacia otras páginas o documentos.

Hay que mencionar que el HTML no es un lenguaje de programación, ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de estos lenguajes, por lo que el HTML únicamente sirve para crear páginas web estáticas.

El HTML nace en 1980 cuando el científico Tim Berners-Lee propuso un nuevo sistema de hipertexto para compartir documentos. Surge ante la necesidad de encontrar documentos que te aporten información relacionada con el tema que necesitas y que, además, puedas acceder a ellos desde cualquier parte del mundo.

Es por ello por lo que un grupo de expertos con Berners-Lee de cabecilla deciden crear un nuevo proyecto: el World Wide Web (WWW). Una de las bases de este proyecto es el lenguaje HTML, que como ya hemos mencionado permitirá que cada uno de los documentos que están en internet tengan un formato unificado y que le dé una estructura.

La forma de hacerlo es a través de etiquetas con un orden determinado. Recordemos que este lenguaje sólo podrá ser procesado por un navegador como los que usamos a diario, como el que posiblemente estés empleando ahora para leer este artículo. Los más conocidos son Safari, Chrome, Firefox y Opera, entre otros.

Un elemento HTML se compone de varias partes importantes:

1. Etiqueta de apertura: Es el nombre del elemento (como <p> para un párrafo), encerrado entre corchetes angulares < >.
2. Contenido: El texto o los datos que aparecerán en la página (por ejemplo, «Mi gato es muy gruñón»).
3. Etiqueta de cierre: Indica el final del elemento y se parece a la etiqueta de apertura, pero con una barra / (por ejemplo, </p>).
4. Atributos (opcional): Los atributos proporcionan información adicional sobre el elemento. Se colocan dentro de la etiqueta de apertura, como class o id.



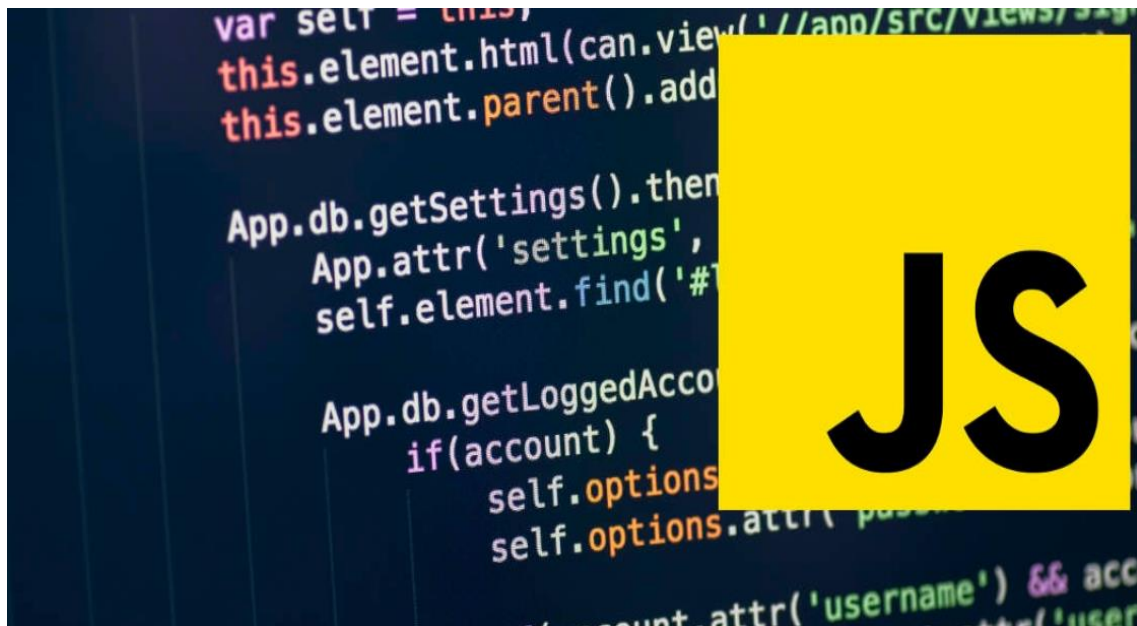
```
95 <div class="container">
96   <h1>One more for good measure.</h1>
97   <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Sed ut elit sed neque elit imperdiet eget.
98   </p>
99   <p><a class="btn btn-lg btn-primary" href="#" role="button">View gallery</a>
100 </p>
101 </div>
102 </div>
103 <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
104   <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
105   <span class="sr-only">Previous</span>
106 </a>
107 <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
108   <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
109   <span class="sr-only">Next</span>
110 </a>
111 </div><!-- /.carousel -->
```


