

# FERIA NACIONAL DE INNOVACIÓN EDUCATIVA

## INFORME TÉCNICO

ETPB - 1  
ESPECIALIDAD: COMPUTACIÓN

2018 - AÑO DEL CENTENARIO DE LA REFORMA UNIVERSITARIA

---



---

*Alumnos expositores*  
IARA FRYC  
GUSTAVO YBAÑEZ

*Docente asesor*  
SANTIAGO TRINI

ET N° 37 "Hogar Naval Stella Maris"  
Pergamino 211, CABA  
Tel: 011-4612-3220

# Índice

<b>1</b>	<b>Resumen</b>	<b>3</b>
1.1	Origen . . . . .	3
1.2	Objetivo . . . . .	3
1.3	Participantes . . . . .	3
1.4	Método . . . . .	5
1.5	Costo . . . . .	5
1.6	Producto . . . . .	5
<b>2</b>	<b>Introducción</b>	<b>5</b>
2.1	Antecedentes . . . . .	5
2.2	Justificación del proyecto . . . . .	6
2.3	El problema . . . . .	8
2.4	Solución propuesta . . . . .	8
2.5	Objetivos . . . . .	9
2.5.1	Termógrafo . . . . .	9
2.5.2	Aplicación móvil . . . . .	10
2.5.3	El <i>backend</i> . . . . .	10
<b>3</b>	<b>Desarrollo</b>	<b>10</b>
3.1	Metodología . . . . .	10
3.2	Materiales . . . . .	11
3.3	Actividades . . . . .	12
3.3.1	Investigación . . . . .	12
3.3.2	ESP8266 . . . . .	13
3.3.3	Sensor de temperatura . . . . .	13
3.3.4	SPIFFS . . . . .	13
3.3.5	RTC . . . . .	14
3.3.6	Aplicación para Android . . . . .	14
3.3.7	Armado de la caja . . . . .	15
3.3.8	Participación en la feria regional . . . . .	15
3.3.9	Aplicación web . . . . .	16
3.3.10	Gráfico de temperaturas . . . . .	16
3.3.11	Revisión del termógrafo . . . . .	16
3.3.12	Configuración inicial del termógrafo . . . . .	19
3.3.13	Visita al CeSAC N° 37 . . . . .	19
3.3.14	Participación en "Mi factura por favor" . . . . .	19
3.3.15	Aplicación móvil . . . . .	20

<b>4</b>	<b>Resultados y análisis</b>	<b>20</b>
4.1	Investigación . . . . .	20
4.2	Termógrafo . . . . .	23
4.3	Aplicación para Android . . . . .	23
4.4	Aplicación web . . . . .	24
<b>5</b>	<b>Conclusiones</b>	<b>26</b>
5.1	Utilidad de MERVA para el sistema de salud . . . . .	26
5.2	Viabilidad de MERVA . . . . .	26
5.3	Futuro del proyecto . . . . .	26
<b>6</b>	<b>Agradecimientos</b>	<b>27</b>
<b>7</b>	<b>Bibliografía</b>	<b>28</b>
<b>8</b>	<b>Anexo I: Devoluciones</b>	<b>29</b>
<b>9</b>	<b>Anexo II: Código fuente</b>	<b>33</b>
9.1	Termógrafo . . . . .	33
9.2	Aplicación web . . . . .	42

# **1 Resumen**

## **1.1 Orígen**

El proyecto que se presenta en este informe surgió en el contexto del aula debido a las inquietudes de los alumnos y al vínculo establecido el año pasado en la última feria de ciencias. Los alumnos involucrados querían trabajar en un proyecto de Internet de las Cosas debido a que alumnos de otros cursos habían estado investigando el uso de plataformas como el Nodemcu. La idea disparadora fue la siguiente: una conservadora para vacunas capaz de ser monitoreada por WiFi desde un teléfono celular. Al poco tiempo de empezar a investigar el proyecto tomó su forma actual. Un sistema de monitoreo de vacunas por red para el último nivel de la cadena de frío. MERVA: Monitoreo En Red de Vacunas.

## **1.2 Objetivo**

MERVA es un sistema online que permite centralizar el registro de temperatura de los vacunatorios ya sea en las heladeras convencionales o las conservadoras de telgopor.

Nuestro proyecto busca una solución económica para darle más transparencia y confiabilidad al manejo de la cadena de frío de vacunas en el nivel local. La solución que proponemos es crear un sistema de monitoreo donde cada heladera es un nodo en una red de sensores, y donde toda la información relativa a la temperatura de las vacunas es registrada y almacenada en una base de datos. También de esta manera se puede certificar el correcto mantenimiento de la cadena de frío en su último eslabón.

## **1.3 Participantes**

El equipo de trabajo está conformado por:

- Iara Fryc (4° año)
- Gustavo Ybañez (4° año)
- Thomas Docampo (4° año)
- Ignacio Cedrola (4° año)
- Franco Mateos (4° año)
- Franco Rabaiotti (4° año)

- Ezequiel Fernandez (4° año)
- Renzo Torrisi (4° año)
- Luana Acosta (4° año)
- Sol Casamian (4° año)
- Tomas Burza (4° año)
- Mathias Banegas (4° año)
- Teo Costa (4° año)
- Pablo Bordigoni (4° año)
- Lautaro Fino (4° año)
- Romina Velazquez (4° año)
- Pablo Molina (4° año)
- Ayrton De Oliveira (4° año)
- Franco Michelini (4° año)
- Ezequiel Genoves (4° año)
- Enzo Medina (4° año)
- Yanina Erquicia (4° año)
- Nicolás Ávila (4° año)
- Fabio Bielat (5° año)
- Alex Giulliano (5° año)
- Cristian Leguizamón (5° año)
- Luciano Marti (5° año)
- Santiago Trini (Docente asesor)

La gran mayoría de los alumnos que participaron pertenecen a 4<sup>o</sup> ya que el proyecto surgió en la materia Base de Datos. Pero a lo largo del mismo se fueron sumando otros alumnos y recibimos la ayuda de varios profesores de la escuela.

## 1.4 Método

El método de trabajo adoptado consiste en un ciclo de vida iterativo del proyecto. Se realizaron varias iteraciones de lo que se conoce como círculo PDCA (*Plan - Do - Check - Act*) para ir mejorando continuamente el producto final. No obstante no se dejó de lado una planificación del alcance y el tiempo a largo plazo, teniendo en cuenta las instancias de participación y las fechas límites.

## 1.5 Costo

El costo del prototipo hasta ahora ronda los \$800 y creemos que está justificado dado el costo de las vacunas del calendario oficial.

## 1.6 Producto

Para nosotros el éxito de este proyecto depende de la finalización de un prototipo del sistema de monitoreo de la cadena de frío y de una prueba del mismo en un entorno real del ámbito de la salud. El sistema está formado por tres productos diferenciados:

- Un termógrafo electrónico capaz de enviar datos de manera inalámbrica.
- Una *app* capaz de recolectar esos datos y generar reportes.
- Un servidor web capaz de recibir datos de la *app* y del termógrafo y almacenar los registros en una base de datos en la nube.

# 2 Introducción

## 2.1 Antecedentes

La escuela técnica 37 "Hogar Naval Stella Maris" había participado en la instancia nacional de la feria de ciencias en el año 2017. En esa oportunidad había tenido la suerte de colaborar con el equipo autor del trabajo que llegó a esa instancia. Lo hice en calidad de asesor y orientador de alumnos de tercer año, formando un vínculo estrecho especialmente con los alumnos que hoy me acompañan como expositores.

Nuestra escuela tiene como objetivo la formación de técnicos en computación: profesionales capacitados para diseñar e implementar soluciones

informáticas y automatizar tareas de la esfera socioproductiva. El énfasis del plan de estudios está puesto en el diseño y desarrollo de aplicaciones y bases de datos. En función de este objetivo a mí me toca dictar en el cuarto año de estudios las materias Base de Datos y Lógica Computacional.

Estos alumnos que habían colaborado en el proyecto de realidad aumentada para la educación inicial y primaria el año pasado quedaron realmente cautivados con la experiencia, a pesar de que ese proyecto no se había originado en su aula. Por lo tanto me puse como objetivo para este ciclo lectivo organizar un equipo de trabajo en el aula durante el dictado de mis materias para darles la oportunidad esta vez de participar como protagonistas, como el equipo autor de un proyecto.

Durante el mes de marzo el profesor Eduardo Ortega me mostró que estaba trabajando con una plataforma, Nodemcu, que fue pensada como una alternativa a Arduino orientado netamente a la Internet de las Cosas. Mis alumnos el año pasado habían aprendido algunos rudimentos sobre Arduino en el Taller de Tecnología y Control del tercer año y enseguida supimos que queríamos hacer un proyecto con esta tecnología, un producto de IoT.

