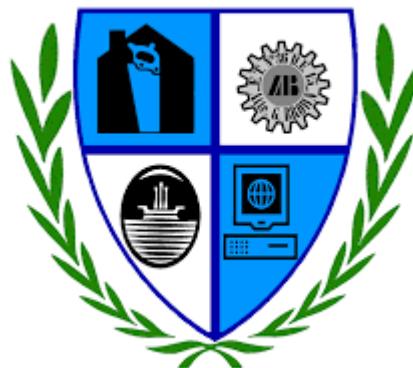


**E.T.N°36 D.E 15 “Alte. Guillermo Brown.”
Galván 3700**



CUE : 20072400

YAKU



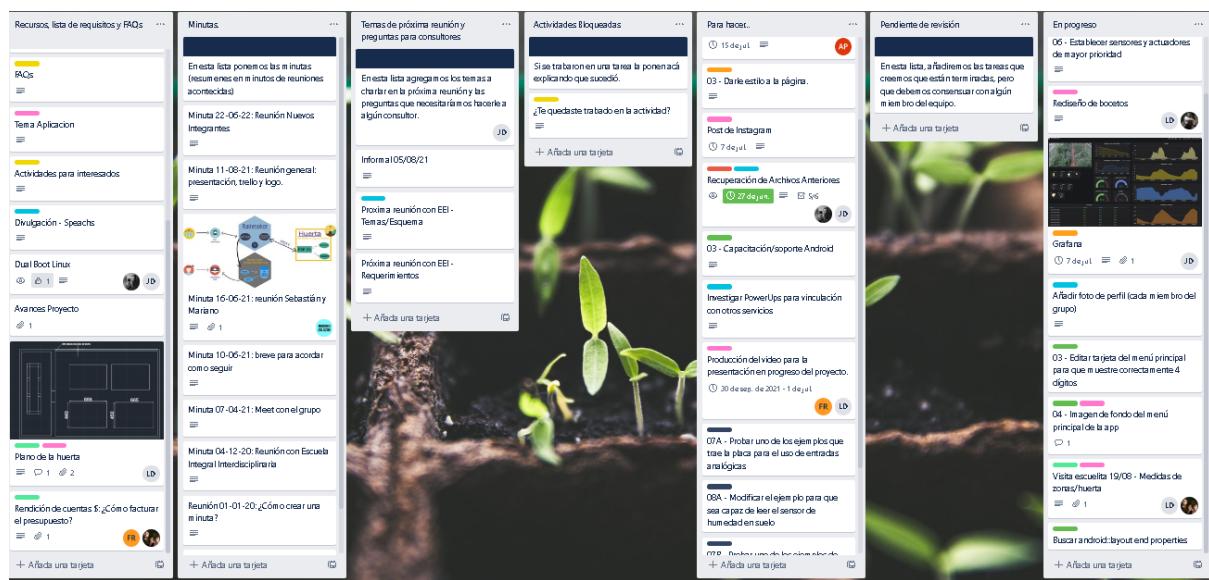
Un “ecosistema” automatizado para tu huerta.

Docente Referente: Franco Rodriguez DNI: 43819192

Alumnos: Jeremias Davison DNI: 45234487, Federico Garcia DNI: 45747650

14/09/22

La información fue recopilada del trello que tenemos para trabajar en común.



Inicio

4 de dic. de 2020

Hubo una reunión con el colegio Escuela Integral Interdisciplinaria en la que se recolectó información que le permitió empezar a darle forma al mapa de empatía. Se nos contó un poco acerca de los proyectos que la escuela estaba trabajando con anterioridad.

4 de dic. de 2020 - 22 de jun. de 2021

Se terminaron de definir herramientas/tecnologías a utilizar.