JavaScript

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho más detalle en otras partes del Área de aprendizaje.

El núcleo del lenguaje JavaScript de lado del cliente consta de algunas características de programación comunes que te permiten hacer cosas como:

- Almacenar valores útiles dentro de variables. En el ejemplo anterior, por ejemplo, pedimos que ingreses un nuevo nombre y luego almacenamos ese nombre en una variable llamada name.
- Operaciones sobre fragmentos de texto (conocidas como "cadenas" (strings) en programación). En el ejemplo anterior, tomamos la cadena "Player1:" y la unimos a la variable name para crear la etiqueta de texto completa, p. ej. "Player1: Chris".
- Y ejecuta código en respuesta a ciertos eventos que ocurren en una página web. Usamos un evento click en nuestro ejemplo anterior para detectar cuándo se hace clic en el botón y luego ejecutar el código que actualiza la etiqueta de texto.



Python

Python es un lenguaje de programación informático que se utiliza a menudo para crear sitios web y software, automatizar tareas y realizar análisis de datos. Python es un lenguaje de propósito general, lo que significa que se puede utilizar para crear una variedad de programas diferentes y no está especializado en ningún problema específico. Esta versatilidad, junto con su facilidad para los principiantes, lo ha convertido en uno de los lenguajes de programación más utilizados en la actualidad.

Python se utiliza a menudo para desarrollar el back-end de un sitio web o una aplicación, las partes que el usuario no ve. El papel de Python en el desarrollo web puede incluir el envío de datos hacia y desde los servidores, el procesamiento de datos y la comunicación con las bases de datos, el enrutamiento de URL y la garantía de seguridad. Python ofrece varios marcos de trabajo para el desarrollo web. Los más utilizados son Django y Flask.



Flask

Flask es un framework de desarrollo web de código abierto, escrito en Python. Fue diseñado para ser un marco minimalista, flexible y fácil de usar para la creación de aplicaciones web.

Proporciona una manera sencilla de crear y desplegar aplicaciones web dinámicas; permite que los desarrolladores se centren en la lógica de la aplicación en lugar de que se preocupen por la infraestructura subyacente. Asimismo, ofrece una gran cantidad de libertad y control sobre el desarrollo de la aplicación, lo que lo vuelve ideal para proyectos pequeños y medianos.

Su integración con bibliotecas y tecnologías Python lo hace fácil de integrar con una amplia variedad de soluciones y herramientas de desarrollo de software.

Los beneficios más notables al usarlo son los siguientes:

- Sencillez y flexibilidad. Puedes comenzar con una aplicación pequeña y expandirla con facilidad a medida que crece.
- Personalización. Puedes construir una aplicación que se adapte perfectamente a tus necesidades específicas.
- Documentación completa y comunidad activa. Siempre puedes encontrar ayuda cuando la necesites.
- Integración fácil con otras tecnologías. Es compatible con una amplia gama de tecnologías, incluidas bases de datos, ORMs, sistemas de autenticación y más.
- Desarrollo rápido y eficiente. Puedes llevar tu aplicación al mercado con mayor rapidez.
- Escalabilidad. Puedes adaptar tu aplicación a una mayor cantidad de usuarios y cargas de trabajo sin tener que reescribir el código desde cero.
- Menor tiempo de carga. Es ligero y eficiente en términos de recursos, lo que significa que las páginas web se cargan pronto y sin problemas.



