Temp Detector

Rodrigo Herera

Gabriel Martin Barrera - TUTOR - TFG

Abstract

Temp Detector es un dispositivo ideado para la registración tanto de entrada como de salida del personal y, al mismo tiempo, poder tomar la temperatura corporal al momento de usar el dispositivo.

CAPÍTULO 1. INTRODUCCIÓN

Temp Detector nace para ofrecer ayuda a un problema de salud de la actualidad, conocido como COVID-19 [1].

Ayudándonos con dispositivos que están inmersos en nuestras culturas desde hace mucho tiempo, los conocemos como "lectores de huella dactilar", vamos a modernizarlos, focalizar el coste-efectivo bajo [2] y principalmente en obtener información básica sobre la salud de las personas.

Pudiendo obtener datos sobre la temperatura corporal de las personas, se puede concientizar sobre futuros problemas y, sin ser menos, alertar a la persona sobre su situación actual.

Las motivaciones principales de hacer este proyecto nace del poder generar una prevención hacia posibles problemas de salud futuros. Como también de poder colaborar con el ecosistema de soluciones open source [3], así cualquier organización que esté interesada en obtener los diagramas y código fuente, lo puede tener sin ningún tipo de coste.

El objetivo principal es proveer de un conjunto de herramientas/soluciones accesibles a nivel mundial, con el foco en la salud de las personas.

Las metas de éste proyecto son:

- Aprender y experimentar con dispositivos IoT y todos sus módulos necesarios.
- Incentivar el desarrollo de software libre gratuita, modificable y accesible para cualquier persona.

CAPÍTULO 2. ESTADO DEL ARTE

El uso masivo de la tecnología IoT (*internet of things o internet de las cosas*) denota sus comienzo en 1980 [4] con una gran aceptación por parte de las industrias entre 2008-2009 [5] y con introducciones en la medicina en 2013 [6], a continuación repasamos la historia y la evolución de IoT.

Internet de las cosas

Se entiende por *internet de las cosas* [7] a un sistema dispositivos computarizados interconectados, máquinas tanto mecánicas como digitales que proveen un identificador único (UIDs) y poseen la habilidad de transferir datos a través de una red sin necesidad de la interferencia humano-a-humano o humano-a-máquina [8].

La definición de *internet de las cosas* fue evolucionando debido a las diferentes adaptaciones de sus múltiples usos con las tecnologías, donde encontramos campos de usos como, analitica en tiempo real, aprendizaje de computadoras, automatización, entre otros [9] [10].

Donde tuvo su gran fuerte *IoT fue* en el concepto conocido como "smart home" o "casa inteligente" [9], la cual consiste en darle reglas o rutinas (entre otras) a los objetos hogareños.

El origen del término *internet of things* fue concebido por Kevin Ashton [11] aunque él prefería el término "Internet para cosas".

Define *internet of things* [11] como “Simplemente el punto en el tiempo donde más cosas u objetos estén conectadas a internet que personas”.

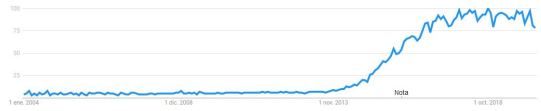


Figura 1. Búsquedas en Internet del término iot.

La imágen fue tomada desde Google Trends [12] nos muestra tendencias de búsquedas, donde 0 es muy poco popular y 100 muy popular. Desde hace varios años *IoT* viene manteniendo una popularidad muy grande.

Herramientas de IoT

Para lo que compete a éste proyecto, vamos a basarnos en una herramienta en particular y sus distintos módulos [13].

Placa Arduino

Las placas Arduino son un conjunto de variedad de microprocesadores y controladores. Viene equipado con sets de entrada y salida tanto digital como analogica. La placa también cuenta con interfaces de comunicación como USB [14].

Los microcontroladores pueden ser programador en lenguajes C y C++ [15].

