



Bioingeniería

## ***“ConfortMed: Gestor de consultorios médicos”***

Proyecto Integrador Individual

Electrónica Programable

# ConfortMed

A simple line drawing of a stethoscope, positioned to the right of the word "ConfortMed".

Wendler Tatiana Ailén

1er Cuatrimestre 2024

## **índice**

<b>Introducción.....</b>	<b>2</b>
<b>Implementación.....</b>	<b>3</b>
Diagrama en bloques.....	3
Sensado de temperatura:.....	4
Emisor infrarrojo:.....	4
Control de PC.....	7
Control de Lámpara UV.....	7
Aplicación Móvil.....	7
Funcionamiento.....	9
Diagrama de flujo.....	13
<b>Prototipo de diseño.....</b>	<b>14</b>
<b>Videos.....</b>	<b>15</b>

## **Introducción**

Uno de los principales objetivos del sistema ambulatorio de salud es garantizar una atención efectiva. Para ello, es necesario contar con un plan de organización y gestión que distribuya correctamente las tareas entre los diferentes agentes, de acuerdo a las responsabilidades de cada uno. A su vez, es esencial generar un entorno que permita alcanzar la completa comodidad tanto de los pacientes como de los profesionales tratantes.

El consultorio, al ser el principal lugar de trabajo y el espacio de interacción entre el médico y el paciente, debe permanecer siempre preparado y en condiciones óptimas. En establecimientos con numerosos profesionales de la salud, también es importante poseer un control en tiempo real sobre qué consultorios se encuentran disponibles u ocupados.

Si bien la gestión de los consultorios es una tarea primordial a realizarse, en los centros de salud no suele haber personal específico asignado para dicha labor. Como consecuencia, el médico termina siendo responsable de actividades no relacionadas con la atención directa al paciente, como por ejemplo, la preparación previa del espacio de consulta. Esta situación puede llevar a aumentar los tiempos de atención, ocasionar retrasos en la agenda de turnos y generar un posible malestar tanto para los pacientes como para el propio médico.

Para brindar una solución práctica a este problema, se propone un sistema automatizado de gestión de consultorios denominado “ConfortMed”. Mediante una aplicación móvil se designan los días y horarios de atención del consultorio seleccionado. El dispositivo se encargará de preparar el lugar para la llegada del médico. Enciende la PC y sensa la temperatura para asegurar que la misma sea agradable. De ser necesario, enciende y/o apaga el aire acondicionado del lugar. Fuera de los horarios de atención, enciende una lámpara ultravioleta para desinfectar el espacio. A su vez, a través de la aplicación puede observarse la disponibilidad del consultorio, los horarios próximos de atención, la temperatura actual y las temperaturas promedio, mínimas y máximas.

De ésta manera, no sólo se mejora la organización y se asegura el confort tanto del médico como del paciente, sino que también se libra al personal médico de tareas no relacionadas con la atención directa, permitiéndoles centrarse en su labor principal y mejorando notablemente la calidad de atención.

## **Implementación**

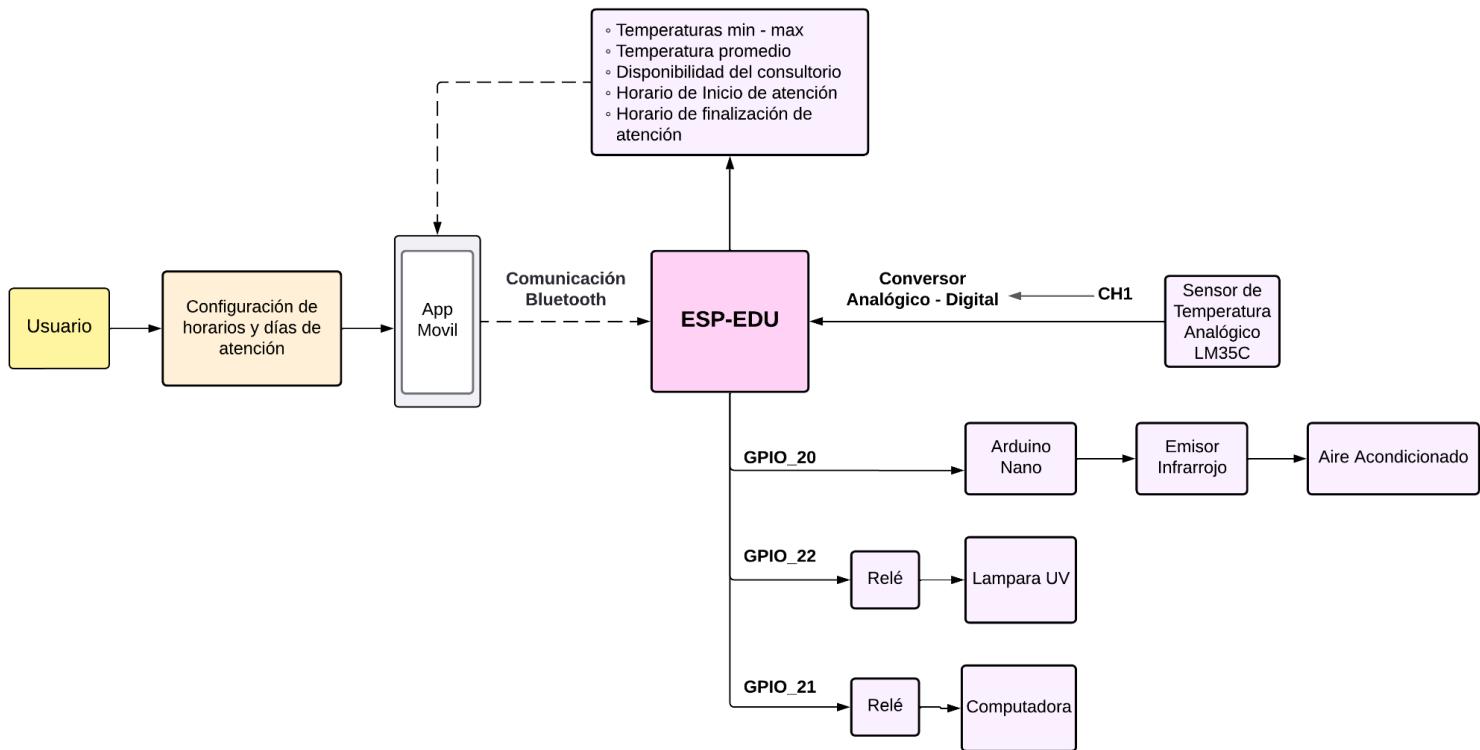
Para la implementación del prototipo se utilizó la ESP-EDU proporcionada por la cátedra y se agregaron los siguientes componentes necesarios para cumplir con los requerimientos de diseño planteados:

- Sensor de temperatura analógico lm35c.
- Módulo relé 2 canales.
- Arduino Nano.
- Receptor infrarrojo.
- Led emisor infrarrojo.
- Transistor NPN 2N3904.
- Resistencia de 100ohm.
- Lámpara Led.
- Lámpara Led color azul.
- Protoboard.
- Cables de conexión.

También se utilizó la aplicación de Android “Bluetooth Electronics” para generar un prototipo de la aplicación a utilizar por el dispositivo.

## **Diagrama en bloques**

El dispositivo posee el siguiente diagrama en bloques:



En el mismo se pueden identificar diferentes módulos. A continuación se especifica la implementación, el funcionamiento y la simulación de cada uno de ellos.

#### Sensado de temperatura:

El mismo está compuesto por el sensor de temperatura analógico lm35c, el cuál envía al microcontrolador una señal en mV equivalente a la temperatura sensada. La misma ingresa a través del CH1 (GPIO\_01) y es leída por el conversor analógico-digital.

Para obtener la temperatura se implementó un driver para el sensor utilizado.

La tarea del microcontrolador encargada de obtener la temperatura actual, también calcula la temperatura mínima, la temperatura máxima y la temperatura promedio y envía dichos datos a la app móvil conectada al dispositivo a través de ble.