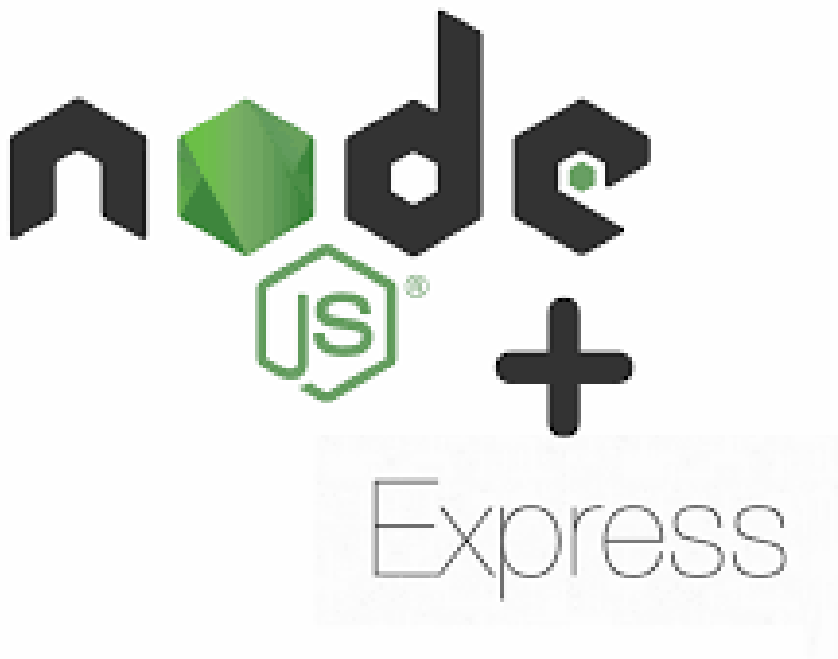
Express y Node

Node (o más correctamente: *Node.js*) es un entorno que trabaja en tiempo de ejecución, de código abierto, multi-plataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo más tradicionales que incluyen HTTP y bibliotecas de sistemas de archivos.

Express es el framework web más popular de *Node*, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para:

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

A pesar de que Express es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. Hay librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos POST, cabeceras de seguridad y muchos más.



Resultados

Simulación de Raspberry Pi en caso de no disponer de ella

En esta prueba he podido comprobar que, aunque se podría intentar imitar el proyecto en una simulación, no es recomendable ya que pierde por completo la posibilidad de usar los pines característicos de entrada y salida de la propia Raspberry Pi física, por lo que no podríamos leer los datos del sensor de temperatura y humedad. Por otro lado, es un entorno ideal para practicar la instalación del software de Raspberry Pi oficial si aún no se dispone de una placa real o de una tarjeta micro SD para realizar la instalación del SO, incluso se podría usar como “máquina virtual” como un ordenador dentro de otro y comprobar que es posible levantar servidores como hemos visto en clase usando Virtual Box emulando una máquina con Linux dentro de un ordenador con Windows. En este caso se haría mediante otro software diferente llamado QEMU ejecutado desde la terminal con los siguientes comandos:

“sudo apt update” y “sudo apt install qemu-system qemu-efi-aarch64”.

Estos comandos actualizan el software del sistema operativo (al usar “sudo” podemos deducir que esto está preparado para usarse en Linux) y se instala el emulador con soporte AArch64, arquitectura de la que está compuesta nuestra Raspberry Pi 3 Modelo B+. A continuación, se descarga el SO correspondiente. Mediante comandos se escribe “wget https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2024-03-15/2024-03-15-raspios-bookworm-arm64.img.xz” y “unxz 2024-03-15-raspios-bookworm-arm64.img.xz”.