25 de mar. de 2021

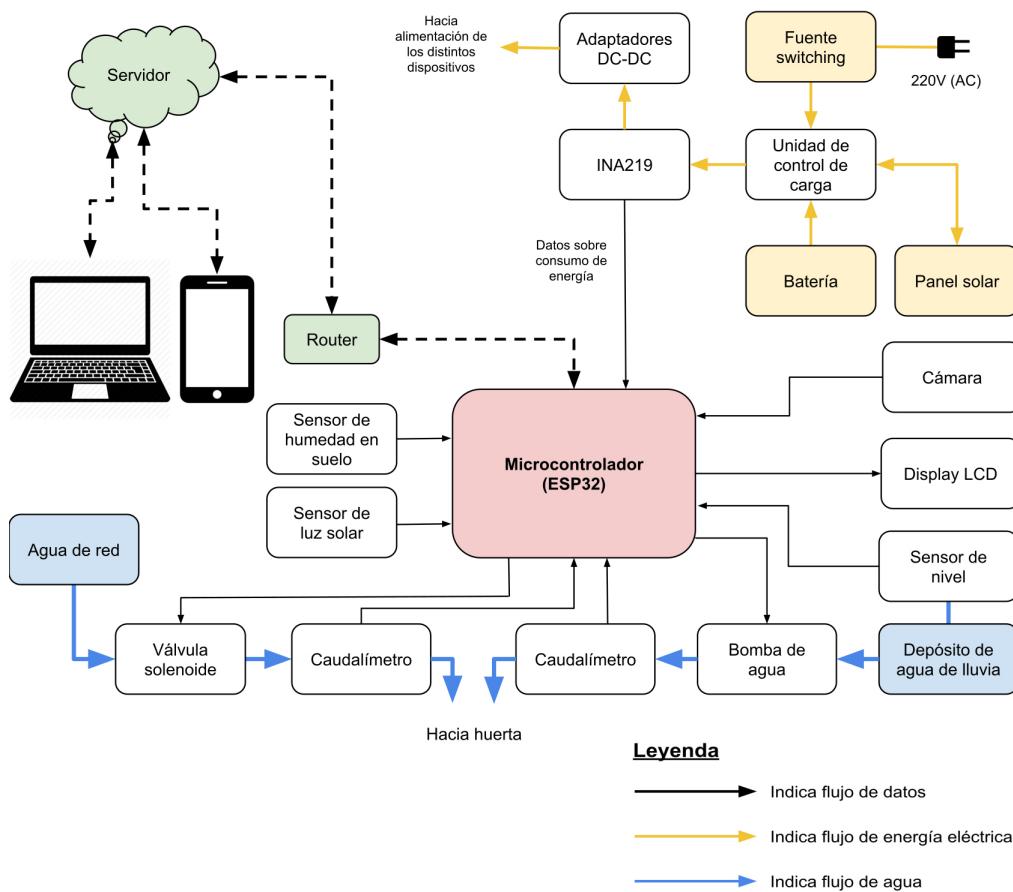
Se crearon los requisitos del proyecto. A continuación una breve captura, que muestra algunos de los requisitos de mayor prioridad.

Requisitos de mayor prioridad

- El dispositivo debe recabar los siguientes parámetros:
 - Humedad en suelo (en distintos puntos de la huerta).
 - Volumen de agua utilizado: con el fin de obtener estadísticas de los cultivos.
 - Luz solar: será utilizado para saber cuánto tiempo de luz tienen los cultivos y paneles fotovoltaicos. Además de permitir obtener otro dato para establecer cuándo es conveniente regar o no, con el objetivo de evitar el efecto lupa.
 - Temperatura y humedad ambiente.
- El dispositivo debe ser capaz de reportar los datos mencionados a la nube para ser accedidos en forma remota.
- Almacenar y utilizar energía solar para disminuir el consumo de la red eléctrica.
- Almacenar agua de lluvia, y utilizar agua de la red únicamente cuando se agote la primera.
- La página web/aplicación debe permitir:
 - Establecer las condiciones que habilitan/deshabilitan el riego.
 - Habilitar/deshabilitar el riego en forma manual.
 - Consultar los valores históricos de los datos recabados.

26 de mar. de 2021

Se planificó un diagrama de bloques con la estructura del proyecto.



6 de abr. de 2021

Se realizó un FODA sobre el proyecto.

YAKU	
Matriz FODA	
FACTORES INTERNOS Fortalezas y Debilidades Propios de cada equipo en relación al proyecto	FACTORES EXTERNOS Oportunidades y Amenazas Relacionadas al contexto que rodea al equipo y en donde se desarrollará el proyecto
FORTALEZAS (+) <ul style="list-style-type: none"> 1 Los integrantes del equipo tenemos conocimientos y áreas de interés que se complementan 2 Ya tenemos un prototipo andando con funcionalidades básicas 3 En la escuela contamos con disponibilidad de varios recursos 4 Nuestro equipo de trabajo es muy capaz y tiene muchas ganas de trabajar 5 Buena comunicación entre los integrantes del equipo 6 Parte de los integrantes ya trabajó en las primeras versiones del proyecto 7 Los integrantes del equipo tenemos conocimientos y áreas de interés que se complementan 8 Algunos miembros del equipo ya participamos en exposiciones 9 10 	OPORTUNIDADES (+) <ul style="list-style-type: none"> 1 Los profesores del polo educativo están dispuestos a colaborar 2 El proyecto permite el ahorro de energía gracias a la utilización de paneles solares 3 El desarrollo que haremos permitirá expandir el proyecto fácilmente a huertas de mayor tamaño 4 El proyecto es de gran ayuda a nuestra comunidad 5 En la escuela contamos con varios profesores dispuestos a ayudarnos 6 Es un sistema que podría tener bastante solicitado por la sociedad 7 El proyecto permite el ahorro de agua debido a que utiliza agua de lluvia. 8 El Polo Educativo Saavedra posee muchos espacios verdes 9 10
DEBILIDADES (-) <ul style="list-style-type: none"> 1 Al ser estudiantes de distintos años resulta más difícil encontrar espacios de trabajo comunes 2 Varios integrantes del grupo trabajan, por lo que puede ser difícil colaborar con el equipo 3 Son varias las funcionalidades que debemos agregar al proyecto 4 Falta de capacidades técnicas en algunas áreas 5 6 7 8 9 10 	AMENAZAS (-) <ul style="list-style-type: none"> 1 La carga horaria de otras materias puede perjudicar el tiempo dedicado al proyecto. 2 El costo de implementar nuestro proyecto es elevado 3 Algunos materiales del proyecto son difíciles de conseguir 4 La mitad de los integrantes del equipo egresarán este año. 5 6 7 8 9 10