#### Emisor infrarrojo:

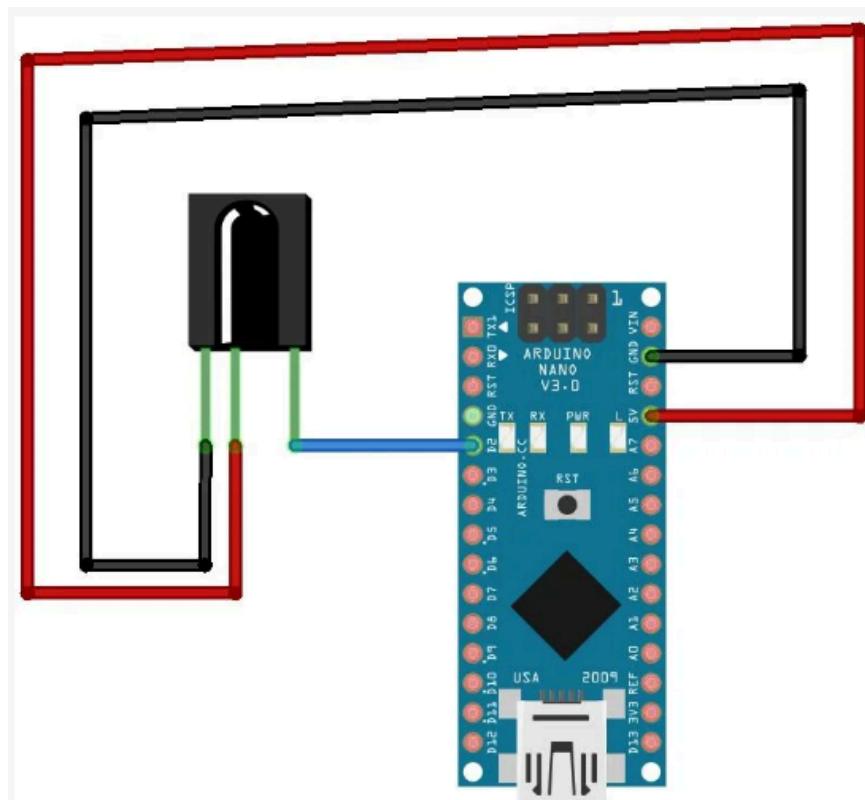
Es el encargado de encender/apagar el aire acondicionado del consultorio a través de la emisión de señales infrarrojas mediante un led infrarrojo. Para ello se emplea la librería IRremote que permite recibir o enviar señales infrarrojas.

En un principio se buscó utilizar esta librería en la ESP-EDU para controlar directamente un emisor infrarrojo. Sin embargo, la librería actualmente solo se encuentra disponible para

Arduino, por ende fue necesario programar un Arduino Nano para emitir las señales infrarrojas.

Como cada dispositivo que es controlado por señales infrarrojas recibe diferentes comandos según el modelo y la marca, fue necesario implementar primero un receptor infrarrojo. Con dicho receptor se lograron codificar las señales de encendido y apagado del control remoto del aire acondicionado a utilizar.

Para implementar este decodificador se conectó un receptor infrarrojo al pin D2 del Arduino nano de la siguiente manera:



*Diagrama obtenido de:*

<https://www.instructables.com/Universal-IR-Remote-Using-Arduino-Nano/>

El código obtenido se mostró a través de la UART en el Serial Monitor para ser copiado. A continuación se adjunta link al video que comprueba el correcto funcionamiento del decodificador.

#### [Video: Funcionamiento del decodificador de señales infrarrojas.](#)

El código implementado se encuentra subido al repositorio dentro de la carpeta correspondiente al proyecto integrador como [IR\\_Decodificador.ino](#)

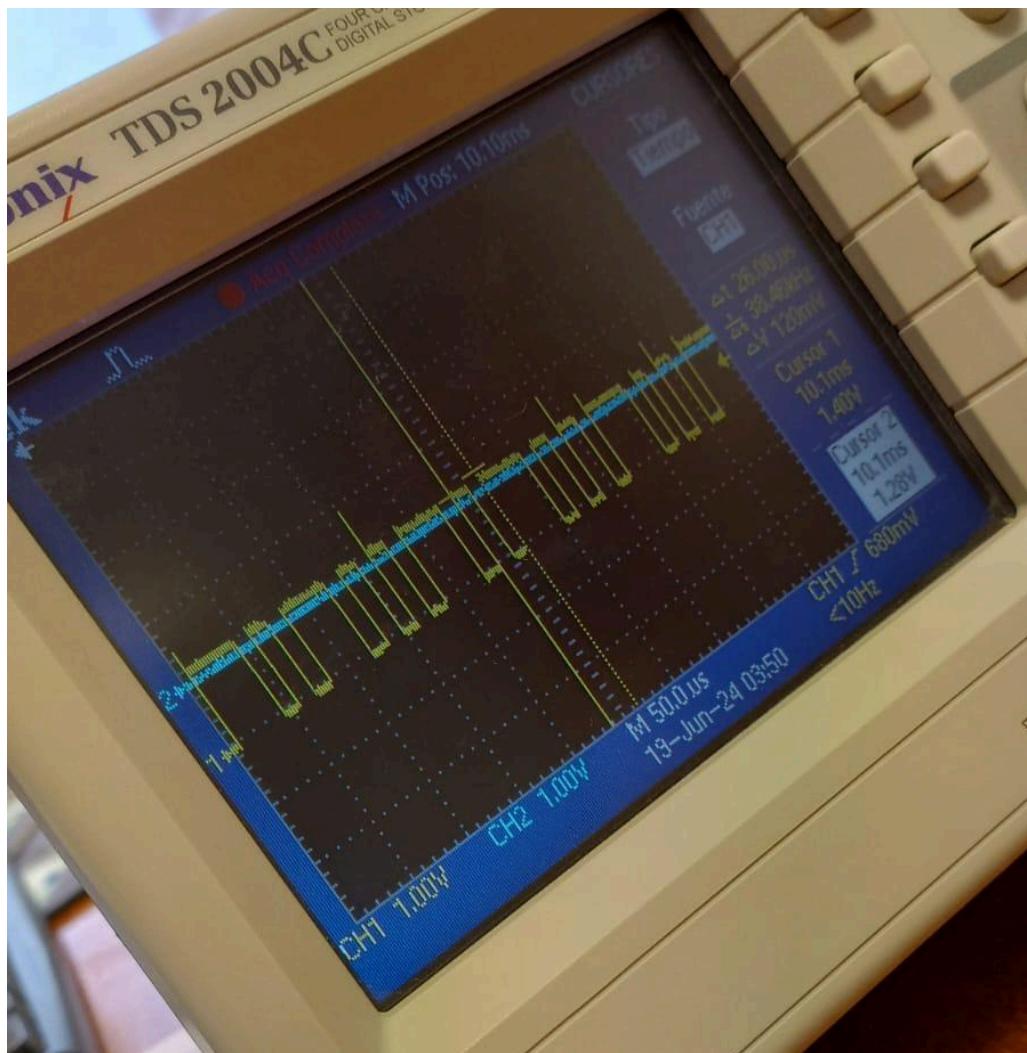
Una vez codificadas las señales de encendido y apagado enviadas por el control remoto, se implementó el Emisor Infrarrojo mediante un Led Infrarrojo conectado al pin D3 del Arduino Nano. Fue necesario utilizar un transistor NPN y una resistencia de 100ohm para polarizar el mismo.

A través del GPIO\_20 de la ESP-EDU se le envía un estado en alto al pin D7 del Arduino Nano cuando es necesario emitir el comando de Encendido y se le envía un estado en bajo cuando es necesario emitir el comando de Apagado.

Como la mayoría de los aires acondicionados reciben sus señales a 38kHz, se implementó dicha frecuencia para el emisor infrarrojo. El código implementado se encuentra subido al repositorio dentro de la carpeta correspondiente al proyecto integrador como [control\\_infrarrojo2.0](#). También se incluye video del mismo:

#### [Video: Funcionamiento del Emisor Infrarrojo](#)

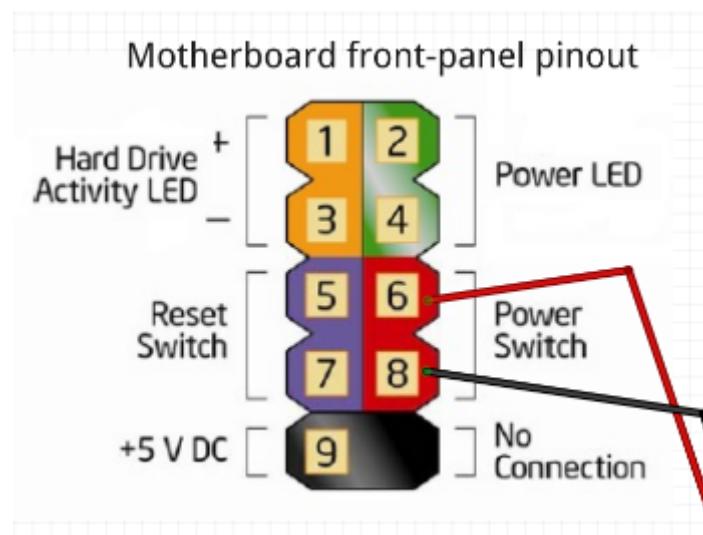
Para corroborar la correcta emisión, se observó la emisión del led a través del osciloscopio:



Como podemos observar, posee los 38kHz necesarios y envía la señal decodificada correctamente.