También se puede renombrar de la siguiente forma:

“mv 2024-03-15-raspios-bookworm-arm64.img raspios64.img”.

Con nuestro SO preparado procedemos a crear la imagen de disco que se permitirá que el emulador sea capaz de simular el firmware de la Raspberry Pi con UEFI.

“wget <https://releases.linaro.org/components/kernel/uefi-edk2/latest/qemu-aarch64/firmware.tar.gz>” y “tar -xvzf firmware.tar.gz”.

Con estos dos últimos comandos creamos un archivo “QEMU_EFI.fd” que contiene la BIOS del sistema ARM64 y otro archivo con las variables UEFI llamado “vars.fd”.

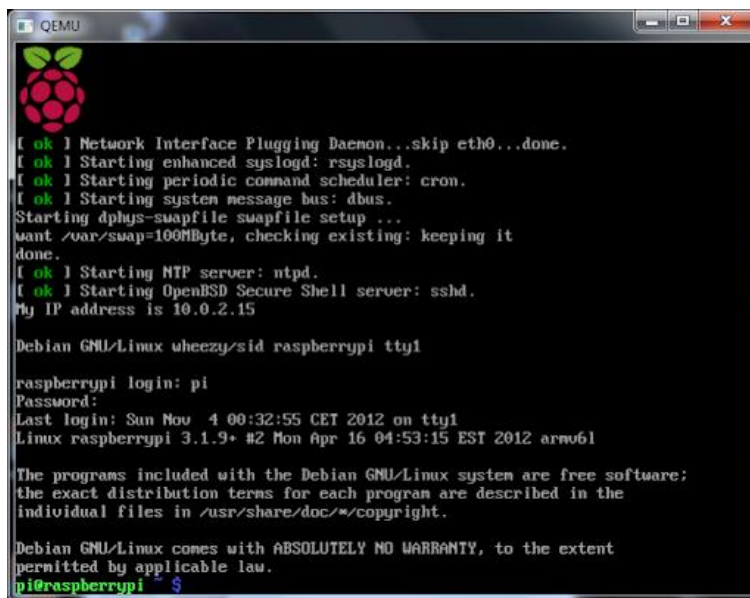
Para finalizar, se lanza el emulador con las siguientes especificaciones:

```
“qemu-system-aarch64 \  
-machine virt \  
-cpu cortex-a53 \  
-m 1024 \  
-bios QEMU_EFI.fd \  
-nographic \  
-drive if=none,file=raspios64.img,format=raw,id=hd0 \  
”
```

```
-device virtio-blk-device,drive=hd0 \
-netdev user,id=net0 -device virtio-net-device,netdev=net0 \
-device virtio-keyboard-device \
-device virtio-mouse-device \
-device virtio-gpu-device”.
```

Con eso ya deberíamos de tener de forma virtual una Raspberry Pi arrancada y lista para usarse. Mediante el usuario “pi” y la contraseña “raspberrypi” que vienen por defecto estará completamente operativa.

Como he mencionado anteriormente, este método no nos deja utilizar los pines de la Raspberry Pi al necesitar una placa física para poder leer el sensor, pero viene bien como método de práctica o enseñanza para entender cómo funciona el software de esta o como toma de contacto para asemejar la Raspberry a un ordenador independiente.



```
QEMU
[ ok ] Network Interface Plugging Daemon...skip eth0...done.
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting system message bus: dbus.
Starting dphys-swapfile swapfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
My IP address is 10.0.2.15

Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Sun Nov  4 00:32:55 CET 2012 on tty1
Linux raspberrypi 3.1.9+ #2 Mon Apr 16 04:53:15 EST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi ~$
```