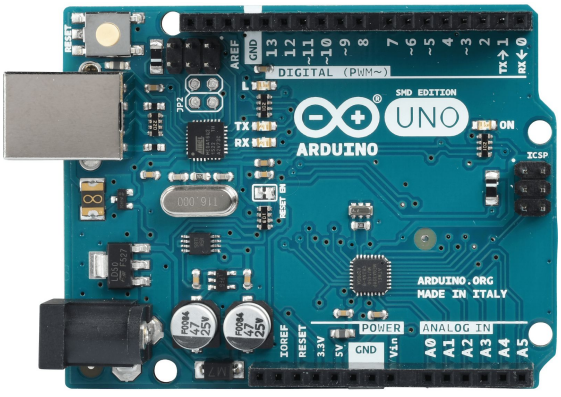


Figura 2. Placa Arduino Uno

Sensores

Los sensores son dispositivos, módulos, máquinas o subsistemas que su propósito es el de detectar eventos o cambios en el ambiente y enviar información a otros componentes electrónicos, normalmente son computadoras. [16]

Figura 3. Sensor de reconocimiento de Co

CAPÍTULO 3. METODOLOGÍA Y TECNOLOGÍA

Este capítulo estará separado en cuatro partes que son las necesidades que se tienen para poder llegar a completar el objetivo de construir el Temp Detector.

A continuación se detallan los programas y plataformas utilizadas en la realización del proyecto y los métodos de trabajo.

Metodologías

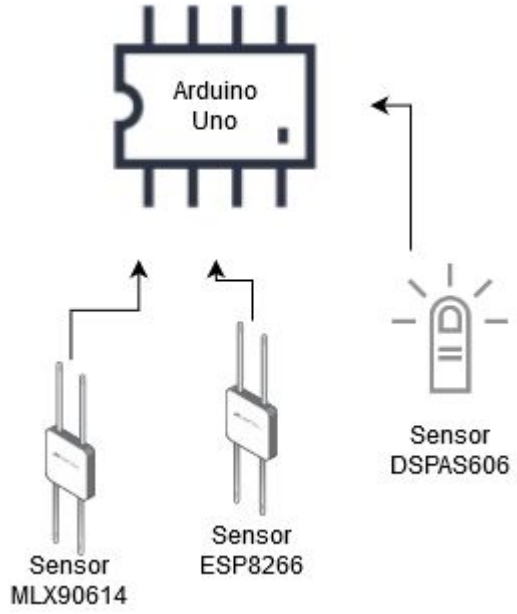
Plan de trabajo IoT

Debido a la poca experiencia con las tecnologías IoT se procede a hacer pequeñas implementaciones individuales de los sensores a utilizar y luego unirlas en una única implementación.

Prototipado rápido

Se elige este tipo de prototipado para poder hacer iteraciones cortas en tiempo así poder ir avanzando con las implementaciones y a medida de incrementar las iteraciones poder mejorar las implementaciones previas. [17]

Diagrama dispositivo IoT

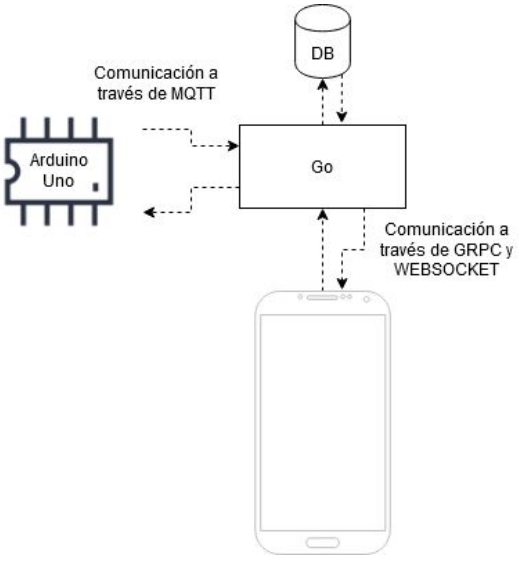


Plan de trabajo Go

Se creará una aplicación en el lenguaje Go [18] el cual tendrá 3 tipos diferentes de funcionalidades:

1. Servidor de MQTT [19] para comunicarse con la aplicación IoT.
2. Servidor GRPC [20] para comunicarse con la aplicación mobile y la aplicación web.
3. Servidor WEBSOCKET[21] para comunicación en tiempo real con la aplicación mobile.