Fue también el año pasado, en el proyecto de realidad aumentada, que los alumnos habían estado trabajando con el App Inventor del MIT, desarrollando apps para Android. Android, IoT y bases de datos, todos eran conceptos que despertaban el interés de los alumnos y tecnologías con un potencial increíble para desarrollar una buena parte de los contenidos curriculares de este año. Pero necesitábamos un proyecto concreto, una solución dónde implementar estas tecnologías tuviera sentido y dónde los contenidos curriculares de mi asignatura estuvieran bien representados.

Entonces surgió en el aula una idea: una conservadora para vacunas capaz de ser monitoreada por WiFi desde un teléfono celular. La idea llegó a través de uno de mis alumnos al notar el transporte de medicamentos en conservadoras de telgopor corrientes en el subte. Al poco tiempo de empezar a investigar el proyecto tomó su forma actual y se fue distanciando de la idea original porque detectamos un problema que creemos digno de atención. Nuestro proyecto es un sistema de monitoreo de vacunas por red para el último nivel de la cadena de frío. MERVA: Monitoreo en Red de Vacunas.

## **2.2 Justificación del proyecto**

Las vacunas son un producto muy sensible a la temperatura. La eficacia de las campañas de vacunación dependen en buena medida de la correcta

conservación de la cadena de frío. De acuerdo a las recomendaciones de la OMS, la cadena de frío para el almacenaje, transporte y distribución de vacunas está organizada en tres niveles: central, regional y local. El último eslabón de esta cadena es también el más débil, donde los controles son menos frecuentes y la tecnología utilizada más rudimentaria. Una recopilación de estudios realizados en distintos países arrojan cifras preocupantes y nos indican que en la actualidad los centros de salud y los vacunatorios locales son en donde la cadena de frío corre mayor riesgo.



Fig. 1: Niveles de la cadena de frío

El concepto de la Internet de las Cosas ha ido tomando más relevancia en los últimos años y pensamos que tal vez podría acercarnos a una solución o mejora de las deficiencias en el manejo de la cadena de frío en el nivel local. La electrónica asociada a este tipo de soluciones es cada año más económica y decidimos buscar una solución al problema mencionado aprovechando estas nuevas tecnologías.

La investigación y capacitación sobre estas tecnologías, a mi parecer, son un elemento motivante en el desarrollo de los conocimientos y destrezas técnicas de los alumnos. Además de aportar continuidad a lo realizado el año pasado, amplía el panorama de conocimientos introduciendo contenidos que de otra manera no son apropiados por los alumnos. La propuesta también se complementa en un buen grado con los contenidos



de Base de Datos, Algoritmos y Estructuras de Datos y de Proyecto Informático del cuarto año, así como con los contenidos de las asignaturas de los años posteriores.

Por último, el proyecto abordado es de carácter integrador entre las dos áreas de la especialidad del Técnico en Computación: la parte relativa al hardware y redes y el desarrollo de aplicaciones y software en general.

### **2.3 El problema**

El problema abordado es la implementación de nuevas tecnologías para mejorar los controles de la cadena de frío y asegurar la eficacia de las campañas de vacunación como también disminuir el desperdicio de las dosis en los vacunatorios. El control de la cadena e frío en los vacunatorios puede ser automatizado para que no dependan o estén sujetos a errores humanos en el proceso.

### **2.4 Solución propuesta**

MERVA es un sistema online que permite centralizar el registro de temperatura de los vacunatorios y centros de salud, ya sea en las heladeras convencionales o las conservadoras de telgopor.

Nuestro proyecto busca una solución económica para darle más transparencia y confiabilidad al manejo de la cadena de frío de vacunas en el nivel local. La solución que proponemos es crear un sistema de monitoreo dónde cada conservadora o heladera es un nodo en una red de sensores, y dónde toda la información relativa a la temperatura de las vacunas es registrada y almacenada en una base de datos.

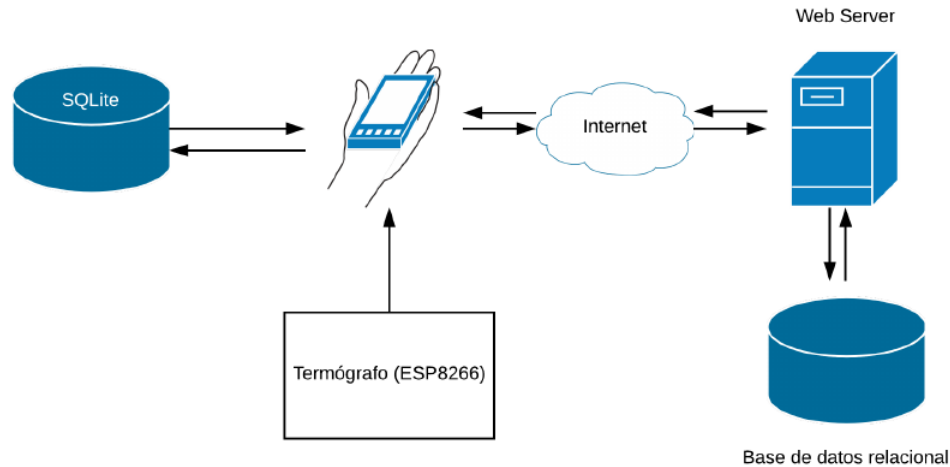


Fig. 2: Arquitectura del sistema, versión preliminar luego de la investigación.

## 2.5 Objetivos

Nuestros objetivos son:

- Crear un termógrafo electrónico con conectividad WiFi y con una capacidad de registro y monitoreo en tiempo real.
- Crear una app para smartphones capaz de conectarse al termógrafo para monitorear y ver reportes del registro de temperaturas.
- Proveer un sistema online dónde periódicamente se recopilen los datos de temperatura de los distintos centros de salud para una mayor transparencia y control.

### 2.5.1 Termógrafo

El termógrafo debe poder monitorear temperaturas de manera precisa y registrarlas con fecha y hora en tiempo real. El dispositivo debe ser económico y debe poder funcionar dentro de una heladera. Debe tener dos modos de funcionamiento: conectado a Internet debe enviar los datos a un servidor remoto y en caso de desconexión los registros tienen que poder ser leídos directamente del aparato por medio de una conexión inalámbrica

con un *smartphone*. Los registros deben ser almacenados internamente en el aparato por un tiempo prolongado hasta que puedan ser persistidos de manera definitiva en una base de datos remota.

### 2.5.2 Aplicación móvil

La app móvil debe poder consumir los registros de temperaturas almacenados en el *backend* y generar diversos reportes y gráficos que resulten adecuados para el personal de salud. Además debe poder conectarse de manera directa por WiFi a un termógrafo para recibir los registros de manera directa desde el aparato. La aplicación tiene que persistir los datos de manera local y subirlos periódicamente al servidor remoto cuando esté conectada a Internet.

### 2.5.3 El *backend*

La aplicación web debe proveer reportes y gráficos de la misma manera que la aplicación móvil, pero también es la encargada de persistir y centralizar la información de todos los *data loggers* en una única base de datos relacional. Además debe proveer una API para que la aplicación móvil pueda acceder a esta base de datos.

## 3 Desarrollo

### 3.1 Metodología

La metodología adoptada para el proyecto es de índole iterativa. Es decir, se comienza con la investigación y se planifica que se quiere hacer. Luego se desarrolla y se realizan las pruebas. Después de extraer las conclusiones y detectar nuevos problemas volvemos a repetir el ciclo.

Pensamos que este círculo de mejora continua o círculo de Deming es adecuado para nuestro proyecto ya que nos enfrentamos a un ámbito y a un sistema que desconocemos y que vamos acercándonos a una solución en una elaboración progresiva. Esto es evidente en el momento mismo en el que al comenzar a investigar con nuestra idea original de realizar una conservadora "inteligente" nos encontramos con nuevos problemas sobre los que profundizar nuestra indagación.

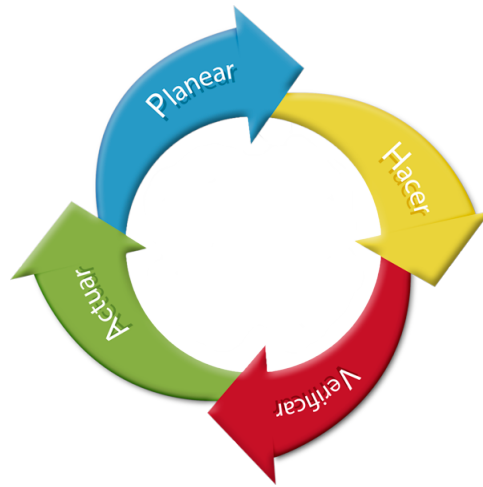


Fig. 3: Círculo de Deming (PDCA)

### 3.2 Materiales

Para este proyecto usamos materiales que teníamos disponibles en el laboratorio y compramos algunos otros. Usamos algunas cosas que teníamos a mano como el sensor de temperatura para las primeras pruebas, aunque más adelante lo cambiamos por uno de mayor precisión y resolución. La siguiente lista resume los materiales utilizados al momento de entregar este informe.