### Control de PC

Para controlar el encendido y apagado de la PC se propone conectar un relé al power switch de la computadora. El mismo se activa/desactiva a través del GPIO\_21.



Modelo de conexión. El relé se conectaría al Power Switch de la motherboard de la PC.

Al no contar con una PC para realizar dicha conexión, la misma se simuló mediante el encendido/apagado de una lámpara led.

### [Video: Simulación funcionamiento de Encendido/Apagado PC](#)

### Control de Lámpara UV

Al igual que para controlar la PC, la lámpara UV se enciende o apaga mediante un relé comandado por el GPIO\_22. La misma debe encenderse fuera del horario de atención.

Dicha lámpara se simuló mediante una lámpara led de color azul.

### [Video: Funcionamiento lámpara UV](#)

### Aplicación Móvil

Debido al diseño del dispositivo, la aplicación móvil requiere que se puedan ingresar:

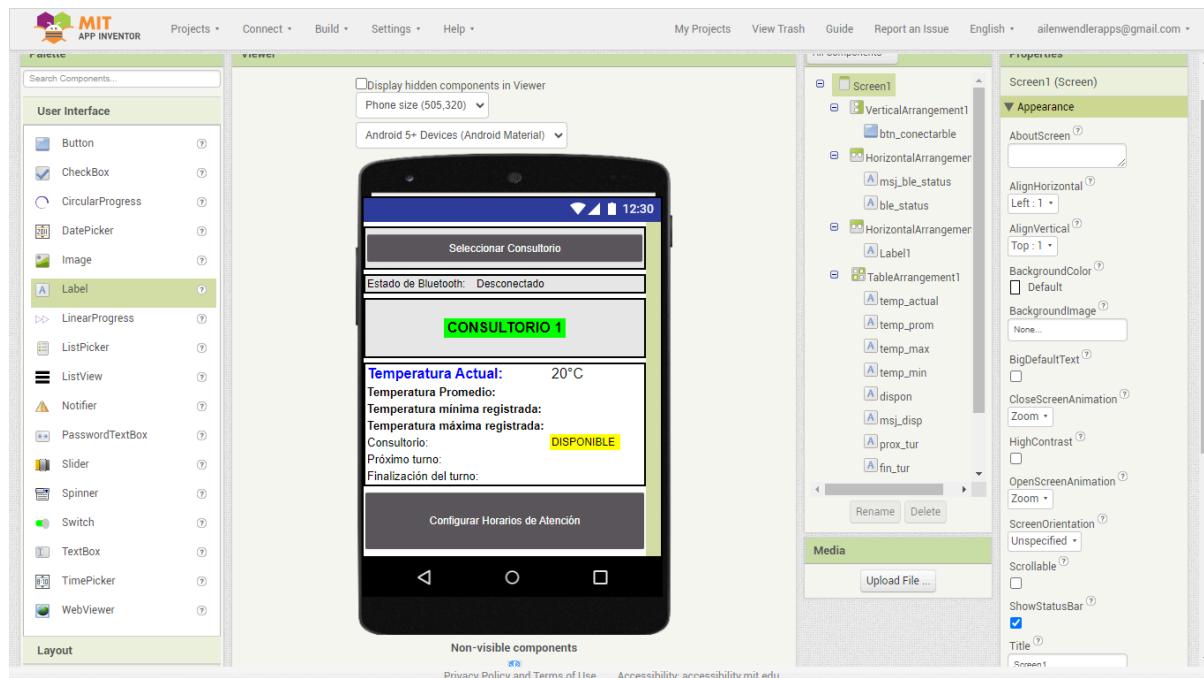
- Días de atención
- Horario de inicio de atención

- Horario de fin de atención

Y a su vez, debe permitir observar:

- Temperatura actual del consultorio
- Temperatura Promedio
- Temperatura máxima registrada
- Temperatura mínima registrada
- Disponibilidad del consultorio
- Horario de la próxima atención
- Horario de finalización de la atención

Debido a la cantidad de requerimientos que debe cumplir la aplicación, se optó por desarrollar la aplicación android mediante la plataforma [MIT App Inventor](#) que implementa la programación en bloques y brinda gran cantidad de herramientas.



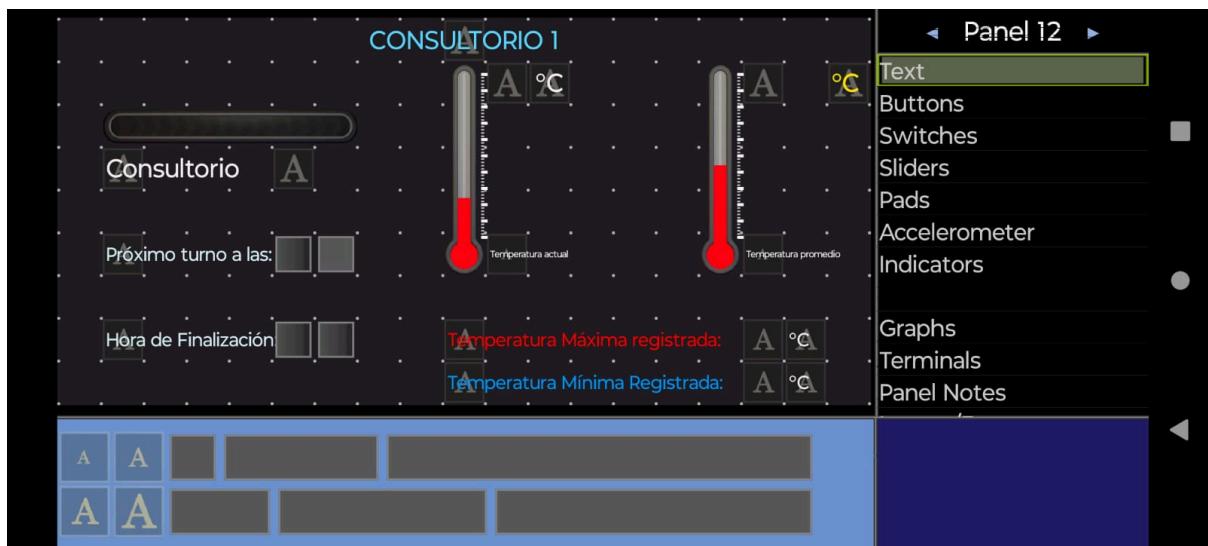
### Aplicación Android en desarrollo

Sin embargo, debido a un error de conexión con la ESP-EDU que no se pudo resolver a tiempo, se debió implementar un prototipo de la aplicación utilizando la Aplicación “Bluetooth Electronics”.

Como dicha aplicación no permite cambiar entre pantallas ni seleccionar desde listados, fue necesario crear una aplicación para cargar los horarios y días de atención y otra para observar los datos. Se asume que ya se seleccionó el Consultorio 1.