Experimento físico

Para este proyecto ha sido necesario pedir prestada una Raspberry Pi 3 Modelo B+ proporcionada por el encargado de mis prácticas. El sensor DHT11 lo he proporcionado yo mismo de mi casa ya que en el grado superior que completé con anterioridad utilicé la placa Arduino Uno que usa los mismos sensores que la Raspberry Pi, pero para un alumno que entra de nuevas a este curso es algo nuevo y, por lo tanto, debería de pedirlo prestado al igual que la Raspberry. Cabe destacar también que no me ha resultado un proyecto difícil de hacer debido a la base que poseo a cerca de la programación con placas y de las conexiones a esta, ya que en mantenimiento electrónico te enseñan esa base y considero que es un aspecto de la programación que es curioso y divertido de aprender junto con la programación web o de aplicaciones multiplataforma. Es una forma de programar en Python bastante llamativa que puede no tener la base suficiente como una asignatura completa, pero podría darse como complemento a la hora de enseñar Python como un lenguaje más de programación. Dejando a un lado esto, procedo a describir el proyecto llevado a cabo.

Lo primero que se debe hacer para este proyecto, a parte de comprar una Raspberry Pi que precisamente barata no es, es preparar una tarjeta micro SD para formatearla con el programa que se encarga de crear las imágenes de disco para Raspberry conocido como “Raspberry Pi Imager”. Una vez formateada la tarjeta micro SD se inserta en la ranura correspondiente de la Raspberry y al encenderla comenzará la instalación del SO como con cualquier ordenador, el propio sistema te guiará por las diferentes pantallas hasta terminar la configuración (importante conectar mínimo un teclado por uno de los puertos USB para poder realizar la configuración, el ratón es opcional pero también puede añadirse). Con el SO instalado podemos comprobar que está creado a partir de Debian, una variante de Linux, por lo que la interfaz y los comandos de terminal serán los mismos.

A continuación, comprobamos las actualizaciones más nuevas de la base de datos con “sudo apt upgrade” y las instalamos con “sudo apt update”, todo desde la terminal. Una vez actualizado todo procedemos a descargar Visual Studio Code, que hará de intermediario para modificar los archivos del proyecto.