<i>Item</i>	<i>Costo</i>
RTC DS3231	\$80
Wemos D1 Mini	\$300
DS18B20	\$120
Cables	\$0
Shield de Batería	\$220
Smartphone	\$0
Caja plástica	\$80
Cargador portátil para celular	\$0
<b>TOTAL</b>	<b>\$800</b>

Tabla 1: Materiales utilizados

Más adelante mientras seguimos desarrollando el código del proyecto decidimos utilizar el sensor de temperatura que viene en el reloj de tiempo real y tratamos de bajar el costo del aparato al mínimo sacando la pantalla LCD que originalmente habíamos usado.

### 3.3 Actividades

#### 3.3.1 Investigación

Se investigó a través de la web buscando documentación oficial sobre inmunización del Ministerio de Salud y se le consultó a personal de salud sobre la misma. Tomamos la documentación local como referencia para el manejo de la cadena de frío. Seguimos investigando encontrando una nota periodística, distintas publicaciones de la OMS sobre cadena de frío y vacunas y estudios de publicaciones médicas sobre problemas en la cadena de frío.

También se realizó una entrevista en el CeSAC N° 13 a un enfermero encargado de llevar los controles de frío y stock del vacunatorio que nos mostró los elementos utilizados y la documentación de registro que ellos utilizan.

Recopilamos toda esta información en la carpeta de campo y decidimos que había buenos fundamentos para defender la utilidad del proyecto. Al

terminar la investigación nos dimos cuenta que el enfoque más provechoso para nuestro proyecto era el de crear un sistema de monitoreo y control para el nivel local de la cadena de frío.

### 3.3.2 ESP8266

Se investigó con los alumnos las capacidades de la plataforma que íbamos a utilizar, el Wemos D1, y se comenzó a realizar pruebas en el laboratorio para entender el funcionamiento del mismo. Se realizó la instalación de las herramientas y librerías para trabajar con el ESP8266 y se copiaron los links con documentación técnica del integrado y de las librerías. Los alumnos realizaron pruebas utilizando tutoriales y la documentación recolectada compilando algunos programas sencillos sobre la plataforma antes de desarrollar el programa para el termógrafo.

### 3.3.3 Sensor de temperatura

Se utilizó el sensor de temperatura y humedad DHT11 ya que teníamos uno en la escuela y comenzamos a realizar pruebas enviando la información por WiFi y levantando la información desde un navegador web.

Se conectó todo en una protoboard para realizar las pruebas y se logró un simple monitor de temperatura conectado al smartphone. Después de esta actividad se decidió ampliar el alcance del proyecto ya con la idea de realizar un *data logger* y el proyecto fue tomando su forma actual.

### 3.3.4 SPIFFS

Cómo había que mantener un registro de las temperaturas tomadas por el sensor incluso cuándo éste no estuviera conectado a Internet o al smartphone se decidió investigar alguna forma de persistir los datos utilizando la misma placa controladora.

Se decidió realizar pruebas con la memoria flash del Wemos, para lo cual se buscó documentación en Internet. Se investigó sobre el manejo de archivos en la memoria flash utilizando SPIFFS (*Serial Peripheral Interface Flash Filing System*). Los alumnos realizaron un programa sencillo que registre los datos tomados por el sensor en un archivo de valores separados por comas.

Se optó por esta opción en vez de usar una tarjeta SD para guardar la información, debido a que encarecería el costo del producto, necesitando

otro módulo conectado al controlador para poder leer los datos desde una tarjeta de memoria.

### 3.3.5 RTC

Como todo *data logger*, necesitábamos también tener registro de la fecha y hora en la que se realizaban las mediciones. La opción más utilizada en estos casos es un integrado RTC (*Real Time Clock*). Se gestionó la compra del módulo mediante cooperadora sin ningún inconveniente. Los alumnos realizaron la conexión del módulo RTC al ESP8266 y se instaló una librería para simplificar el manejo del mismo. Se realizaron pruebas en el laboratorio para agregar un *timestamp* a cada registro en el .csv.

### 3.3.6 Aplicación para Android

Después de terminar el modelo básico del termógrafo se comenzó a investigar cómo realizar la *app* en Android Studio. Se instalaron las herramientas necesarias y se empezó a desarrollar el código para la aplicación. Como los alumnos no tenían experiencia previa con Java ni con Android Studio se estudiaron distintas apps de ejemplo y comenzaron a estudiar el material de entrenamiento para desarrollo de Android provisto por Google.

Finalmente se decidió adoptar las prácticas recomendadas en los manuales de Google para la arquitectura de la aplicación: los *Architecture Components* que facilitan de manera notable el manejo de la base de datos SQLite a través de una serie de clases y objetos que cubren las necesidades comunes de un buen número de aplicaciones.

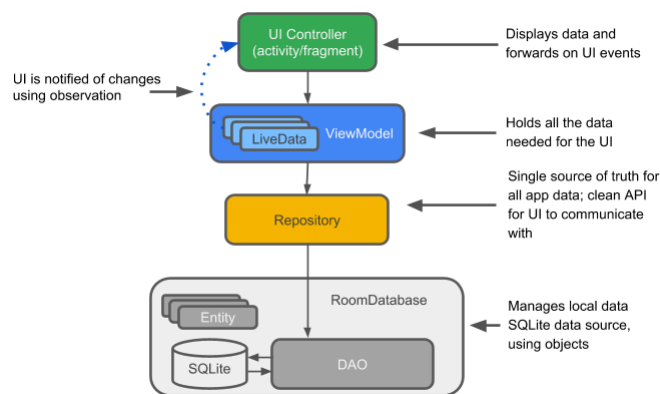


Fig. 4: Architecture Components

Los alumnos desarrollaron una aplicación muy simple que pudiera leer el archivo de registros en el termógrafo y persistirlo en una base de datos local, que en Android se realiza con SQLite. Los registros de la base de datos en el celular se actualizan cada vez que la app se conecta al termógrafo y de esta manera se puede dar una vista en la app de la lista de temperaturas con su tiempo correspondiente, destacando la última temperatura de la heladera registrada por el termógrafo.

Para esta primer versión de la app se dejaron muchas cosas pendientes debido a que necesitábamos un prototipo para mostrar el día de la feria regional.

### 3.3.7 Armado de la caja

Se realizó la conexión de los componentes electrónicos y se dispuso todo en un gabinete plástico. Se realizó una ventana en la caja para dejar visible una pantalla LCD mostrando la temperatura actual. Se decidió usar un cargador portátil de celular como fuente de alimentación.



Fig. 5: Primer prototipo del termógrafo

### 3.3.8 Participación en la feria regional

El 4 de junio participamos de la feria regional en la ET 27. Los alumnos tuvieron la oportunidad de recibir *feedback* sobre el proyecto y ensayar la exposición del mismo. Se realizaron dos afiches como material de apoyo gráfico resumiendo los puntos más importantes del proyecto. Los expositores recibieron felicitaciones de los asistentes a la exposición y clasificaron para la instancia jurisdiccional de octubre.



Por otro lado recibimos la visita de personal de Presidencia de la Nación que felicitaron a los chicos por su proyecto y nos invitaron a participar del concurso Escuelas Solidarias.

### 3.3.9 Aplicación web

Terminada la feria regional se empezó a desarrollar el último componente del sistema: la aplicación web. Para esto los alumnos comenzaron a capacitarse en NodeJS y Express, tecnologías que se utilizan hoy en día para el desarrollo del *backend* de una aplicación. Es decir, todo el código que corre sobre un servidor en la web.

En este momento del proyecto también se empezó a definir la arquitectura de todo el sistema de manera más detallada, incluyendo la interconexión e integración de los tres componentes principales del sistema.

Para la base de datos centralizada se decidió utilizar MySQL, ya que los alumnos aprenden esta tecnología en la materia Base de Datos. Se diseñó una base de datos sencilla capaz de persistir los registros de temperatura y se empezó a trabajar en la conexión entre esta base de datos y una aplicación web realizada con el framework Express en JavaScript.

La aplicación web además debía proveer vistas adecuadas para que los usuarios del sistema pudieran acceder a los registros también a través de un navegador web. Para las vistas se decidió utilizar un lenguaje de plantillas HTML, y las tecnologías web usuales para darles un diseño atractivo como Bootstrap para el CSS.

### 3.3.10 Gráfico de temperaturas

Como uno de los requisitos del sistema es poder visualizar rápidamente los valores de las temperaturas a lo largo del tiempo se tenía que dar a la aplicación una vista en forma de gráfico. Para esto los alumnos utilizaron una librería de JavaScript: Chart.js de manera que el servidor sólo provea los datos en crudo (temperatura y tiempo) y el cliente, es decir el navegador web, se encargue de procesar y dibujar un gráfico en base a esos datos.