Diagrama backend Go



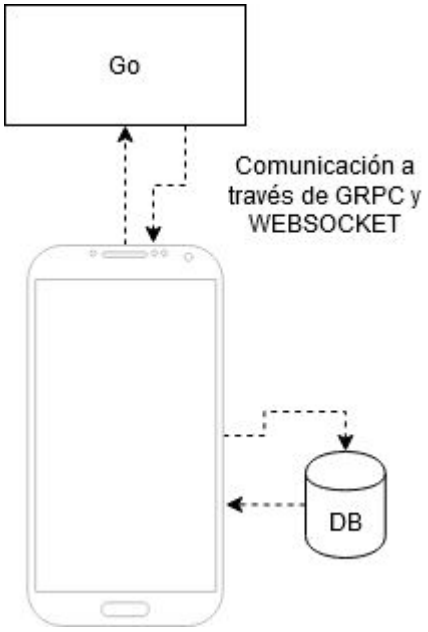
Plan de trabajo Flutter

Se creará una aplicación mobile híbrida [22] para mostrar información proveniente del backend[1] hecho en Go.

El hecho de decidir por una aplicación híbrida viene de que la aplicación sólo va a recibir/enviar información a través del protocolo GRPC, no va a tener muchas interacciones con los componentes del dispositivo mobile entonces se hace mucho más fácil cubrir los dos sistemas operativos más populares del mercado sin tener que escribir dos aplicaciones por separado.

[1] Cuando hablamos de backend, nos referimos a un servidor que centraliza y procesa la información en un sólo lugar.

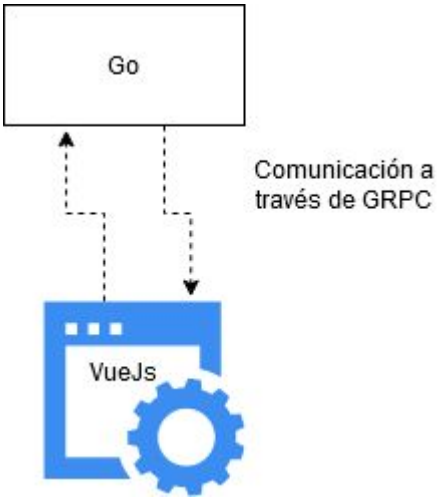
Diagrama mobile Flutter



Plan de trabajo Vue js

Se creará una aplicación web [23] para mostrar la información de todos los usuarios, será un panel administrativo.

Diagrama web VueJs



Tecnologías

El proyecto consta de 4 distintos lenguajes:

1. C++. Para programar el Arduino.
2. Go. Para programar el backend.
3. Flutter. Para programar la aplicación mobile.
4. VueJS. Para programar la aplicación web.

Enfoque de software libre

Éste proyecto estará totalmente disponible bajo licencia **GPLv3** [24], tanto el código fuente de la implementación del Arduino en C++, el backend en Go, la aplicación mobile ee Flutter y la aplicación web en VueJs.

¿Por qué se elige este enfoque?

- **Revisión entre compañeros:** dado que se puede acceder al código fuente libremente y que la comunidad open source es muy activa, los colegas programadores verifican y mejoran el open source. Considéralo un código vivo, en lugar de un código privado y estancado.
- **Transparencia:** ¿necesita saber exactamente qué tipos de datos se trasladan y a dónde, o qué clase de cambios se aplicaron en el código? Con el open source, se puede verificar esta información y realizar un seguimiento, sin tener que depender de las promesas de los proveedores.
- **Confiabilidad:** el código propietario depende de un solo autor o una sola empresa que lo controlan para mantenerlo actualizado, con parches y en funcionamiento. El open source sobrevive a sus autores originales porque las comunidades open source activas lo actualizan constantemente. Los estándares abiertos y la revisión entre compañeros garantizan que el open source se evalúa de manera regular y adecuada.
- **Flexibilidad:** dado el énfasis del open source en la modificación, puede utilizarlo para abordar los problemas específicos de su empresa o comunidad. No está obligado a usar el código de una manera específica, y puede contar con la ayuda de la comunidad y la revisión entre compañeros al momento de implementar soluciones nuevas.
- **Menor costo:** con el open source, el código en sí es gratuito.

- **Sin dependencia de un solo proveedor:** la libertad para el usuario significa que puede trasladar el open source a cualquier parte y usarlo para lo que sea en cualquier momento.
- **Colaboración abierta:** las comunidades open source activas brindan la posibilidad de buscar ayuda, recursos y puntos de vista que trascienden el interés de un grupo o una empresa.