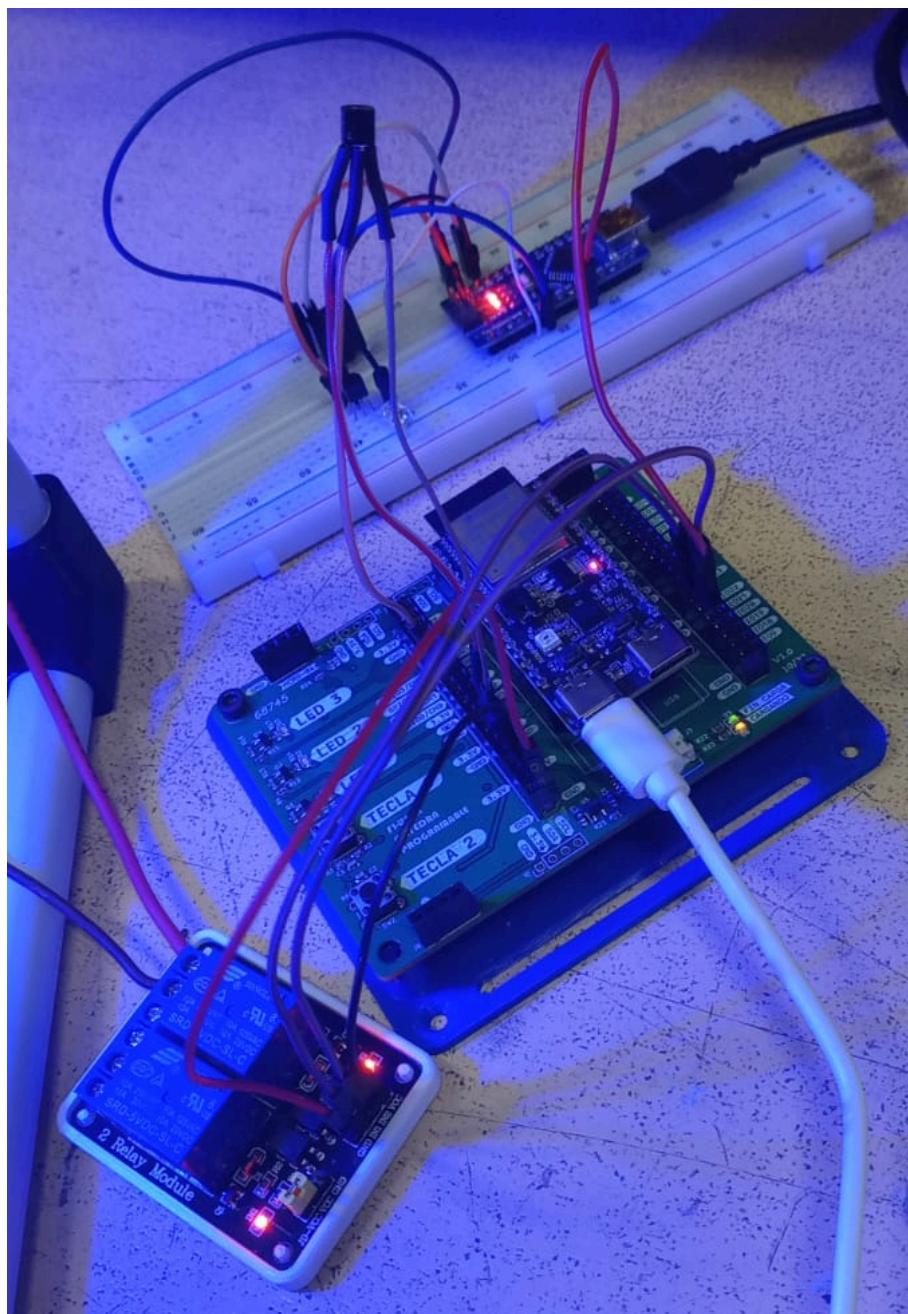
*Aplicación de configuración. En la misma deben indicarse los días de atención e ingresarse la hora de inicio y la hora de finalización.*



*Aplicación de control. En la misma se observan los datos de temperaturas, la disponibilidad del consultorio y los horarios de atención.*

## Funcionamiento

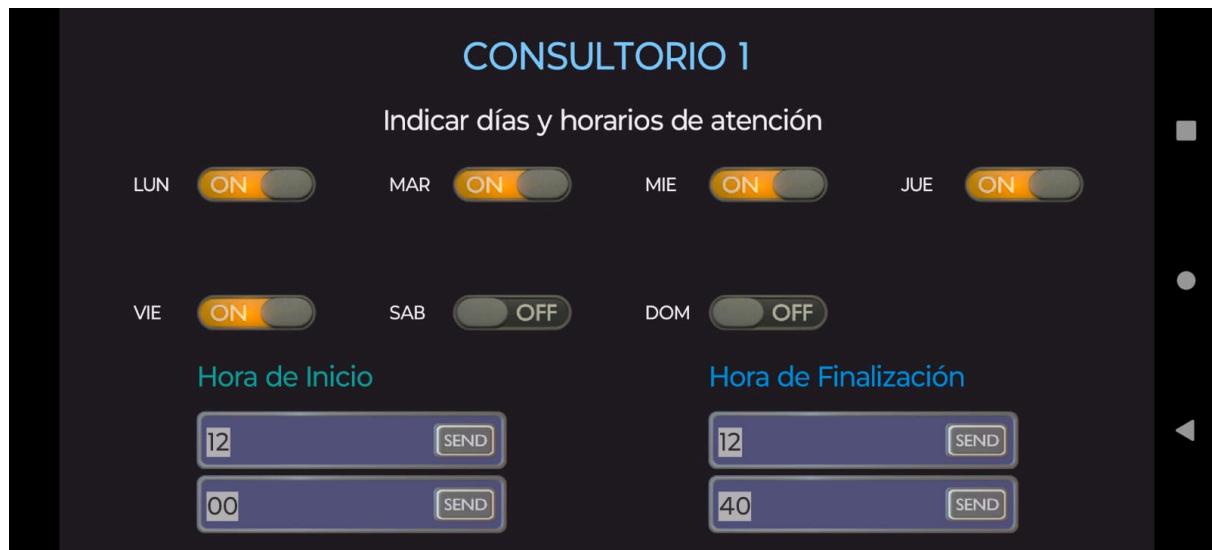
Luego de compilar el firmware y flashear en el microcontrolador, para el funcionamiento del prototipo se procede a conectar la ESP-EDU con el sensor de temperatura lm35c, el Arduino Nano ya configurado con el emisor infrarrojo y el módulo relé de 2 canales con las conexiones descritas anteriormente.



*Conexión del dispositivo.*

El Arduino Nano se alimentó mediante conexión USB con una computadora. Se conectó una sola lámpara (que representa la lámpara UV) a la salida del relé de 2 canales ya que se contaba con solamente una.

Posteriormente se procede a realizar la conexión bluetooth y se cargan los días de atención, la hora de inicio y la hora de finalización.

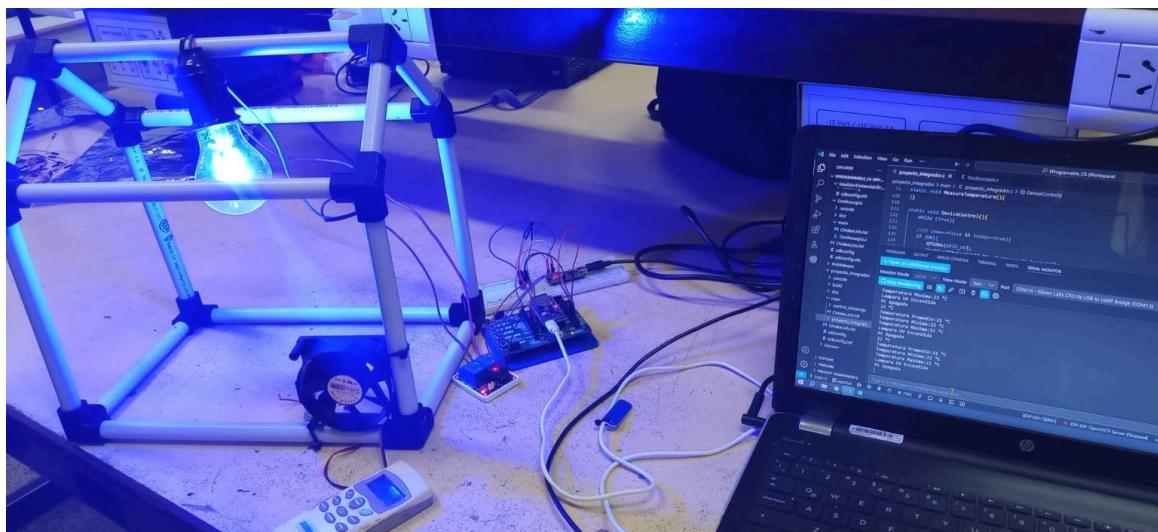


Corroboramos que los datos de temperatura se envían correctamente a la aplicación de control.