### 3.3.11 Revisión del termógrafo

Mientras se desarrollaba la aplicación para el servidor web se decidió simplificar el diseño del termógrafo utilizando sólo el controlador y el módulo RTC. También hubo que depurar algunos *bugs* del funcionamiento del RTC

y actualizar el código del mismo para realizar los distintos modos de conexión a la red. Tanto de manera directa a la app de Android, funcionando como *access point*, como conectándose a un router y a través de la red (Internet) al servidor web. Para este último punto procuramos un router WiFi y los alumnos realizaron la configuración del mismo.

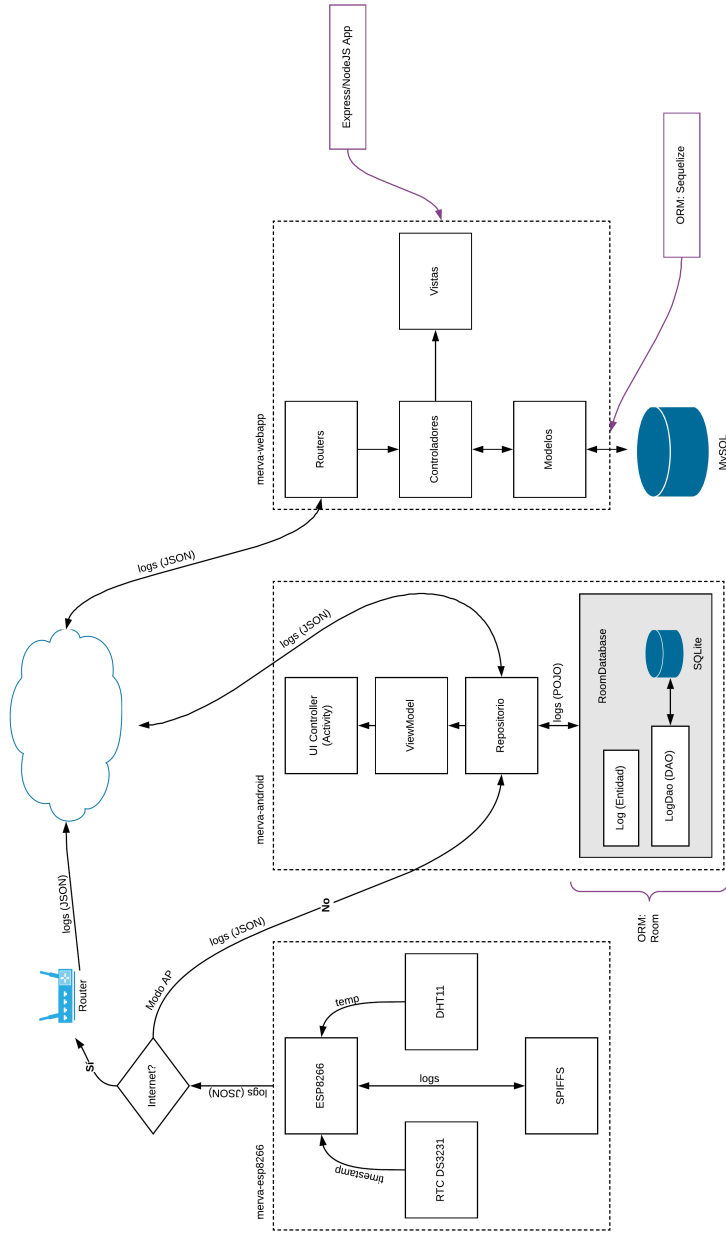


Fig. 6: Arquitectura del sistema, después de comenzar el desarrollo del *backend*

### **3.3.12 Configuración inicial del termógrafo**

Uno de los problemas que tuvieron que resolver los alumnos es el de la configuración inicial del termógrafo. ¿Cómo sabe el aparato que construimos a qué red conectarse? El problema es muy similar a instalar un *chromecast* por primera vez en un televisor. El aparato tiene que realizar un *broadcast* de una red WiFi y hay que conectarse a esa red para darle los parámetros del router inalámbrico al que queremos conectarnos.

Se decidió realizar esta configuración utilizando la aplicación de Android que se venía desarrollando. Para eso, al iniciar el aparato por primera vez hay que conectarse a él mismo como si fuera un *access point* y pasarle a través de un formulario en el celular el nombre y la contraseña de la red de nuestro router. También se tuvo que implementar lógica en el aparato para que vuelva a funcionar como *access point* en el caso de la caída de la red del router, permitiendo así una conexión directa con la aplicación del teléfono.

### **3.3.13 Visita al CeSAC N° 37**

Los alumnos expositores visitaron por su cuenta el CeSAC 37 en Mataderos para recabar información y material fílmico para el stand. El padre de Iara les abrió la puertas ya que trabaja allí y pudieron mostrarles un prototipo del sistema al personal del vacunatorio obteniendo una retroalimentación muy importante.

### **3.3.14 Participación en "Mi factura por favor"**

Las profesoras Margarita Gonzalez y Silvia Guarracino, que participan con un grupo de alumnos en el concurso "Mi factura por favor" de la AGIP necesitaban un proyecto solidario para la presentación del 8 de octubre en relación a éste último concurso. Ya que Iara y Gustavo participan en el desarrollo de este proyecto y en el concurso de la AGIP, se decidió que MERVA sirva como proyecto solidario a desarrollar con el premio que ofrece la AGIP.

Debido a esto los alumnos que integran el grupo de éste concurso se sumaron al proyecto entusiasmados, ya que de ganar MERVA, tendríamos disponible el dinero del premio para armar más termógrafos con el objetivo de repartirlos en los centros de salud con los que hicimos contacto de manera gratuita. De esta manera obtendríamos una retroalimentación valiosa sobre el funcionamiento del sistema en un entorno real.

### **3.3.15 Aplicación móvil**

Se siguió trabajando sobre el código de la aplicación móvil para dar una interfaz unificada entre la versión de Android y la aplicación web. Se consideraron distintas alternativas a Android Studio, como MIT App Inventor y React Native. Al final los alumnos decidieron seguir desarrollando la *app* en MIT App Inventor debido a que hubo bastantes dificultades para desarrollar en Android Studio. Todavía se sigue trabajando sobre la misma ya que hay que considerar los distintos casos de uso del sistema. Se necesita trabajar sobre el aspecto relativo a la seguridad e integridad de los datos, punto pendiente aún en el desarrollo del proyecto.

## **4 Resultados y análisis**

En la siguiente sección se presentan los resultados obtenidos hasta el mes de octubre y el análisis de los mismos.

### **4.1 Investigación**

La investigación realizada arrojó datos preocupantes y nos dió impulso a seguir con nuestra idea. Las encuestas e inspecciones resumidas en los artículos encontrados nos indican que incluso en países desarrollados el control de la cadena de frío de las vacunas en el nivel local puede ser insuficiente. Se verificó que el total de centros vacunatorios que realizaban un control de temperatura al menos una vez al día podía estar casi en el 25% del total. Esto es en base a estudios realizados entre 1992 y 2002 en países del primer mundo. Estos estudios se resumen en la tabla 2 indicando a qué año pertenece el estudio y que resultados arrojó.

<i>Año del estudio</i>	<i>Control al menos una vez al día %</i>
1992	20,00
1992	2,50
1995	3,70
1998	20,00
1999	73,33
2001	19,61
2002	97,67
<b>Total</b>	<b>26,88</b>

Tabla 2: Prevalencia de vacunatorios en los que se realiza el control de temperatura al menos una vez al día según estudios de distintos años.

En la tabla 3 se resumen también los estudios recopilados sobre la preservación del rango óptimo de temperatura en las heladeras destinadas a las vacunas, mostrando claras falencias al momento de la visita de los investigadores que realizaron los estudios.

<i>Año del estudio</i>	<i>Temperatura 2 - 8 °C %</i>
1992	78,00
1995	7,41
1995	68,15
1996	62,16
1999	85,71
1999	72,00
1999	59,38
2001	31,37
2002	93,02
2002	78,78
<b>Total</b>	<b>70,68</b>

Tabla 3: Prevalencia de vacunatorios con heladeras en el rango óptimo de temperatura.

Sin embargo necesitamos de un estudio estadístico a nivel local para confirmar o desestimar del todo nuestra hipótesis de trabajo y la utilidad del sistema propuesto. Además cabe destacar que la lectura de publicaciones especializadas indica que el calor no es el único problema en la conservación de las vacunas. Y que actualmente es mayor el desperdicio de dosis por congelación de las mismas en algunas vacunas sensibles a temperaturas inferiores a cero grados centígrados.

Por otro lado la literatura especializada advierte sobre el uso del test de agitación como medio para determinar la correcta conservación de las dosis. Y que si bien es útil este método, no siempre se puede aplicar debido a la presentación de las vacunas. Por lo tanto un control riguroso de la temperatura es esencial para garantizar la efectividad de las mismas.