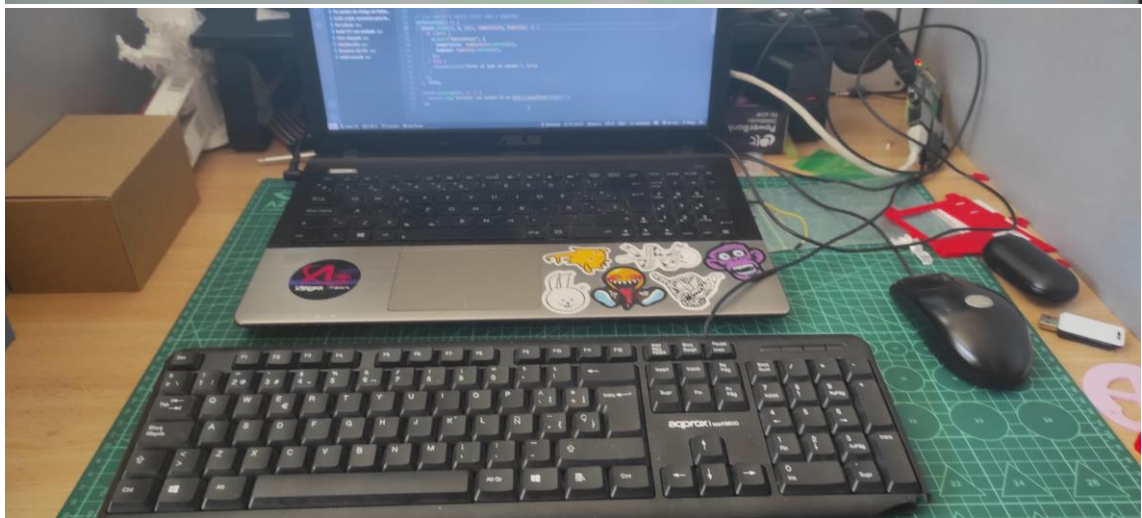
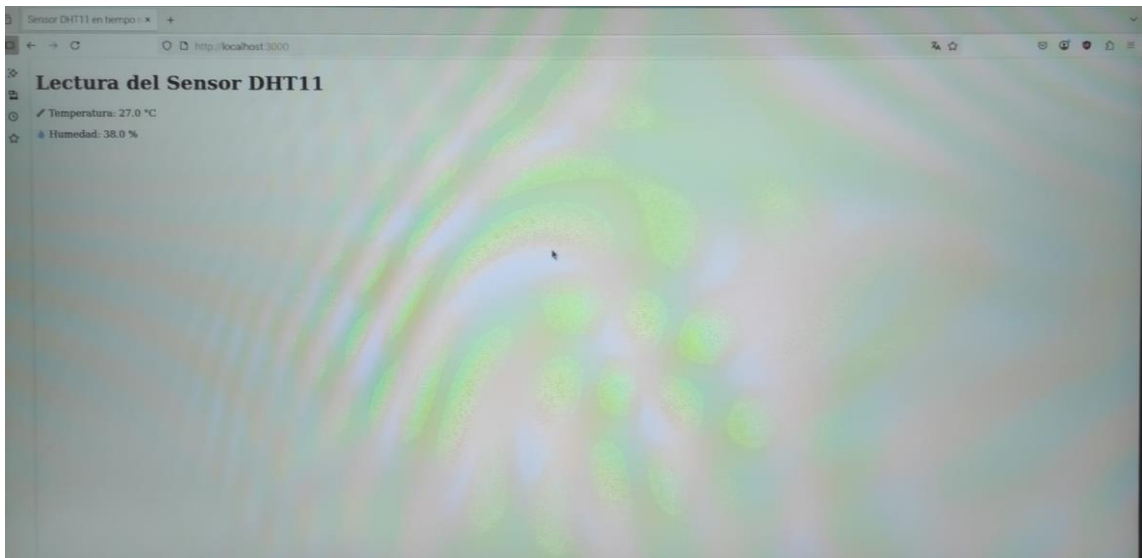
Una vez que se instala todo, procedemos a conectar el sensor DHT11. La patilla de VCC va conectada a 3.3V o a 5V, la patilla GND se conecta al negativo de la Raspberry (también se llama GND) y la patilla de DATA al pin GPIO4 de la Raspberry Pi (uso el pin 4 pero hay más pines en caso de ampliar el proyecto o de querer usar otro pin en particular).

Con todo el hardware conectado, procedemos a instalar las dependencias en Python con el comando: “pip install Adafruit_DHT flask flask-socketio”. Con esto le decimos que instale la librería correspondiente al control del sensor, la de Flask y la parte concreta de Flask que puede comunicarse mediante Socket.io.

Desgraciadamente, estas librerías están sin actualizar desde hace mucho tiempo y da problemas a la hora de instalarlas en nuestra Raspberry, por lo que encontramos una solución más rápida si cambiamos de lenguaje de programación a JavaScript, cuya lógica es la misma que Python, aunque sintácticamente no se parezca demasiado. Personalmente me siento más cómodo usando JavaScript al ser un lenguaje que hemos dado a lo largo de dos años. Cambiamos el archivo del servidor “server.py” por otro nuevo llamado “server.js” y seguimos con la preparación.

Ahora se crean el archivo “index.html” que contendrá la visualización de nuestro servidor con una función asíncrona que obtendrá los datos de nuestro script, y en una carpeta llamada “public” lo moveremos para una mayor organización a la hora de acceder a los datos. Una vez todo se crea y se rellena con nuestro código, se levanta el servidor con “node server.js”, accedemos en nuestro navegador a la dirección “http://localhost:5000” y tendremos nuestra página web con los datos del sensor.