Desde ese enfoque es por lo cual se utilizan 3 lenguajes de programación open source [25], tanto como los motores de base de datos, creemos que ser consecuentes en nuestras acciones es una de las mejores formas de demostrar el apoyo a la causa open source.

CAPÍTULO 4. EXPERIMENTACIÓN EN IOT

Luego de realizar la planificación necesaria para llevar adelante el proyecto, se comenzó a probar los diferentes sensores de Arduino.

El primer prototipo que se realizó fue la configuración del sensor de WiFi (Fig 4) a la placa Arduino y mediante la exposición de un sitio web, poder prender y apagar un led.

Prototipo WiFi

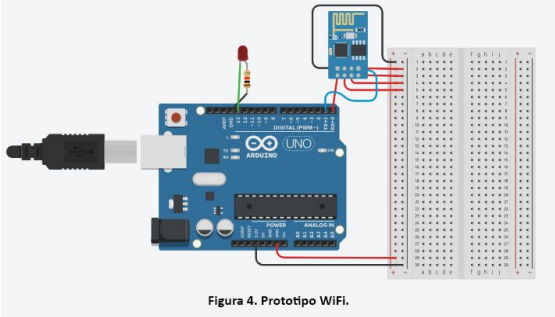


Figura 4. Prototipo WiFi.

Prototipo Sensor Temperatura

Implementación de prototipo de sensor de calor (Fig 5).

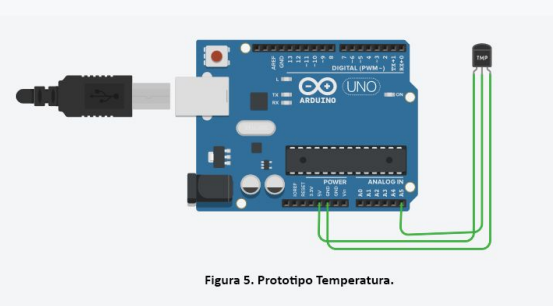


Figura 5. Prototipo Temperatura.

Prototipo Sensor Lector de Huella

Para éste prototipo nos basamos en un prototipado existente y replicaremos su funcionamiento ya que satisface las necesidades básicas del prototipo (Fig 6).

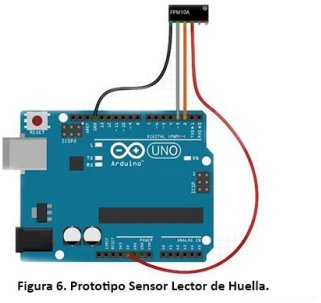


Figura 6. Prototipo Sensor Lector de Huella.

La realización de los tres prototipos permite un acercamiento casi ideal para la creación del dispositivo IoT final.

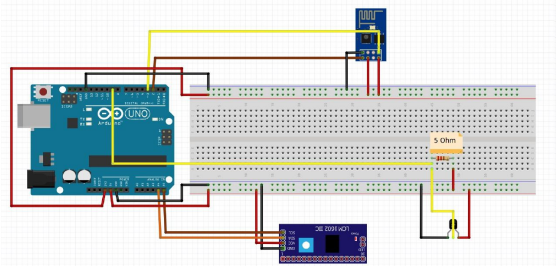
Se experimentó de manera muy satisfactoria con circuitos electrónicos y con los diferentes módulos/sensores que se utilizaron. Gracias a la cantidad de información que hay en disponible y de fácil acceso, teniendo un conocimiento básico de electrónica, siguiendo guías y leyendo sobre los módulos/sensores, se pudo llegar a un muy buen resultado.

CAPÍTULO 5. PROTOTIPADO

Se logró un prototipo totalmente funcional, pudiendo interconectar el dispositivo IoT con el backend, la aplicación mobile y las aplicación web, a continuación se dejan referencias de las diferentes aplicaciones.

Todos los repositorios van a estar alojados en la plataforma GitHub [27].

Para el prototipado de IoT con Arduino se comparte un diagrama hecho en el sitio web Tinkercad [26] con todos sus componentes utilizados y el código fuente hecho en C++.



Cada uno de los repositorios tiene un archivo README.ME sobre la utilización de cada una de las aplicaciones.