La entrevista que realizamos en el CeSAC N° 13 sin embargo nos dió un panorama más alentador. Además la información provista por Diego Raffo en el CeSAC no sólo nos sirvió en lo relativo a la cadena de frío, ya que nos informó sobre los procedimientos, formularios y protocolos para cuidar la conservación de las vacunas. Estos detalles nos permiten orientar nuestro

proyecto de manera que resulte útil para la comunidad de los centros de salud de la Ciudad de Buenos Aires en lo específico de su forma de control.

## 4.2 Termógrafo

Todavía faltan pulir algunos aspectos del dispositivo, pero consideramos que al menos tenemos una prueba de factibilidad para proseguir nuestro trabajo. Logramos completar un *data logger* funcional y deberíamos poder resolver los detalles que surjan de las pruebas de funcionamiento en un entorno más próximo a la realidad.

El termógrafo tiene dos modos de funcionamiento relativos a su conexión a la red. En modo AP (*access point*) nos provee de una red WiFi a la cual conectarse, de manera similar a un router. Si nos conectamos a esta red desde la aplicación de Android podemos ver los registros que el aparato almacena con su fecha, hora y temperatura. La aplicación de Android almacena automáticamente estos registros cuando se conecta al dispositivo. También en este modo podemos configurar al termógrafo para que se conecte a un router WiFi y acceda a Internet.

El segundo modo de funcionamiento se denomina *station* en la jerga de redes inalámbricas y es cuando le damos al *data logger* los parámetros de nuestra red WiFi. En este modo todos los datos que el aparato registra son enviados por Internet al servidor web para que los almacene en la base de datos y puedan ser consultados desde cualquier lugar tanto desde la app como desde un navegador.

## 4.3 Aplicación para Android

Al momento de la feria regional la aplicación no era más que una prueba de concepto que realizamos para testear la conexión con el ESP8266. Sin embargo gracias al uso los Architecture Components de Google ya teníamos una sólida base para el manejo de la persistencia de los logs dentro de la app, en una base de datos SQLite.

La aplicación, de manera similar al termógrafo cumple dos funciones. Si está conectado a Internet recibe los datos directamente desde el servidor web y puede generar vistas con los mismos, tanto en forma de una lista de registros como en forma de gráfico. También es importante destacar en la vista principal de la app la última temperatura registrada por el termógrafo.

Si el teléfono no está conectado a Internet, debido tal vez a que no tenemos red o estamos en una situación de uso donde en general no tenemos



acceso a una red WiFi, la app es igualmente capaz de realizar una conexión directa al termógrafo siempre y cuando se encuentre en el rango de alcance de la radio WiFi del dispositivo y así descargar los últimos registros de temperatura directamente desde el termógrafo. Cuando el teléfono se conecta nuevamente a Internet la app es responsable por subir los registros al servidor actualizando de esta manera la base de datos.

#### 4.4 Aplicación web

La aplicación web que se realizó para el proyecto sirve dos propósitos:

- Proveer un servidor web capaz de persistir los registros en una base de datos relacional, ya sea que provengan directamente del termógrafo o de un smartphone a través de Internet.
- Proveer una serie de vistas generadas a partir de los datos almacenados.

Una aplicación web, conocida en la jerga del diseño web como *backend*, es el código que se ejecuta en un servidor conectado a Internet. Esta aplicación tiene que procesar la información que recibe a través de la red y persistirla. En nuestro caso se trata de una aplicación de JavaScript capaz de almacenar los registros que llegan a través de la web desde el termógrafo (cuando se encuentra conectado a un router WiFi) o desde un teléfono que tenga la app de MERVA.

El servidor es el encargado de validar esos datos y persistirlos, constituyendo así un repositorio central de los registros de temperatura de los distintos termografos que estén conectados. Para eso se sirve de una base de datos relacional central común a todo el sistema.

El *backend* a diferencia de la aplicación móvil siempre recibe los registros a través de Internet ya que lo más probable es que en un caso de uso real se encuentre alejado del termógrafo y debe recibir los registros de más de un dispositivo. No tiene la posibilidad de conectarse directamente a las termógrafos porque su objetivo es simplemente centralizar la información.

Pero por este motivo puede proveer reportes gráficos y en forma de texto a cualquier usuario a través de un navegador web desde cualquier parte, sin necesidad de instalar la aplicación móvil.

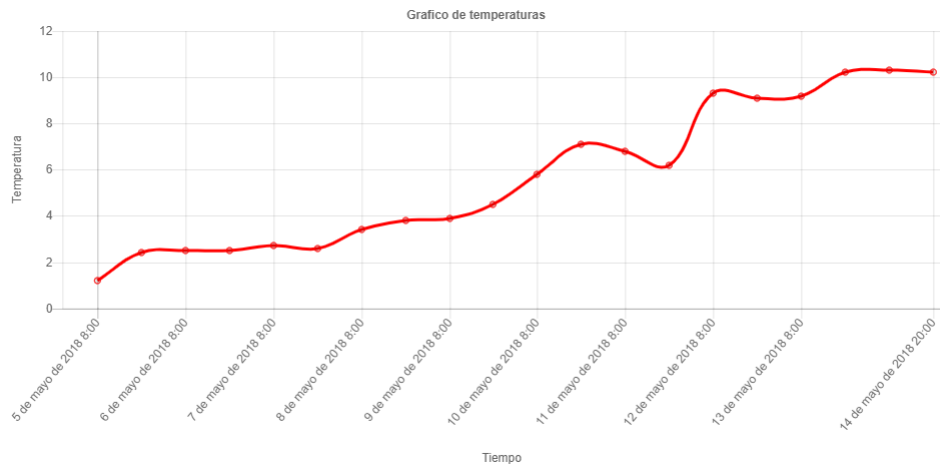


Fig. 7: Gráfico de temperaturas en el navegador.

## 5 Conclusiones

### 5.1 Utilidad de MERVA para el sistema de salud

Si bien todavía hay muchos detalles que pulir y mucho trabajo por realizar el equipo del proyecto confía plenamente en la utilidad de un sistema de monitoreo central para los programas de inmunización regionales. Ponemos el énfasis en el nivel local de la cadena de frío porque es dónde creemos que la tecnología aún no llega y se puede desarrollar a un bajo costo una solución técnica en relación a dos objetivos:

- Automatizar el control de la cadena de frío.
- Monitorear de manera centralizada la cadena de frío.

Los grandes almacenes de vacunas, tanto a nivel nacional, como regional ya cuentan con tecnología de control, ya que los volúmenes que se manejan en las cámaras destinadas a la guarda de vacunas representan mucho dinero, pero indefectiblemente las dosis tienen que pasar por los centros de salud y los vacunatorios donde la tecnología que se aplica para grandes cantidades de dosis resulta prohibitiva en lo económico en la pequeña escala.

### 5.2 Viabilidad de MERVA

Esperamos que el producto que comenzamos a desarrollar sirva como prueba de su viabilidad y que lo podamos probar dentro del sistema de salud para poder mejorarlo y convertirlo en una herramienta indispensable para certificar la correcta conservación de las dosis de los vacunatorios.

### 5.3 Futuro del proyecto

Después de la participación en la feria todavía tenemos mucho trabajo por realizar. A partir de ahora debemos poner el énfasis en la socialización del proyecto. Nuestra idea es construir más *data loggers* y llevarlos a los centros de salud y hospitales con los que hicimos contacto para probar su funcionamiento y relevar el impacto que tiene en los lugares de trabajo.

Además, para que el producto sea viable tenemos que estudiar seriamente las vulnerabilidades de seguridad del sistema porque los registros tienen que ser inviolables y el sistema tiene que garantizar la integridad de los mismos.

## 6 Agradecimientos


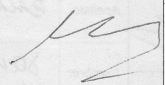
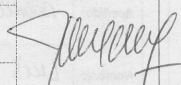
Extendemos nuestros más sinceros agradecimientos a las autoridades del colegio, el rector Ángel Merce y el vicerector Claudio Chapachnik. También a los profesores Nahuel Silva Dassis, Osvaldo Marcovecchio y especialmente Eduardo Ortega por brindarnos su asesoramiento. No podríamos haber llegado a este punto sin la ayuda de la Asociación Cooperadora Hogar Naval Stella Maris que nos brindó su apoyo en lo económico. Agradecemos la confianza depositada en nuestro proyecto a las profesoras Silvia Guarracino y Margarita González que nos instaron a tomar una iniciativa solidaria y llevar nuestro proyecto a más gente.

Un especial agradecimiento al personal de salud: Diego Raffo, por la entrevista brindada en el CeSAC N° 13 y a las enfermeras y doctoras del CeSAC N° 37. No puedo olvidar a mi hermana Julia Trini y al padre de Iara, Walter Fryc por habernos abierto las puertas en el área de salud pública y habernos orientado en la dirección correcta en temas que desconocíamos..