Debido a que no disponía de una tarjeta Micro SD lo suficientemente rápida se hacía muy tedioso comprobar el funcionamiento de cada cosa, ya que se tardaba una mañana en hacer que la Raspberry Pi se actualizase debido a la velocidad de escritura de la tarjeta usada. Afortunadamente se pudo llevar a cabo el experimento, aunque con demasiadas limitaciones, de haber usado una tarjeta con una velocidad de escritura/lectura más rápida se podría haber realizado mucho más rápido y sencillo.



Discusión

Con los resultados obtenidos en este experimento podemos deducir que, con el debido tiempo y dedicación, es posible llevar este proyecto a una escala mayor con diferentes sensores con finalidades diferentes sin problema, solo es necesario buscar las librerías necesarias para cada sensor y se realiza sin problema. También es verdad que, añadiéndole un archivo CSS, es posible darle un estilo a la página estática para verla un poco más bonita, pero en ese caso he decidido hacerlo más simple para ver su funcionamiento y en un futuro con más sensores conectados y más funciones se procuraría buscar la belleza visual.

Como he mencionado anteriormente, es importante usar tanto software como hardware específico para la realización del proyecto. A lo que quiero llegar con esto es que una Raspberry Pi 3 Modelo B+ (la que yo he usado) puede salir perfectamente por entre 40€ y 50€ dependiendo del lugar en el que se compre, más la tarjeta Micro SD de mínimo clase 10 por el tema de la velocidad de escritura/lectura. Más o menos en total saldría por unos 70€ realizar este proyecto incluyendo el sensor DHT11 (el más básico) por lo que es una inversión que no todo el mundo está dispuesto a hacer sin conocer realmente la capacidad de una Raspberry Pi y su potencial. Pero a nivel académico tiene mucho potencial, ya que puede ayudar a asentar unas bases de programación aprendidas en el curso además de aprender material nuevo como puede ser el lenguaje de programación Python, dedicado sobre todo a la automatización, o a la conexión de hardware externo a los pines de la Raspberry y unos principios básicos de electrónica que siempre viene bien aprender en cualquier momento teniendo en cuenta que posteriormente podríamos dedicarnos a trabajar con hardware de ordenadores y aprender cómo pueden funcionar por dentro nos ayuda a diagnosticar con mayor facilidad el fallo en un ordenador. Además de ser una herramienta de aprendizaje muy divertida de usar en proyectos pequeños sin necesidad de tener que usar un equipo de características muy específicas ni caro.

Conclusiones

En conclusión, este proyecto visto desde el punto de vista individual es una inversión bastante notable para que sea únicamente con esta finalidad, debe de hacerse con vista a ser usada posteriormente en muchos más proyectos personales y con el paso del tiempo y habiendo adquirido los conocimientos necesarios se comenzará a obtener beneficios de haber invertido en ello en el pasado. Pero, visto desde un punto de vista más colectivo (para una escuela, por ejemplo), se le puede sacar provecho a la inversión de adquirir varias Raspberrys con la finalidad de enfocarlas en el aprendizaje de su funcionamiento, de adquirir unos principios básicos en la electrónica para entender el funcionamiento interno de los dispositivos utilizados que pueden fallar con el uso prolongado de estos, y además de aprender lenguajes de programación nuevos que no están en la programación que a su vez serviría de refuerzo a la parte de programación del curso.

Bibliografía

Raspberry Pi: ideas y usos para 2023. (2022, July 4). IONOS. Retrieved May 26, 2025, from <https://www.ionos.es/digitalguide/servidores/know-how/un-vistazo-a-proyectos-basados-en-raspberry-pi/>

Los mejores proyectos para novatos con Raspberry Pi. (2025, May 13). Hard Zone. Retrieved May 26, 2025, from <https://hardzone.es/tutoriales/componentes/proyectos-novatos-raspberry-pi/>

Crear un servidor en la Raspberry Pi – Prometec. (n.d.). Prometec. Retrieved May 26, 2025, from <https://www.prometec.net/raspberry-pi-servidor/>