A continuación se deja el link del repositorio.

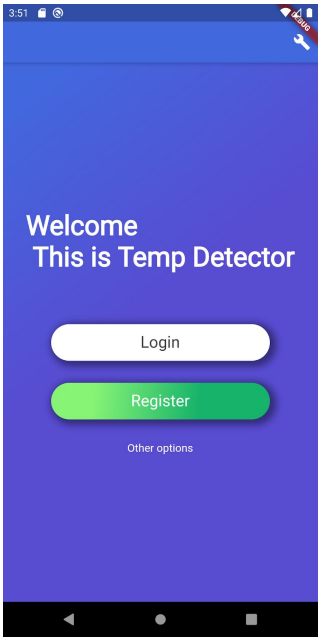
- https://github.com/rodrigoherera/tesis-temp-detector

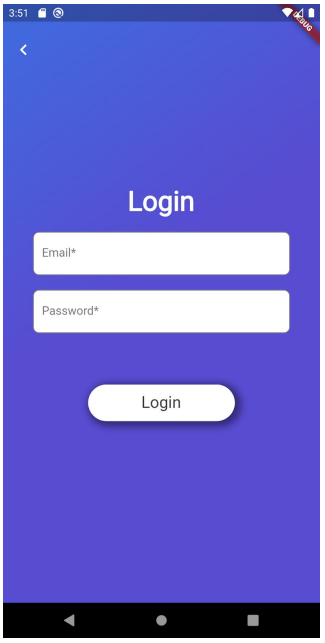
Dentro del repositorio vamos a tener 4 proyectos:

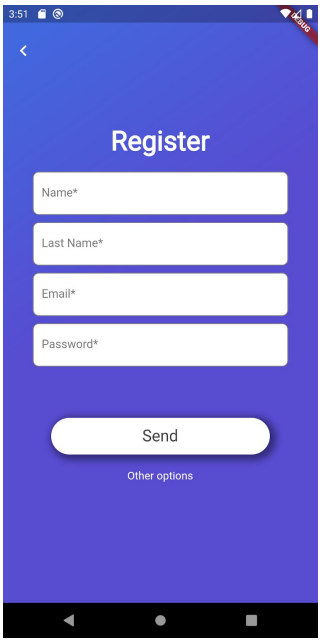
1. c_temp_detector es la aplicación IoT hecha en C++.
2. flutter_temp_detector es la aplicación mobile hecha en Flutter.
3. go-temp-detector es la aplicación backend hecha en Go.
4. vue-temp-detector es la aplicación web hecha en Vue.

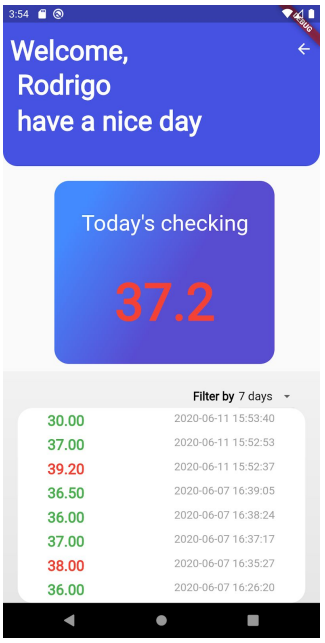
A continuación se presentan unas pantallas de los desarrollos visuales finales.

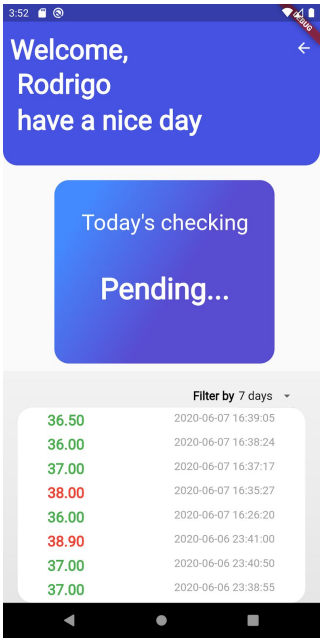
Aplicación mobile:

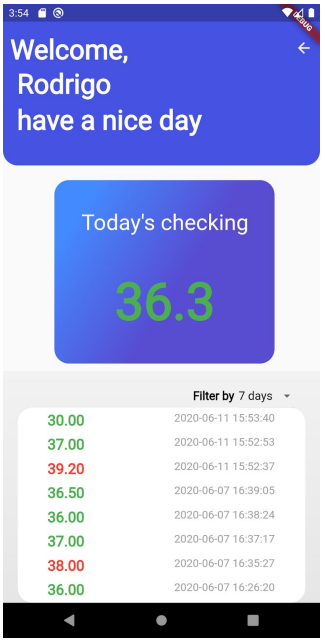




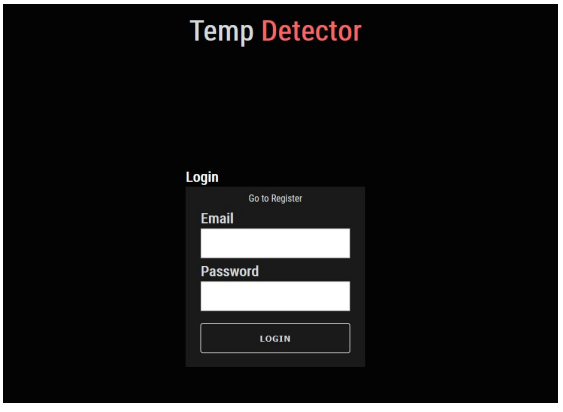


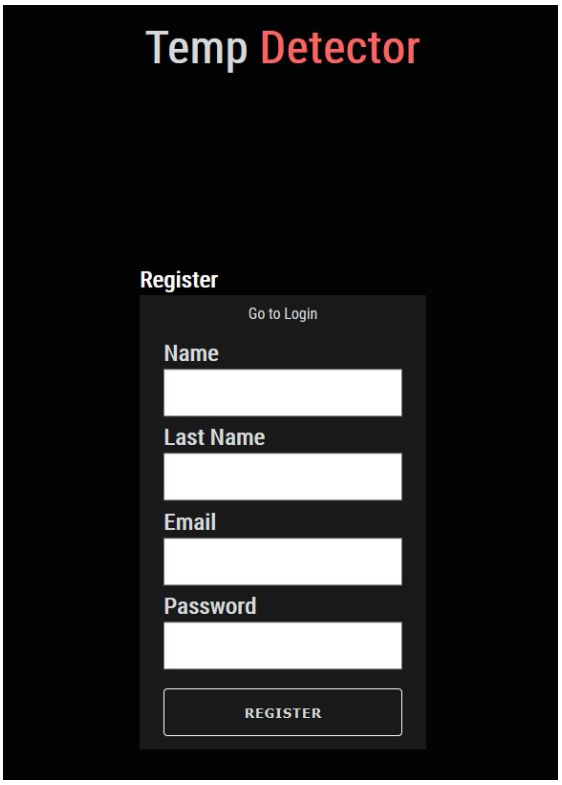


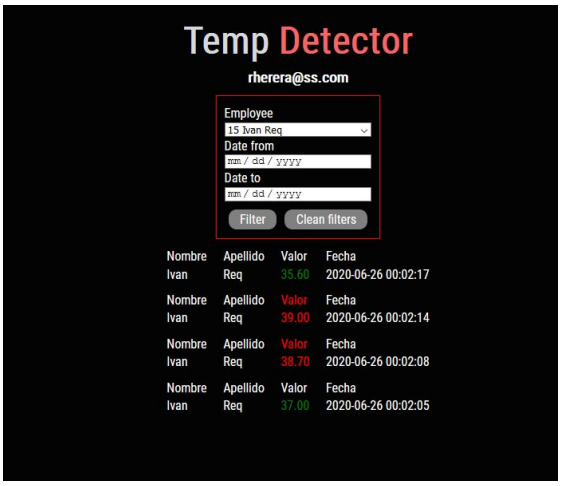




Aplicación web:







CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS

Se lograron ampliamente las metas propuestas para la creación del Temp Detector.

Por un lado, toda la investigación y puesta en marcha el arduino fue lo que más tiempo demoró debido a la situación actual tanto de Argentina como del mundo en general.

Los sensores base del proyecto, principalmente el de temperatura al tener tanta demanda, fue lo último en llegar pero aún así se realizó una simulación fiel utilizando tecnologías alternas, así poder solventar ese problema.