## 7 Bibliografía

- AAVV. (2013). Guía de los fundamentos para la dirección de proyectos (guía del PMBOK®) – Quinta edición.
- AAVV. (2015). Immunization in practice. A practical guide for health staff. OMS.
- AAVV. (2012). Recomendaciones nacionales de vacunación argentina. Ministerio de Salud de la Nación Argentina.
- AAVV. (2006). Temperature sensitivity of vaccines. OMS.
- Benedetti, D. (7 de abril de 2018). Tuvieron que tirar 55 mil vacunas porque se cortó la cadena de frío. Diario Clarín.
- Ortega Molina P, Astasio Arbiza P, Albaladejo Vicente R, Arrazola Martínez P, Villanueva Orbáiz R, Ramón de Juanes Pardo J. (2007). Mantenimiento de la cadena del frío para las vacunas: una revisión sistemática. Gac Sanit.
- Ruiz García, M. (2008). ¿Qué debemos saber de la cadena del frío? Ponencia realizada en las IX Jornadas sobre Vacunas en Atención Primaria.

## 8 Anexo I: Devoluciones

	<b>Ministerio de Educación e Innovación de la Ciudad de Buenos Aires</b> <b>1.ª Feria Regional de Educación, Artes, Ciencias y Tecnología</b>
<b>PLANILLA DE DEVOLUCIÓN DE EVALUACIÓN</b>	
<b>PROYECTO:</b> MORVA	<b>CÓDIGO:</b>
<b>INSTITUCIÓN:</b> ET. Nº 37 DE 11	<b>UBICACIÓN:</b>
<b>EVALUADORES:</b> RODRIGUEZ, Luis y BALADO, Daniel.	
<b>FORTALEZAS</b>	
<b>ASPECTOS INSTITUCIONALES:</b> Se ofrece un apoyo y colaboración permanente del equipo directivo.	
<b>ASPECTOS DEL PROYECTO:</b> Es pertinente con el perfil profesional del técnico. - Identifica la situación problemática. - La opinión de los alumnos es clara y da cuenta del conocimiento del tema. -	
<b>ASPECTOS PEDAGÓGICOS:</b> La elección del tema es pertinente al diseño curricular. -	
<b>EVALUADOR 1</b>	<b>EVALUADOR 2</b>
Apellido: RODRIGUEZ	Apellido: BALADO
Nombre: Luis	Nombre: Daniel
 Firma del Evaluador	 Firma del Evaluador



Ministerio de Educación e Innovación de la Ciudad de Buenos Aires

1.º Feria Regional de Educación, Artes, Ciencias y Tecnología

PLANILLA DE DEVOLUCIÓN DE EVALUACIÓN

PROYECTO: <i>Proyecto Merva</i>	CÓDIGO:
INSTITUCIÓN: <i>ET 37</i>	UBICACIÓN:
EVALUADORES: <i>Balado, Rodriguez</i>	

**SUGERENCIAS**

ASPECTOS INSTITUCIONALES:

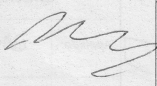
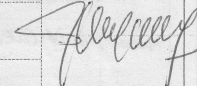
.....

ASPECTOS DEL PROYECTO:

*Se sugiere el apoyo de elementos propios (carteles afiches) para la participación del público -  
Se recomienda observar un mayor repertorio para la carpeta de trabajo.*

ASPECTOS PEDAGÓGICOS:

*Se recomienda la participación de docentes de otros espacios curriculares.*

EVALUADOR 1		EVALUADOR 2	
Apellido: <i>RODRIGUEZ</i>		Apellido: <i>BALADO</i>	
Nombre: <i>Luis</i>		Nombre: <i>Daniela</i>	
Firma del Evaluador		Firma del Evaluador	



PLANILLA DE DEVOLUCIÓN DE EVALUACIÓN

PROYECTO: <u>MERVA</u>	CÓDIGO: <u>ETPB1</u>
INSTITUCIÓN: <u>ETNº 37</u>	UBICACIÓN: <u>CABA</u>

EVALUADORES: STANCATO, GUSTAVO. MAZZIOTTO FOUNDO

FORTALEZAS

ASPECTOS INSTITUCIONALES:

NOTABLE COMPROMISO DE LA INSTITUCIÓN EN LA PARTICIPACIÓN EN FERIA DE CIENCIAS Y EN LA CONSTRUCCIÓN DE LAS EMPRESAS ASOCIADAS.

ASPECTOS DEL PROYECTO:

- 1. SE ALCANZA UNA PARALELIDAD REAL Y CONCRETA FINANCIA EN LA INVESTIGACIÓN CON DISTINTOS ÁMBITOS DE LA SOCIEDAD.
- 2. MUY BUEN TRABAJO DE INVESTIGACIÓN DE CAMPO QUE INCLUYEN LAS TÉCNICAS DE PROFESIONALES ESTUDIOS Y ENSAYOS EN LABORATORIOS, ETC.
- 3. BUENA PRESENTACIÓN DE STAND.
- 4. EXPOSICIÓN UN PRODUCTO TERMINADO Y RELACIONADO.
- 5. EL PROYECTO PUEDE DEVENIR EN UN EQUIPO COMERCIALIZABLE EN TODO EL MUNDO.
- 6. EL PRODUCTO PUEDE SER TOMADO COMO DE USO ORGÁNICO EN VACUNACIONES.

ASPECTOS PEDAGÓGICOS:

- 1. NOTABLE COMPROMISO POR PARTE DE LOS ALUMNOS.
- 2. COHERENCIA ENTRE EL PROYECTO Y EL PERFIL PNT DE LAS EMPRESAS.
- 3. CONSTRUCCIÓN DE SOBRES ENTRE DISTINTAS DISCIPLINAS INICIADOS DESDE LA ESCUELA.

EVALUADOR 1		EVALUADOR 2	
Apellido: <u>STANCATO</u>		Apellido: <u>MAZZIOTTO</u>	
Nombre: <u>GUSTAVO</u>		Nombre: <u>FOUNDO</u>	
	Firma del Evaluador		Firma del Evaluador





Ministerio de Educación e Innovación de la Ciudad de Buenos Aires

1.ª Feria Regional de Educación, Artes, Ciencias y Tecnología

PLANILLA DE DEVOLUCIÓN DE EVALUACIÓN

PROYECTO: <u>MERVA</u>	CÓDIGO: <u>ETPBI</u>
INSTITUCIÓN: <u>ET. Nº 37</u>	UBICACIÓN: <u>CABA</u>
EVALUADORES: <u>STANCATO, GUSTAVO. MAZZUZZA, FOUUNDO.</u>	

**SUGERENCIAS**

ASPECTOS INSTITUCIONALES: SUGERIMOS TENER EN CUENTA:  
DAJ PARTICIPACION A OTROS GRUPOS CON EL FIN DE  
ESTIMULAR COSTOS, RESPONSABILIDAD, PRESENTACION DE TRABAJOS  
DISCOS, ETC.

ASPECTOS DEL PROYECTO:

RECOMENDAMOS DAR HACER CRECER EL PROYECTO:

- SEPARAR LA ETAPA DE CONTROL DE LOS SENSORES  
CON EL FIN DE GARANTIZAR LA PROTECCION DE LOS  
CIRCUITOS ELECTRONICOS
- SE RECOMIENDA NO SOLO DESTINAR EL DISPOSITIVO  
AL MONITORIO SINO AL ALTO DE ANIMACIONES  
EN LOS VALORES DE TEMPERATURA REQUERIDA.
- RECOMENDAMOS TRABAJAR EN LA AUTONOMIA DEL FUNCIONAMIENTO  
CON EL FIN DE ESTUDIAR DE DISTINTOS SENSORES  
CON EL FIN DE ELEJIR EL MAS ADECUADO.
- INCLUIR CIRCUITOS ELECTRONICOS
- INCLUIR CODIGO DEL PROGRAMA
- INCLUIR INFORMACION DE LAS APLICACIONES  
PRESENTADAS.

ASPECTOS PEDAGÓGICOS: ACONSEJAMOS:

- INCLUIR EN EL REGISTRO PEDAGOGICO  
MAÑANERA LABORAL EN LA DESCRIPCION  
DE TRABAJO SEMANAL CUANDO ESTO SEA POSIBLE.