Conexión GPIO de Raspberry Pi 3. (2016, November 20). Electronica y Ciencia. Retrieved May 26, 2025, from <https://www.electronicayciencia.com/2016/11/conexion-gpio-de-raspberry-pi-3.html>

¿Que es Raspberry Pi? (n.d.). Raspberry Pi. Retrieved May 26, 2025, from <https://raspberrypi.cl/que-es-raspberry/>

DHT11 - Sensor de humedad y temperatura. (n.d.). Ultra-lab. Retrieved May 26, 2025, from <https://ultra-lab.net/producto/dht11-sensor-de-humedad-y-temperatura/>

Cómo utilizar el DHT11 para medir la temperatura y humedad con Arduino. (2017, February 2). Programarfácil.com. Retrieved May 26, 2025, from <https://programarfácil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

Mallón, X. (2025, April 14). Sockets: Qué son, cómo funcionan y tipos - Guía 2025. KeepCoding. Retrieved May 27, 2025, from <https://keepcoding.io/blog/que-es-un-socket/>

▷ HTML: Qué es y para qué sirve – VADAVO. (n.d.). VADAVO. Retrieved May 27, 2025, from <https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/>

¿Qué es JavaScript? - Aprende desarrollo web | MDN. (2025, March 31). MDN Web Docs. Retrieved May 27, 2025, from https://developer.mozilla.org/es/docs/Learn_web_development/Core/Scripting/What_is_JavaScript

¿Qué es Python y para qué se usa? Guía para principiantes. (2023, November 29). Coursera. Retrieved May 27, 2025, from <https://www.coursera.org/mx/articles/what-is-python-used-for-a-beginners-guide-to-using-python>

¿Qué es Flask? Beneficios y usos. (n.d.). MyTaskPanel Consulting. <https://www.mytaskpanel.com/que-es-flask-beneficios-y-usos/>

Introducción a Express/Node - Aprende desarrollo web | MDN. (2024, December 20). MDN Web Docs. Retrieved May 27, 2025, from https://developer.mozilla.org/es/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/Introduction

Anexos

```
1  const express = require("express");
2  const http = require("http");
3  const { Server } = require("socket.io");
4  const sensor = require("node-dht-sensor");
5
6  const app = express();
7  const server = http.createServer(app);
8  const io = new Server(server);
9
10 const port = 3000;
11
12 // Servir archivos estáticos (tu index.html debe estar en la carpeta 'public')
13 app.use(express.static("public"));
14
15 // Leer sensor y emitir datos cada 2 segundos
16 setInterval(() => {
17   sensor.read(11, 4, (err, temperature, humidity) => {
18     if (!err) {
19       io.emit("datosSensor", {
20         temperatura: temperature.toFixed(1),
21         humedad: humidity.toFixed(1),
22       });
23     } else {
24       console.error("Error al leer el sensor:", err);
25     }
26   });
27 }, 2000);
28
29 server.listen(port, () => {
30   console.log(`Servidor con Socket.IO en http://localhost:${port}`);
31 });
```

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6    <title>Sensor DHT11 en tiempo real</title>
7  </head>
8  <body>
9    <h1>Lectura del Sensor DHT11</h1>
10   <div class="sensor-data">
11     <p>🌡 Temperatura: <span id="temp">--</span> °C</p>
12     <p>💧 Humedad: <span id="hum">--</span> %</p>
13   </div>
14   <p id="error" class="error"></p>
15
16   <!-- Cliente de Socket.IO -->
17   <script src="/socket.io/socket.io.js"></script>
18   <script>
19     const socket = io();
20
21     socket.on("datosSensor", (data) => {
22       document.getElementById("temp").textContent = data.temperatura;
23       document.getElementById("hum").textContent = data.humedad;
24     });
25
26     socket.on("connect_error", () => {
27       document.getElementById("error").textContent = "Error de conexión con el servidor.";
28     });
29   </script>
30 </body>
31 </html>
```