Fuera del horario de atención observamos lo siguiente:



A su vez, se enciende la lámpara UV y la PC se encuentra apagada:



Los datos son enviados por la UART al Serial Monitor para corroborarse con los datos recibidos en la app:

```

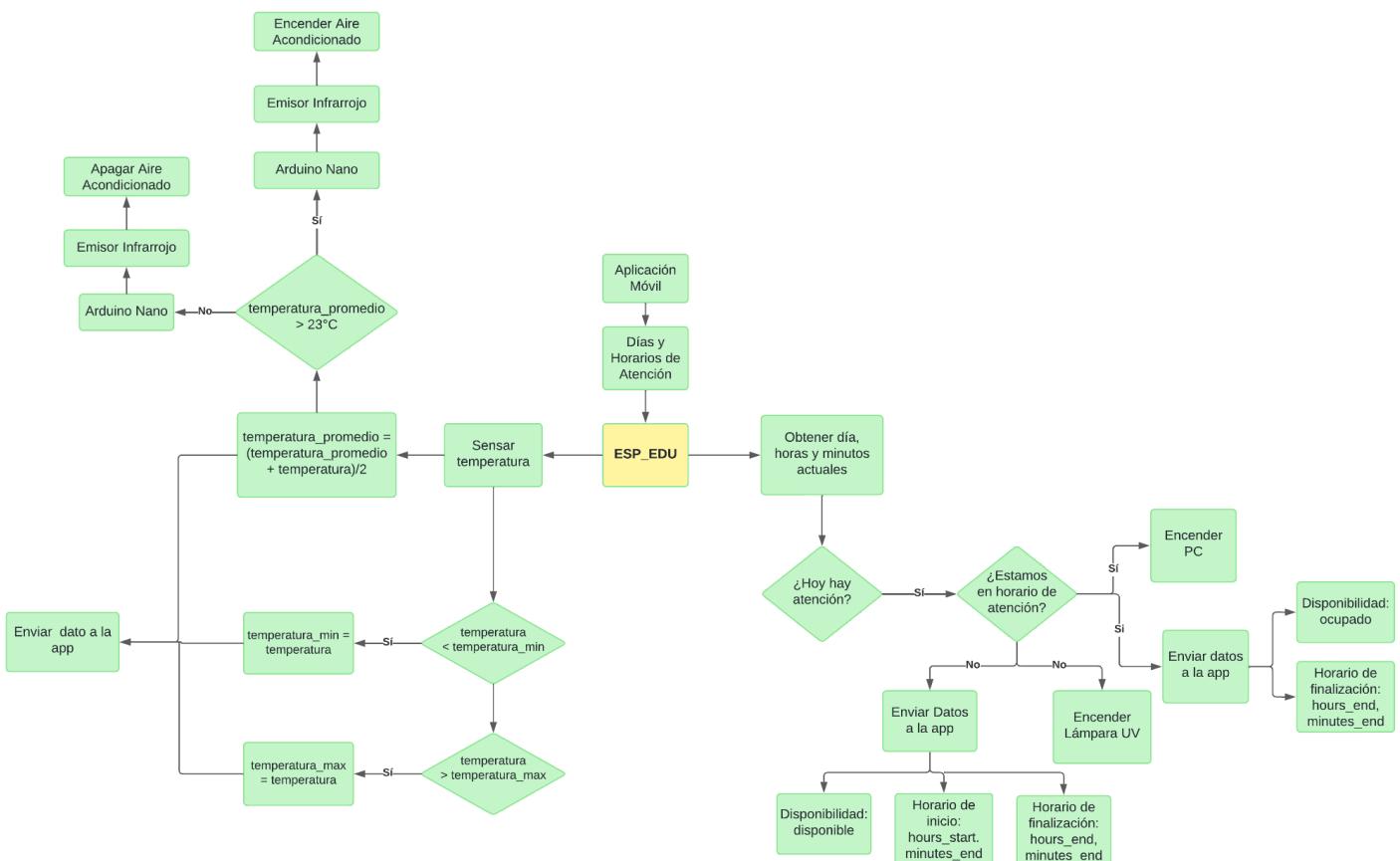
132
133     //if (now==false && today==true){
134         if (UV){
135             GPIOOn(GPIO_UV);
136         }
137     }
138
139     //if (now==true && today==false){
140         if (!UV){
141             GPIOOff(GPIO_UV);
142         }
143     }
144
145     //if (now==true && today==true){
146         if (UV){
147             GPIOOn(GPIO_UV);
148         }
149     }
150
151     //if (now==false && today==false){
152         if (!UV){
153             GPIOOff(GPIO_UV);
154         }
155     }
156
157     //if (now==true && today==true){
158         if (UV){
159             GPIOOn(GPIO_UV);
160         }
161     }
162
163     //if (now==false && today==false){
164         if (!UV){
165             GPIOOff(GPIO_UV);
166         }
167     }
168
169     //if (now==true && today==true){
170         if (UV){
171             GPIOOn(GPIO_UV);
172         }
173     }
174
175     //if (now==false && today==false){
176         if (!UV){
177             GPIOOff(GPIO_UV);
178         }
179     }
180
181     //if (now==true && today==true){
182         if (UV){
183             GPIOOn(GPIO_UV);
184         }
185     }
186
187     //if (now==false && today==false){
188         if (!UV){
189             GPIOOff(GPIO_UV);
190         }
191     }
192
193     //if (now==true && today==true){
194         if (UV){
195             GPIOOn(GPIO_UV);
196         }
197     }
198
199     //if (now==false && today==false){
200         if (!UV){
201             GPIOOff(GPIO_UV);
202         }
203     }
204
205     //if (now==true && today==true){
206         if (UV){
207             GPIOOn(GPIO_UV);
208         }
209     }
210
211     //if (now==false && today==false){
212         if (!UV){
213             GPIOOff(GPIO_UV);
214         }
215     }
216
217     //if (now==true && today==true){
218         if (UV){
219             GPIOOn(GPIO_UV);
220         }
221     }
222
223     //if (now==false && today==false){
224         if (!UV){
225             GPIOOff(GPIO_UV);
226         }
227     }
228
229     //if (now==true && today==true){
230         if (UV){
231             GPIOOn(GPIO_UV);
232         }
233     }
234
235     //if (now==false && today==false){
236         if (!UV){
237             GPIOOff(GPIO_UV);
238         }
239     }
240
241     //if (now==true && today==true){
242         if (UV){
243             GPIOOn(GPIO_UV);
244         }
245     }
246
247     //if (now==false && today==false){
248         if (!UV){
249             GPIOOff(GPIO_UV);
250         }
251     }
252
253     //if (now==true && today==true){
254         if (UV){
255             GPIOOn(GPIO_UV);
256         }
257     }
258
259     //if (now==false && today==false){
260         if (!UV){
261             GPIOOff(GPIO_UV);
262         }
263     }
264
265     //if (now==true && today==true){
266         if (UV){
267             GPIOOn(GPIO_UV);
268         }
269     }
270
271     //if (now==false && today==false){
272         if (!UV){
273             GPIOOff(GPIO_UV);
274         }
275     }
276
277     //if (now==true && today==true){
278         if (UV){
279             GPIOOn(GPIO_UV);
280         }
281     }
282
283     //if (now==false && today==false){
284         if (!UV){
285             GPIOOff(GPIO_UV);
286         }
287     }
288
289     //if (now==true && today==true){
290         if (UV){
291             GPIOOn(GPIO_UV);
292         }
293     }
294
295     //if (now==false && today==false){
296         if (!UV){
297             GPIOOff(GPIO_UV);
298         }
299     }
299
300     //if (now==true && today==true){
301         if (UV){
302             GPIOOn(GPIO_UV);
303         }
304     }
305
306     //if (now==false && today==false){
307         if (!UV){
308             GPIOOff(GPIO_UV);
309         }
310     }
311
312     //if (now==true && today==true){
313         if (UV){
314             GPIOOn(GPIO_UV);
315         }
316     }
317
318     //if (now==false && today==false){
319         if (!UV){
320             GPIOOff(GPIO_UV);
321         }
322     }
323
324     //if (now==true && today==true){
325         if (UV){
326             GPIOOn(GPIO_UV);
327         }
328     }
329
330     //if (now==false && today==false){
331         if (!UV){
332             GPIOOff(GPIO_UV);
333         }
334     }
335
336     //if (now==true && today==true){
337         if (UV){
338             GPIOOn(GPIO_UV);
339         }
340     }
341
342     //if (now==false && today==false){
343         if (!UV){
344             GPIOOff(GPIO_UV);
345         }
346     }
347
348     //if (now==true && today==true){
349         if (UV){
350             GPIOOn(GPIO_UV);
351         }
352     }
353
354     //if (now==false && today==false){
355         if (!UV){
356             GPIOOff(GPIO_UV);
357         }
358     }
359
360     //if (now==true && today==true){
361         if (UV){
362             GPIOOn(GPIO_UV);
363         }
364     }
365
366     //if (now==false && today==false){
367         if (!UV){
368             GPIOOff(GPIO_UV);
369         }
370     }
371
372     //if (now==true && today==true){
373         if (UV){
374             GPIOOn(GPIO_UV);
375         }
376     }
377
378     //if (now==false && today==false){
379         if (!UV){
380             GPIOOff(GPIO_UV);
381         }
382     }
383
384     //if (now==true && today==true){
385         if (UV){
386             GPIOOn(GPIO_UV);
387         }
388     }
389
390     //if (now==false && today==false){
391         if (!UV){
392             GPIOOff(GPIO_UV);
393         }
394     }
395
396     //if (now==true && today==true){
397         if (UV){
398             GPIOOn(GPIO_UV);
399         }
400     }
401
402     //if (now==false && today==false){
403         if (!UV){
404             GPIOOff(GPIO_UV);
405         }
406     }
407
408     //if (now==true && today==true){
409         if (UV){
410             GPIOOn(GPIO_UV);
411         }
412     }
413
414     //if (now==false && today==false){
415         if (!UV){
416             GPIOOff(GPIO_UV);
417         }
418     }
419
420     //if (now==true && today==true){
421         if (UV){
422             GPIOOn(GPIO_UV);
423         }
424     }
425
426     //if (now==false && today==false){
427         if (!UV){
428             GPIOOff(GPIO_UV);
429         }
430     }
431
432     //if (now==true && today==true){
433         if (UV){
434             GPIOOn(GPIO_UV);
435         }
436     }
437
438     //if (now==false && today==false){
439         if (!UV){
440             GPIOOff(GPIO_UV);
441         }
442     }
443
444     //if (now==true && today==true){
445         if (UV){
446             GPIOOn(GPIO_UV);
447         }
448     }
449
450     //if (now==false && today==false){
451         if (!UV){
452             GPIOOff(GPIO_UV);
453         }
454     }
455
456     //if (now==true && today==true){
457         if (UV){
458             GPIOOn(GPIO_UV);
459         }
460     }
461
462     //if (now==false && today==false){
463         if (!UV){
464             GPIOOff(GPIO_UV);
465         }
466     }
467
468     //if (now==true && today==true){
469         if (UV){
470             GPIOOn(GPIO_UV);
471         }
472     }
473
474     //if (now==false && today==false){
475         if (!UV){
476             GPIOOff(GPIO_UV);
477         }
478     }
479
480     //if (now==true && today==true){
481         if (UV){
482             GPIOOn(GPIO_UV);
483         }
484     }
485
486     //if (now==false && today==false){
487         if (!UV){
488             GPIOOff(GPIO_UV);
489         }
490     }
491
492     //if (now==true && today==true){
493         if (UV){
494             GPIOOn(GPIO_UV);
495         }
496     }
497
498     //if (now==false && today==false){
499         if (!UV){
500             GPIOOff(GPIO_UV);
501         }
502     }
503
504     //if (now==true && today==true){
505         if (UV){
506             GPIOOn(GPIO_UV);
507         }
508     }
509
510     //if (now==false && today==false){
511         if (!UV){
512             GPIOOff(GPIO_UV);
513         }
514     }
515
516     //if (now==true && today==true){
517         if (UV){
518             GPIOOn(GPIO_UV);
519         }
520     }
521
522     //if (now==false && today==false){
523         if (!UV){
524             GPIOOff(GPIO_UV);
525         }
526     }
527
528     //if (now==true && today==true){
529         if (UV){
530             GPIOOn(GPIO_UV);
531         }
532     }
533
534     //if (now==false && today==false){
535         if (!UV){
536             GPIOOff(GPIO_UV);
537         }
538     }
539
540     //if (now==true && today==true){
541         if (UV){