EVALUADOR 1		EVALUADOR 2	
Apellido: <u>STANCATO</u>		Apellido: <u>MAZZUZZA</u>	
Nombre: <u>GUSTAVO</u>	Firma del Evaluador	Nombre: <u>FOUNDO</u>	Firma del Evaluador

## 9 Anexo II: Código fuente

### 9.1 Termógrafo

```
1 // merva.ino
2
3 // LIBRERIAS
4
5 #include <ESP8266WiFi.h>
6 #include <ESP8266WebServer.h>
7 #include <FS.h>
8 #include <ESP8266HTTPClient.h>
9 #include <Wire.h>
10 #include <RtcDS3231.h>
11 #include <OneWire.h>
12 #include <DallasTemperature.h>
13
14 // CONSTANTES
15
16 #define ONE_WIRE_BUS 0
17
18 const char* host = "192.168.0.100"; // Server IP
19 const char* url = "/logs"; // Ruta para el POST
20 const int port = 3000; // Server Port
21 const int delayEntreLogs = 30000;
22
23 // GLOBALES
24
25 RtcDS3231<TwoWire> Rtc(Wire);
26 OneWire oneWire(ONE_WIRE_BUS);
27 DallasTemperature DS18B20(&oneWire);
28 char ssid[63] = "merva"; // ssid del router
29 char pass[63] = "merva1234"; // pass del router
30 String creado = "";
31 float temp = 0.0;
32 const int termografo_id = 1;
33 unsigned long desdeUltimoLog = 0;
34 boolean ap = false, station = false;
35 boolean primeraVezAP = true;
36 ESP8266WebServer server(80);
```

```

37 HTTPClient http;
38
39 // ***** SETUP *****
40
41 void setup() {
42     delay(2000);
43     Serial.begin(115200);
44     // setup del filesystem (memoria flash)
45     setupFS();
46     // setup del RTC DS3231
47     setupRTC();
48     // setup del ds18b20
49     DS18B20.begin();
50     // primero intento conectar con ssid y pass al router
51     conectarRouter(ssid, pass);
52     if (WiFi.status() == WL_DISCONNECTED) {
53         // inicio el aparato en modo AP sin pass
54         WiFi.softAP("mervaAP");
55         Serial.println("estoy funcionando como AP");
56         Serial.println(WiFi.softAPIP());
57         ap = true;
58         station = false;
59     }
60 }
61
62 // ***** MAIN LOOP *****
63
64 void loop() {
65     // si voy como AP
66     if (ap) {
67         Serial.println("Modo AP");
68         // que el server se ocupe de los clientes que se conectan
69         // registrar temps
70         actualizarTmpTime();
71         delay(100);
72         if (timeout(desdeUltimoLog, delayEntreLogs))
73             escribirArchivo(creado, temp);
74         // manejar conexiones al AP
75         if (primeraVezAP) {
76             startAPServer();

```

```

77     primeraVezAP = false;
78 }
79     server.handleClient();
80 }
81 // si voy como station
82 else if (station) {
83     Serial.println("Modo station");
84     // registrar temps y enviarlas a mysql
85     actualizarTmpTime();
86     if (timeout(desdeUltimoLog, delayEntreLogs)) {
87         escribirArchivo(creado, temp);
88         leerArchivoLogs();
89     }
90     if (WiFi.status() == WL_DISCONNECTED) station = false;
91 }
92 else {
93     WiFi.softAP("mervaAP");
94     Serial.println("Modo AP");
95     Serial.println(WiFi.softAPIP());
96     ap = true;
97     station = false;
98 }
99 }
100
101 // funcion conectarRouter con parametros ssid y pass, devuelve true si tiene exito
102 // o false si llega a un timeout
103
104 boolean conectarRouter(char ssid[63], char pass[63]) {
105     int comienzo = millis();
106     WiFi.begin(ssid, pass);
107     while (WiFi.status() != WL_CONNECTED) {
108         Serial.print(".");
109         delay(200);
110         if (timeout(comienzo, 10000)) {
111             Serial.println();
112             station = false;
113             ap = true;
114             return false;
115         }
116     }

```

```

117     Serial.println();
118     Serial.println("Conectado al router");
119     Serial.println(WiFi.localIP());
120     station = true;
121     ap = false;
122     primeraVezAP = true;
123     return true;
124 }
125
126 // funcion timeout devuelve true si pasaron mas de n segs desde comienzo
127
128 boolean timeout(int comienzo, int fin) {
129     int ahora = millis();
130     boolean res = false;
131     (ahora - comienzo > fin) ? res = true : res = false;
132     return res;
133 }
134
135 // funcion para enviar los logs a la webapp
136
137 void subirLogs(String creado, String temp) {
138     http.begin(host,port,url);
139     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
140     http.POST("creado=" + creado + "&temp=" + temp + "&termografo_id=" + termografo_id);
141     http.end();
142     Serial.println("INSERT OK, esperando 50 ms");
143     delay(50);
144 }
145
146 // inicializa el filesystem
147
148 void setupFS() {
149     SPIFFS.begin();
150     SPIFFS.remove("/log.csv"); // comentar despues
151 }
152
153 // lee los contenidos de log.csv y los manda al webserver
154
155 void leerArchivoLogs() {
156     File file = SPIFFS.open("/log.csv", "r");

```

```

157     if (!file) {
158         Serial.println("No se pudo abrir el archivo");
159     }
160     Serial.println("Leyendo de log.csv ...");
161     while (file.available()) {
162         String creado    = file.readStringUntil(',');
163         String temp      = file.readStringUntil('\n');
164         Serial.print("Timestamp: ");
165         Serial.print(creado);
166         Serial.print("\tTemp: ");
167         Serial.println(temp);
168         subirLogs(creado, temp);
169     }
170     file.close();
171 }
172
173 // escribe un nuevo log en log.csv
174
175 void escribirArchivo(String creado, float temp) {
176     File file = SPIFFS.open("/log.csv", "a");
177     if (!file) {
178         Serial.println("No se pudo abrir el archivo");
179     }
180     Serial.println("Escribiendo en log.csv ...");
181     file.print(creado);
182     file.print(",");
183     file.println(temp);
184     file.close();
185     desdeUltimoLog = millis();
186 }
187
188 // ***** FUNCIONES DEL MODO AP *****
189
190 // inicializa el webserver del modo AP
191
192 void startAPServer() {
193     // declaro la funcion handler para un POST a /config
194     server.on("/config", HTTP_POST, handleConfig);
195     // el handler para la app
196     server.on("/logs", HTTP_GET, handleLogsRequest);

```

```

197     server.on("/ultimo", HTTP_GET, handleUltimoLogRequest);
198     server.begin();
199 }
200
201 // esta funcion manda el ultimo log del archivo
202
203 void handleUltimoLogRequest() {
204     String message = "{";
205     File file = SPIFFS.open("/log.csv", "r");
206     if (!file) {
207         Serial.println("No se pudo abrir el archivo");
208     }
209     Serial.println("Leyendo de log.csv ...");
210     String creado, temp;
211     while (file.available()) {
212         creado = file.readStringUntil(',');
213         temp = file.readStringUntil('\n');
214         Serial.print("Timestamp: ");
215         Serial.print(creado);
216         Serial.print("\tTemp: ");
217         Serial.println(temp);
218     }
219     message += "\"creado\": ";
220     message += creado;
221     message += ", \"temp\": ";
222     message += temp;
223     message += ", \"termografo_id\": ";
224     message += termografo_id;
225     message += "}";
226     file.close();
227     Serial.println(message);
228     server.send(200, "application/json", message);
229 }
230
231 // funcion para enviar los logs en modo AP a la app de Android
232
233 void handleLogsRequest() {
234     // para la app, envia todos los registros en el archivo
235     String message = "[";
236     File file = SPIFFS.open("/log.csv", "r");

```

```

237     if (!file) {
238         Serial.println("No se pudo abrir el archivo");
239     }
240     Serial.println("Leyendo de log.csv ...");
241     while (file.available()) {
242         String creado    = file.readStringUntil(',');
243         String temp      = file.readStringUntil('\n');
244         // DEBUG
245         Serial.print("Timestamp: ");
246         Serial.print(creado);
247         Serial.print("\tTemp: ");
248         Serial.print(temp);
249         Serial.print("\tID Termografo: ");
250         Serial.println(termografo_id);
251         message += "{\"creado\": ";
252         message += creado;
253         message += ", \"temp\": ";
254         message += temp;
255         message += ", \"termografo_id\": ";
256         message += termografo_id;
257         message += "}, \n";
258     }
259     file.close();
260     message += "]\n";
261     Serial.println(message);
262     server.send(200, "application/json", message);
263 }
264
265 // handler del POST para configurar la conexion al router
266
267 void handleConfig() {
268     // los args vienen como string, los paso a array de char
269     server.arg(0).toCharArray(ssid, 63);
270     server.arg(1).toCharArray(pass, 63);
271     server.send(200, "text/plain", "POST OK");
272     conectarRouter(ssid, pass);
273 }
274
275 // ***** FUNCIONES DEL RTC Y DEL SENSOR DE TEMPERATURA *****
276