Para las demás herramientas, se pudo realizar de forma rápida el backend con la aplicación mobile, así poder probar como sería una experiencia real de un posible usuario.

Acompañado de una UX/UI intuitiva en conjunto con una gran fluidez, se logra una muy buena experiencia de usuario.

Para el panel administrativo se mantuvo la simplicidad de sólo mostrar información relevante para quien lo manipule.

Se pudo mantener un coste muy bajo en la creación del arduino, gracias al bajo coste de sus partes.

A continuación se detallan con el valor a la fecha 25/06/2020:

ID	MODELO	PRECIO
1	Arduino Uno	10 USD
2	Sensor MLX90614	28 USD
3	Sensor ESP8266	3 USD
4	Pantalla LED	8 USD
5	Sensor DSP AS606	15 USD

En conjunto, las cuatro herramientas van a proveer una experiencia excelente, siendo usuario tanto como administrador.

Trabajos futuros

Si bien las metas iniciales están totalmente completas, a medida que se fue realizando el proyecto comenzaron a surgir ciertas mejoras que estaría muy bueno agregar y que le agregarían gran valor a la solución como un conjunto.

1. Prototipar case para el Arduino: de momento se entregan los diagramas y

cómo realizar las conexiones de todos los componentes en el Arduino, no tiene una presentación en un case de plástico, por ejemplo. Lo ideal sería idear un case realizando modelado 3D para poder ser impreso y utilizado.

2. Mejorar el manejo de GRPC: las aplicaciones ya están preparadas para hablar autenticandose a través de JWT, sería ideal agregar un middleware grpc para hacer uso y mejorar la seguridad de las aplicaciones.
3. Sistema de alertas: para lograr el ideal a la aplicación web, se debería crear un sistema de alertas configurables para notificar en casos particulares.

REFERENCIAS

1. Coronavirus disease (COVID-19) 2020.

   World Health Organization.

2. K. Vasanth, J. Sbert, "*Creating solutions for health through technology innovation". Texas Instruments*, Dic. 2014.

   Disponible online.

3. "What is open source?".

   OpenSource.com

4. J.Teicher, "The little-known story of the first IoT device", Feb, 2018.

   IBM

5. International Conference for Industry and Academia, Mar, 2008.

   Iot Conference

6. Z. Pang, "Technologies and architectures of the Internet-of-Things (IoT) for health and well-being", Ene. 2013.

   Google Scholar

7. J. Morgan, "A Simple Explanation Of 'The Internet Of Things'", May, 2014.

   Forbes

8. J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, D. Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", Amsterdam, The Netherlands:Elsevier, 2014.

   Google Scholar

9. G. Kortuem, F. Kawsar, D. Fitton, V. Sundramoorthy, "Smart objects as building blocks for the Internet of Things", IEEE Internet Comput., vol. 14, pp. 44-51, Ene./Feb. 2010.

   Google Scholar

10. K. Romer, B. Ostermaier, F. Mattern, M. Fahrmair, W. Kellerer, "Real-time search for real-world entities: A survey", *Proc. IEEE*, vol. 98, no. 11, pp. 1887-1902, Nov. 2010.

    Google Scholar

11. A. Gabbai, "Kevin Ashton Describes the Internet of Things", Ene, 2015.

    SmithSonian Magazine

12. Google Trends, búsqueda de la palabra clave "IoT", con un rango de fechas de 2004 a hoy.

    Google Trends

13. I. Gronbaek, "Architecture for the Internet of Things (IoT): API and interconnect", Proc. Int. Conf. Sensor Technol. Appl., pp. 802-807, Ago. 2008.

    Google Scholar

14. Arduino Uno

    PDF

15. Sciforce, "Embedded Programming for the Internet of Things", May, 2019.

    IoTForAll

16. Sensors for IoT.

    TeConnectivity

17. Rapid Prototyping.

    Engineering Product Design

18. Golang.

    Golang

19. MQTT.

    Mqtt

20. GRPC.

    Grpc

21. Websockets - Api.

    Mozilla

22. Flutter.

    Flutter

23. Vue Js.

    Vue

24. Proyecto GNU - GPLv3.

    Gnu

25. ¿Que es el open source?.

    RedHat

26. Tinkercad

    Tinkercad

27. Github

    Github.