542             GPIOOn(GPIO_UV);
543         }
544     }
545
546     //if (now==false && today==false){
547         if (!UV){
548             GPIOOff(GPIO_UV);
549         }
550     }
551
552     //if (now==true && today==true){
553         if (UV){
554             GPIOOn(GPIO_UV);
555         }
556     }
557
558     //if (now==false && today==false){
559         if (!UV){
560             GPIOOff(GPIO_UV);
561         }
562     }
563
564     //if (now==true && today==true){
565         if (UV){
566             GPIOOn(GPIO_UV);
567         }
568     }
569
570     //if (now==false && today==false){
571         if (!UV){
572             GPIOOff(GPIO_UV);
573         }
574     }
575
576     //if (now==true && today==true){
577         if (UV){
578             GPIOOn(GPIO_UV);
579         }
580     }
581
582     //if (now==false && today==false){
583         if (!UV){
584             GPIOOff(GPIO_UV);
585         }
586     }
587
588     //if (now==true && today==true){
589         if (UV){
590             GPIOOn(GPIO_UV);
591         }
592     }
593
594     //if (now==false && today==false){
595         if (!UV){
596             GPIOOff(GPIO_UV);
597         }
598     }
599
599
600     //if (now==true && today==true){
601         if (UV){
602             GPIOOn(GPIO_UV);
603         }
604     }
605
606     //if (now==false && today==false){
607         if (!UV){
608             GPIOOff(GPIO_UV);
609         }
610     }
611
612     //if (now==true && today==true){
613         if (UV){
614             GPIOOn(GPIO_UV);
615         }
616     }
617
618     //if (now==false && today==false){
619         if (!UV){
620             GPIOOff(GPIO_UV);
621         }
622     }
623
624     //if (now==true && today==true){
625         if (UV){
626             GPIOOn(GPIO_UV);
627         }
628     }
629
630     //if (now==false && today==false){
631         if (!UV){
632             GPIOOff(GPIO_UV);
633         }
634     }
635
636     //if (now==true && today==true){
637         if (UV){
638             GPIOOn(GPIO_UV);
639         }
640     }
641
642     //if (now==false && today==false){
643         if (!UV){
644             GPIOOff(GPIO_UV);
645         }
646     }
647
648     //if (now==true && today==true){
649         if (UV){
650             GPIOOn(GPIO_UV);
651         }
652     }
653
654     //if (now==false && today==false){
655         if (!UV){
656             GPIOOff(GPIO_UV);
657         }
658     }
659
660     //if (now==true && today==true){
661         if (UV){
662             GPIOOn(GPIO_UV);
663         }
664     }
665
666     //if (now==false && today==false){
667         if (!UV){
668             GPIOOff(GPIO_UV);
669         }
670     }
671
672     //if (now==true && today==true){
673         if (UV){
674             GPIOOn(GPIO_UV);
675         }
676     }
677
678     //if (now==false && today==false){
679         if (!UV){
680             GPIOOff(GPIO_UV);
681         }
682     }
683
684     //if (now==true && today==true){
685         if (UV){
686             GPIOOn(GPIO_UV);
687         }
688     }
689
690     //if (now==false && today==false){
691         if (!UV){
692             GPIOOff(GPIO_UV);
693         }
694     }
695
696     //if (now==true && today==true){
697         if (UV){
698             GPIOOn(GPIO_UV);
699         }
700     }
701
702     //if (now==false && today==false){
703         if (!UV){
704             GPIOOff(GPIO_UV);
705         }
706     }
707
708     //if (now==true && today==true){
709         if (UV){
710             GPIOOn(GPIO_UV);
711         }
712     }
713
714     //if (now==false && today==false){
715         if (!UV){
716             GPIOOff(GPIO_UV);
717         }
718     }
719
720     //if (now==true && today==true){
721         if (UV){
722             GPIOOn(GPIO_UV);
723         }
724     }
725
726     //if (now==false && today==false){
727         if (!UV){
728             GPIOOff(GPIO_UV);
729         }
730     }
731
732     //if (now==true && today==true){
733         if (UV){
734             GPIOOn(GPIO_UV);
735         }
736     }
737
738     //if (now==false && today==false){
739         if (!UV){
740             GPIOOff(GPIO_UV);
741         }
742     }
743
744     //if (now==true && today==true){
745         if (UV){
746             GPIOOn(GPIO_UV);
747         }
748     }
749
750     //if (now==false && today==false){
751         if (!UV){
752             GPIOOff(GPIO_UV);
753         }
754     }
755
756     //if (now==true && today==true){
757         if (UV){
758             GPIOOn(GPIO_UV);
759         }
760     }
761
762     //if (now==false && today==false){
763         if (!UV){
764             GPIOOff(GPIO_UV);
765         }
766     }
767
768     //if (now==true && today==true){
769         if (UV){
770             GPIOOn(GPIO_UV);
771         }
772     }
773
774     //if (now==false && today==false){
775         if (!UV){
776             GPIOOff(GPIO_UV);
777         }
778     }
779
780     //if (now==true && today==true){
781         if (UV){
782             GPIOOn(GPIO_UV);
783         }
784     }
785
786     //if (now==false && today==false){
787         if (!UV){
788             GPIOOff(GPIO_UV);
789         }
790     }
791
792     //if (now==true && today==true){
793         if (UV){
794             GPIOOn(GPIO_UV);
795         }
796     }
797
798     //if (now==false && today==false){
799         if (!UV){
800             GPIOOff(GPIO_UV);
801         }
802     }
803
804     //if (now==true && today==true){
805         if (UV){
806             GPIOOn(GPIO_UV);
807         }
808     }
809
810     //if (now==false && today==false){
811         if (!UV){
812             GPIOOff(GPIO_UV);
813         }
814     }
815
816     //if (now==true && today==true){
817         if (UV){
818             GPIOOn(GPIO_UV);
819         }
820     }
821
822     //if (now==false && today==false){
823         if (!UV){
824             GPIOOff(GPIO_UV);
825         }
826     }
827
828     //if (now==true && today==true){
829         if (UV){
830             GPIOOn(GPIO_UV);
831         }
832     }
833
834     //if (now==false && today==false){
835         if (!UV){
836             GPIOOff(GPIO_UV);
837         }
838     }
839
840     //if (now==true && today==true){
841         if (UV){
842             GPIOOn(GPIO_UV);
843         }
844     }
845
846     //if (now==false && today==false){
847         if (!UV){
848             GPIOOff(GPIO_UV);
849         }
850     }
851
852     //if (now==true && today==true){
853         if (UV){
854             GPIOOn(GPIO_UV);
855         }
856     }
857
858     //if (now==false && today==false){
859         if (!UV){
860             GPIOOff(GPIO_UV);
861         }
862     }
863
864     //if (now==true && today==true){
865         if (UV){
866             GPIOOn(GPIO_UV);
867         }
868     }
869
870     //if (now==false && today==false){
871         if (!UV){
872             GPIOOff(GPIO_UV);
873         }
874     }
875
876     //if (now==true && today==true){
877         if (UV){
878             GPIOOn(GPIO_UV);
879         }
880     }
881
882     //if (now==false && today==false){
883         if (!UV){
884             GPIOOff(GPIO_UV);
885         }
886     }
887
888     //if (now==true && today==true){
889         if (UV){
890             GPIOOn(GPIO_UV);
891         }
892     }
893
894     //if (now==false && today==false){
895         if (!UV){
896             GPIOOff(GPIO_UV);
897         }
898     }
899
900     //if (now==true && today==true){
901         if (UV){
902             GPIOOn(GPIO_UV);
903         }
904     }
905
906     //if (now==false && today==false){
907         if (!UV){
908             GPIOOff(GPIO_UV);
909         }
910     }
911
912     //if (now==true && today==true){
913         if (UV){
914             GPIOOn(GPIO_UV);
915         }
916     }
917
918     //if (now==false && today==false){
919         if (!UV){
920             GPIOOff(GPIO_UV);
921         }
922     }
923
924     //if (now==true && today==true){
925         if (UV){
926             GPIOOn(GPIO_UV);
927         }
928     }
929
930     //if (now==false && today==false){
931         if (!UV){
932             GPIOOff(GPIO_UV);
933         }
934     }
935
936     //if (now==true && today==true){
937         if (UV){
938             GPIOOn(GPIO_UV);
939         }
940     }
941
942     //if (now==false && today==false){
943         if (!UV){
944             GPIOOff(GPIO_UV);
945         }
946     }
947
948     //if (now==true && today==true){
949         if (UV){
950             GPIOOn(GPIO_UV);
951         }
952     }
953
954     //if (now==false && today==false){
955         if (!UV){
956             GPIOOff(GPIO_UV);
957         }
958     }
959
960     //if (now==true && today==true){
961         if (UV){
962             GPIOOn(GPIO_UV);
963         }
964     }
965