```



```

277 // Setup del RTC
278
279 void setupRTC() {
280     Serial.print("Compilado: ");
281     Serial.print(__DATE__);
282     Serial.println(__TIME__);
283     Rtc.Begin();
284     RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
285     Serial.println();
286     if (!Rtc.IsDateTimeValid()) {
287         Serial.println("RTC lost confidence in the DateTime!");
288         Rtc.SetDateTime(compiled);
289     }
290     if (!Rtc.GetIsRunning()) {
291         Serial.println("RTC was not actively running, starting now");
292         Rtc.SetIsRunning(true);
293     }
294     RtcDateTime now = Rtc.GetDateTime();
295     if (now < compiled) {
296         Serial.println("RTC is older than compile time! (Updating DateTime)");
297         Rtc.SetDateTime(compiled);
298     }
299     else if (now > compiled) {
300         Serial.println("RTC is newer than compile time. (this is expected)");
301     }
302     else if (now == compiled) {
303         Serial.println("RTC is the same as compile time! (not expected but all is fine)");
304     }
305     Rtc.Enable32kHzPin(false);
306     Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);
307 }
308
309 // Toma el timestamp del RTC y lo guarda como string
310
311 #define countof(a) (sizeof(a) / sizeof(a[0]))
312
313 void saveDateTime(const RtcDateTime& dt) {
314     char datestring[20];
315     snprintf_P(datestring,
316                countof(datestring),

```

```

317         PSTR("%04u-%02u-%02uT%02u:%02u:%02u"),
318         dt.Year(),
319         dt.Month(),
320         dt.Day(),
321         dt.Hour(),
322         dt.Minute(),
323         dt.Second() );
324     creado = datestring;
325 }
326
327 // Actualiza las variables globales creado y temp
328
329 void actualizarTmpTime() {
330     if (!Rtc.IsDateTimeValid()) {
331         Serial.println("RTC lost confidence in the DateTime!");
332     }
333     RtcDateTime now = Rtc.GetDateTime();
334     saveDateTime(now);
335     getTemperature();
336     // DEBUG
337     Serial.print("creado: ");
338     Serial.print(creado);
339     Serial.print("\t temp: ");
340     Serial.print(temp);
341     Serial.print(" °C");
342     Serial.print("\t id_term: ");
343     Serial.println(termografo_id);
344     delay(50);
345 }
346
347 // Funcion para tomar la temperatura del ds18b20
348
349 void getTemperature() {
350     do {
351         DS18B20.requestTemperatures();
352         temp = DS18B20.getTempCByIndex(0);
353         delay(100);
354     } while (temp == 85.0 || temp == (-127.0));
355 }

```

## 9.2 Aplicación web

```
1 // index.js
2
3 // modulos de npm
4 const express = require('express');
5 const path    = require('path');
6 const mysql   = require('mysql2');
7 const moment  = require('moment'); // libreria de JS para laburar con fechas
8
9 // creo el objeto app
10 const app = express();
11
12 // conexion a MySQL
13 const connection = mysql.createConnection({
14   host: 'localhost',
15   user: 'root',
16   database: 'merva'
17 });
18
19 // config de app (para que funcionen los parametros del POST)
20 app.use(express.json());
21 app.use(express.urlencoded({ extended: false }));
22
23 // para que funcionen los archivos estaticos (css por ejemplo)
24 app.use(express.static('public'));
25
26 // config de vistas (views)
27 app.set('view engine', 'pug');
28 app.set('views', './views');
29
30 // funciones enrutadoras (routers)
31 // mostrar index
32 app.get('/', (req, res) => {
33   res.render('index', { title: 'MERVA' });
34 });
35
36 // mostrar logs en forma de lista sacados de la DB
37 app.get('/logs', (req, res) => {
38   // query usando el conector mysql2 (sin ORM)
```

```

39 connection.query(
40   'SELECT * FROM logs L JOIN termografos T ON L.termografo_id = T.id ORDER BY creado
41   (err, rs, md) => {                                     // FUNCION CALLBACK DE la query
42     moment.locale('es');
43     for (let log of rs) {
44       log.creado = moment(log.creado).format('L LTS');
45     }
46     res.render('logs', { title: 'Logs',
47                       logs: rs });
48   }
49 );
50 });
51
52 // logs en formato json
53 app.get('/logs.json', (req, res) => {
54   connection.query(
55     'SELECT * FROM logs',
56     (err, rs, md) => {
57       moment.locale('es');
58       for (let log of rs) {
59         log.creado = moment(log.creado).format('L LTS');
60       }
61       res.json(rs);
62     }
63   );
64 });
65
66 // el ultimo log de la tabla
67 app.get('/ultimo.json', (req, res) => {
68   connection.query(
69     'SELECT creado, temp FROM logs ORDER BY creado DESC LIMIT 1',
70     (err, rs, md) => {
71       moment.locale('es');
72       for (let log of rs) {
73         log.creado = moment(log.creado).format('L LTS');
74       }
75       res.json(rs);
76     }
77   );
78 });

```

```

79
80 // uso el metodo post para cargar logs a la DB
81 // (vienen del formulario o del esp8266)
82 app.post('/logs', (req, res, next) => {
83   // hago un insert usando parametros (los ?)
84   if (req.body.creado) {
85     connection.query(
86       'INSERT IGNORE INTO logs(creado ,temp, termografo_id) VALUES (?,?,?)',
87       [req.body.creado, req.body.temp, req.body.termografo_id],
88       (err, rs, md) => {
89         if (err) res.send('Error');
90         console.log("INSERT OK desde el ESP");
91         res.send('INSERT OK')
92       }
93     );
94   } else {
95     connection.query(
96       'INSERT INTO logs(temp, termografo_id) VALUES (?,?)',
97       [req.body.temp, req.body.termografo_id],
98       (err, rs, md) => {
99         if (err) res.send('Error');
100        else res.send('INSERT OK');
101      }
102    );
103  }
104 });
105
106 // para arrancar el servidor http
107 app.listen(3000, () => {
108   console.log('Server listening on localhost:3000');
109 });

1 // chart.js
2
3 fetch('http://localhost:3000/logs.json')
4   .then((res) => {
5     return res.json();
6   })
7   .then((resJson) => {
8     var datos = resJson;

```

```

9     console.log(datos); // DEBUG
10    var temps = [];
11    var labels = [];
12    for (let log of datos) {
13        temps.push({ x: log.creado, y: log.temp });
14        labels.push(log.creado);
15    }
16    // configuracion de datos de chart.js
17    const data = {
18        labels: labels,
19        datasets: [{ data: temps, fill: false, borderColor: 'red' }]
20    };
21    // config del grafico de chart.js
22    const options = {
23        legend: { display: false },
24        title: {
25            display: true, text: 'Grafico de temperaturas'
26        },
27        scales: {
28            xAxes: [{
29                scaleLabel: { display: true, labelString: 'Tiempo' },
30                ticks: { beginAtZero: true, min: 0 }
31            }],
32            yAxes: [{
33                scaleLabel: { display: true, labelString: 'Temperatura' },
34                ticks: { beginAtZero: true, min: 0 }
35            }]
36        }
37    };
38    // el canvas de html5 donde va a ir el grafico
39    var canvas = document.getElementById('chart').getContext('2d');
40    // aca es donde usamos chart.js para crear el grafico y ponerlo en el canvas
41    var grafico = new Chart(canvas, {
42        type: 'line',
43        data: data,
44        options: options
45    }); // cierre de new Chart
46    }); // cierre de then(jstr) y de fetch

1 // logs.pug

```

```

2
3 doctype html
4 html(lang='es')
5   head
6     title= title
7     link(rel='stylesheet', href='/css/bootstrap.min.css')
8     script(src='/js/Chart.min.js')
9     script(src='/js/moment.min.js')
10    script(src='/js/jquery-3.3.1.js')
11    script(src='/js/bootstrap.min.js')
12  body
13    nav(class="navbar navbar-expand-lg navbar-light bg-light")
14      img(src='/img/merva-logo.png' class='img-fluid' width='300')
15    div(class='container-fluid')
16      div(class='row')
17        div(class='col-4')
18          h1 Registros
19          table(class='table table-striped')
20            thead(class='thead-dark')
21              tr
22                th Fecha y hora
23                th Temperatura
24                th Ubicacion
25            tbody
26              each log in logs
27                tr
28                  td= log.creado
29                  td= log.temp
30                  td= log.ubicacion
31            div(class='col-8')
32              canvas(id='chart')
33              script
34                include chart.js
35            div(class='row')
36          footer(class='page-footer font-small blue')
37            div(class='container-fluid text-center')
38              p ET No 37 "Hogar Naval Stella Maris"

1  -- db.sql
2

```

```

3 CREATE DATABASE merva;
4 USE merva;
5
6 CREATE TABLE termografos(
7     id INT NOT NULL PRIMARY KEY,
8     ubicacion VARCHAR(60)
9 );
10
11 CREATE TABLE logs(
12     id INT PRIMARY KEY AUTO_INCREMENT,
13     creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP(),
14     temp REAL,
15     termografo_id INT,
16     FOREIGN KEY (termografo_id) REFERENCES termografos(id),
17     UNIQUE KEY timestamp_termografo (creado, termografo_id)
18 );
19
20 INSERT INTO termografos VALUES (1, 'CESAC 37'), (2, 'CESAC 13');

```