966     //if (now==false && today==false){
967         if (!UV){
968             GPIOOff(GPIO_UV);
969         }
970     }
971
972     //if (now==true && today==true){
973         if (UV){
974             GPIOOn(GPIO_UV);
975         }
976     }
977
978     //if (now==false && today==false){
979         if (!UV){
980             GPIOOff(GPIO_UV);
981         }
982     }
983
984     //if (now==true && today==true){
985         if (UV){
986             GPIOOn(GPIO_UV);
987         }
988     }
989
990     //if (now==false && today==false){
991         if (!UV){
992             GPIOOff(GPIO_UV);
993         }
994     }
995
996     //if (now==true && today==true){
997         if (UV){
998             GPIOOn(GPIO_UV);
999         }
1000
1001     //if (now==false && today==false){
1002         if (!UV){
1003             GPIOOff(GPIO_UV);
1004         }
1005     }
1006
1007     //if (now==true && today==true){
1008         if (UV){
1009             GPIOOn(GPIO_UV);
1010         }
1011     }
1012
1013     //if (now==false && today==false){
1014         if (!UV){
1015             GPIOOff(GPIO_UV);
1016         }
1017     }
1018
1019     //if (now==true && today==true){
1020         if (UV){
1021             GPIOOn(GPIO_UV);
1022         }
1023     }
1024
1025     //if (now==false && today==false){
1026         if (!UV){
1027             GPIOOff(GPIO_UV);
1028         }
1029     }
1030
1031     //if (now==true && today==true){
1032         if (UV){
1033             GPIOOn(GPIO_UV);
1034         }
1035     }
1036
1037     //if (now==false && today==false){
1038         if (!UV){
1039             GPIOOff(GPIO_UV);
1040         }
1041     }
1042
1043     //if (now==true && today==true){
1044         if (UV){
1045             GPIOOn(GPIO_UV);
1046         }
1047     }
1048
1049     //if (now==false && today==false){
1050         if (!UV){
1051             GPIOOff(GPIO_UV);
1052         }
1053     }
1054
1055     //if (now==true && today==true){
1056         if (UV){
1057             GPIOOn(GPIO_UV);
1058         }
1059     }
1060
1061     //if (now==false && today==false){
1062         if (!UV){
1063             GPIOOff(GPIO_UV);
1064         }
1065     }
1066
1067     //if (now==true && today==true){
1068         if (UV){
1069             GPIOOn(GPIO_UV);
1070         }
1071     }
1072
1073     //if (now==false && today==false){
1074         if (!UV){
1075             GPIOOff(GPIO_UV);
1076         }
1077     }
1078
1079     //if (now==true && today==true){
1080         if (UV){
1081             GPIOOn(GPIO_UV);
1082         }
1083     }
1084
1085     //if (now==false && today==false){
1086         if (!UV){
1087             GPIOOff(GPIO_UV);
1088         }
1089     }
1090
1091     //if (now==true && today==true){
1092         if (UV){
1093             GPIOOn(GPIO_UV);
1094         }
1095     }
1096
1097     //if (now==false && today==false){
1098         if (!UV){
1099             GPIOOff(GPIO_UV);
1100         }
1101     }
1102
1103     //if (now==true && today==true){
1104         if (UV){
1105             GPIOOn(GPIO_UV);
1106         }
1107     }
1108
1109     //if (now==false && today==false){
1110         if (!UV){
1111             GPIOOff(GPIO_UV);
1112         }
1113     }
1114
1115     //if (now==true && today==true){
1116         if (UV){
1117             GPIOOn(GPIO_UV);
1118         }
1119     }
1120
1121     //if (now==false && today==false){
1122         if (!UV){
1123             GPIOOff(GPIO_UV);
1124         }
1125     }
1126
1127     //if (now==true && today==true){
1128         if (UV){
1129             GPIOOn(GPIO_UV);
1130         }
1131     }
1132
1133     //if (now==false && today==false){
1134         if (!UV){
1135             GPIOOff(GPIO_UV);
1136         }
1137     }
1138
1139     //if (now==true && today==true){
1140         if (UV){
1141             GPIOOn(GPIO_UV);
1142         }
1143     }
1144
1145     //if (now==false && today==false){
1146         if (!UV){
1147             GPIOOff(GPIO_UV);
1148         }
1149     }
1150
1151     //if (now==true && today==true){
1152         if (UV){
1153             GPIOOn(GPIO_UV);
1154         }
1155     }
1156
1157     //if (now==false && today==false){
1158         if (!UV){
1159             GPIOOff(GPIO_UV);
1160         }
1161     }
1162
1163     //if (now==true && today==true){
1164         if (UV){
1165             GPIOOn(GPIO_UV);
1166         }
1167     }
1168
1169     //if (now==false && today==false){
1170         if (!UV){
1171             GPIOOff(GPIO_UV);
1172         }
1173     }
1174
1175     //if (now==true && today==true){
1176         if (UV){
1177             GPIOOn(GPIO_UV);
1178         }
1179     }
1180
1181     //if (now==false && today==false){
1182         if (!UV){
1183             GPIOOff(GPIO_UV);
1184         }
1185     }
1186
1187     //if (now==true && today==true){
1188         if (UV){
1189             GPIOOn(GPIO_UV);
1190         }
1191     }
1192
1193     //if (now==false && today==false){
1194         if (!UV){
1195             GPIOOff(GPIO_UV);
1196         }
1197     }
1198
1199     //if (now==true && today==true){
1200         if (UV){
1201             GPIOOn(GPIO_UV);
1202         }
1203     }
1204
1205     //if (now==false && today==false){
1206         if (!UV){
1207             GPIOOff(GPIO_UV);
1208         }
1209     }
1210
1211     //if (now==true && today==true){
1212         if (UV){
1213             GPIOOn(GPIO_UV);
1214         }
1215     }
1216
1217     //if (now==false && today==false){
1218         if (!UV){
1219             GPIOOff(GPIO_UV);
1220         }
1221     }
1222
1223     //if (now==true && today==true){
1224         if (UV){
1225             GPIOOn(GPIO_UV);
1226         }
1227     }
1228
1229     //if (now==false && today==false){
1230         if (!UV){
1231             GPIOOff(GPIO_UV);
1232         }
1233     }
1234
1235     //if (now==true && today==true){
1236         if (UV){
1237             GPIOOn(GPIO_UV);
1238         }
1239     }
1240
1241     //if (now==false && today==false){
1242         if (!UV){
1243             GPIOOff(GPIO_UV);
1244         }
1245     }
1246
1247     //if (now==true && today==true){
1248         if (UV){
1249             GPIOOn(GPIO_UV);
1250         }
1251     }
1252
1253     //if (now==false && today==false){
1254         if (!UV){
1255             GPIOOff(GPIO_UV);
1256         }
1257     }
1258
1259     //if (now==true && today==true){
1260         if (UV){
1261             GPIOOn(GPIO_UV);
1262         }
1263     }
1264
1265     //if (now==false && today==false){
1266         if (!UV){
1267             GPIOOff(GPIO_UV);
1268         }
1269     }
1270
1271     //if (now==true && today==true){
1272         if (UV){
1273             GPIOOn(GPIO_UV);
1274         }
1275     }
1276
1277     //if (now==false && today==false){
1278         if (!UV){
1279             GPIOOff(GPIO_UV);
1280         }
1281     }
1282
1283     //if (now==true && today==true){
1284         if (UV){
1285             GPIOOn(GPIO_UV);
1286         }
1287     }
1288
1289     //if (now==false && today==false){
1290         if (!UV){
1291             GPIOOff(GPIO_UV);
1292         }
1293     }
1294
1295     //if (now==true && today==true){
1296         if (UV){
1297             GPIOOn(GPIO_UV);
1298         }
1299     }
1300
1301     //if (now==false && today==false){
1302         if (!UV){
1303             GPIOOff(GPIO_UV);
1304         }
1305     }
1306
1307     //if (now==true && today==true){
1308         if (UV){
1309             GPIOOn(GPIO_UV);
1310         }
1311     }
1312
1313     //if (now==false && today==false){
1314         if (!UV){
1315             GPIOOff(GPIO_UV);
1316         }
1317     }
1318
1319     //if (now==true && today==true){
1320         if (UV){
1321             GPIOOn(GPIO_UV);
1322         }
1323     }
1324
1325     //if (now==false && today==false){
1326         if (!UV){
1327             GPIOOff(GPIO_UV);
1328         }
1329     }
1330
1331     //if (now==true && today==true){
1332         if (UV){
1333             GPIOOn(GPIO_UV);
1334         }
1335     }
1336
1337     //if (now==false && today==false){
1338         if (!UV){
1339             GPIOOff(GPIO_UV);
1340         }
1341     }
1342
1343     //if (now==true && today==true){
1344         if (UV){
1345             GPIOOn(GPIO_UV);
1346         }
1347     }
1348
1349     //if (now==false && today==false){
1350         if (!UV){
1351             GPIOOff(GPIO_UV);
1352         }
1353     }
1354
1355     //if (now==true && today==true){
1356         if (UV){
1357             GPIOOn(GPIO_UV);
1358         }
1359     }
1360
1361     //if (now==false && today==false){
1362         if (!UV){
1363             GPIOOff(GPIO_UV);
1364         }
1365     }
1366
1367     //if (now==true && today==true){
1368         if (UV){
1369             GPIOOn(GPIO_UV);
1370         }
13
```



Se puede observar que los termómetros de la temperatura actual y de la temperatura promedio no son proporcionales. Esta variación puede deberse a que el valor de temperatura actual es de tipo entero y el valor de temperatura promedio es de tipo flotante.

### Diagrama de flujo

Una vez explicado y comprobado el funcionamiento del prototipo, se adjunta el diagrama de flujo general del mismo:



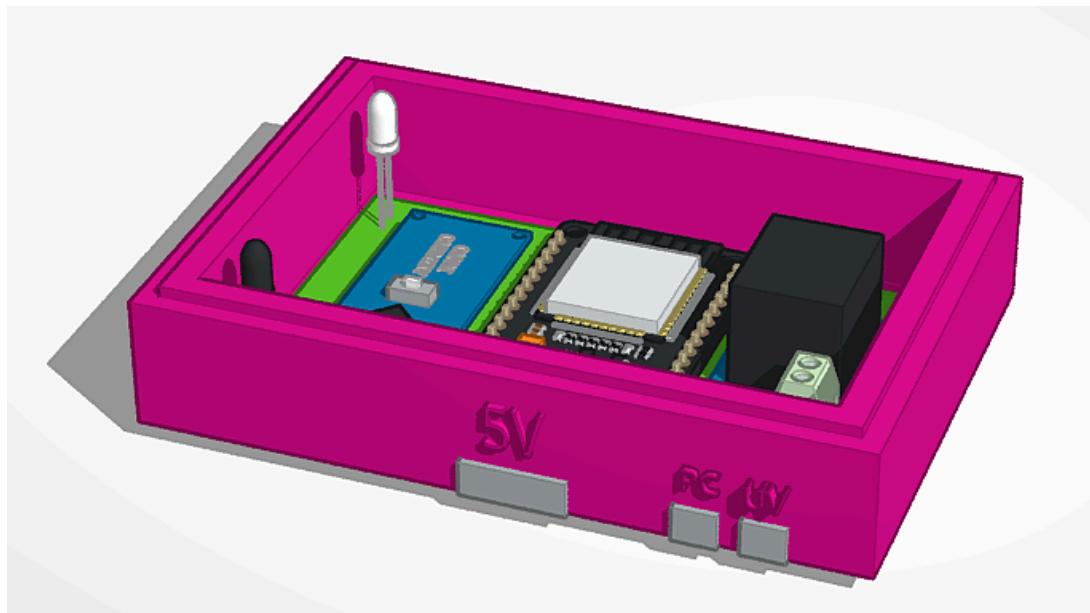
## Prototipo de diseño

Considerando una propuesta de diseño para el dispositivo, una vez optimizado y con una aplicación móvil eficiente y que cumpla con todos los requerimientos necesarios, se propone el siguiente modelo:



*Prototipo de Diseño del dispositivo ConfortMed.*

El mismo se puede obtener mediante impresión 3D con material PLA. Se observa la entrada de alimentación y las salidas para las conexiones con el PC y la Lámpara UV. También se observa el led emisor infrarrojo (en blanco) y el receptor infrarrojo (en negro).



*Vista interna. Observamos también la ESP-EDU, el Arduino Nano, el módulo relé de 2 canales.*

Se debe implementar un módulo ConfortMed en cada consultorio. Luego, mediante la aplicación, se elige el consultorio a gestionar y se le asignan los días y horarios de atención, a su vez se observa la disponibilidad de consultorios y las temperaturas de cada uno.

## **Videos**

- [Funcionamiento decodificador infrarrojo](#)
- [Funcionamiento emisor infrarrojo](#)
- [Funcionamiento control PC](#)
- [Funcionamiento control Lámpara UV](#)
- [Diseño de prototipo vista interna](#)
- [Diseño de prototipo vista externa](#)