7 de abr. de 2021

Se hizo una reunión en la que se determinaron factores importantes del proyecto, como ponerse en marcha de saber que placa utilizar, se comentó acerca del presupuesto a recibir, se determinaron tareas para definir ciertas cuestiones sobre la base de datos, y el front-end.

10 de jun. de 2021

Se determinó mediante una reunión que se iban a convocar a más alumnos para la realización del proyecto.

Se pusieron sobre la mesa los temas a tener en cuenta para la reunión con los nuevos integrantes.

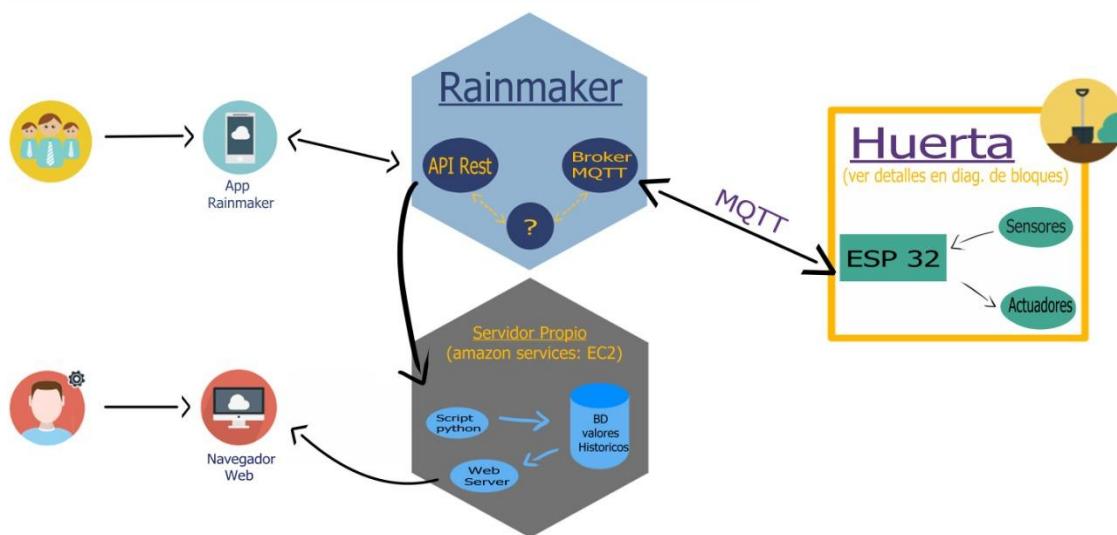
Se habló sobre cómo era preferible montar el server side.

13 de jun. de 2021

Se propuso una iniciativa para que el proyecto lleve el nombre YAKU poniéndolo como tema para una reunión.

16 de jun. de 2021

Se plasmó un boceto de cómo estaría compuesto el proyecto, y que tareas iban a tener que ser realizadas por alumnos, y cuales ya habían sido resueltas por elegir el servicio de ESP-Rainmaker.



22 de jun. de 2021

Actividades para interesados - Se invitaron a nuevos miembros al proyecto. Los grupos propuestos eran los siguientes:

- Colaborar en la organización del proyecto
- Desarrollo del firmware ESP32
- Desarrollo de BOT para Rainmaker
- Página web para mostrar métricas de la huerta
- Modificación de aplicación para celular (Android)
- Instalación y puesta en marcha de la huerta
- Diseño gráfico
- Traducción

30 de jun. de 2021

Se creó un documento interno con explicación acerca del funcionamiento del proyecto.

¿Qué es Yaku?

Es un sistema capaz de activar el riego en forma automática en función de la humedad en tierra y la cantidad de luz solar, permitiendo establecer los valores mínimos y máximos aceptables para comenzar el riego.

A su vez tiene un depósito donde recolecta agua de lluvia para hacer un uso eficiente de este recurso indispensable. Ya que en caso de que el depósito de agua de lluvia se encuentre vacío, el sistema automáticamente empezará a utilizar agua de la red.

Además cuenta con paneles solares permitiéndole reducir su consumo energético de la red.

¿Cómo funciona?

Smart watering permitirá establecer las condiciones ante las que debe regar los cultivos desde cualquier dispositivo con conexión a internet.

Con el objetivo de brindar una solución de riego integral, lo que hace inteligente a nuestro dispositivo son los distintos datos que recaba:

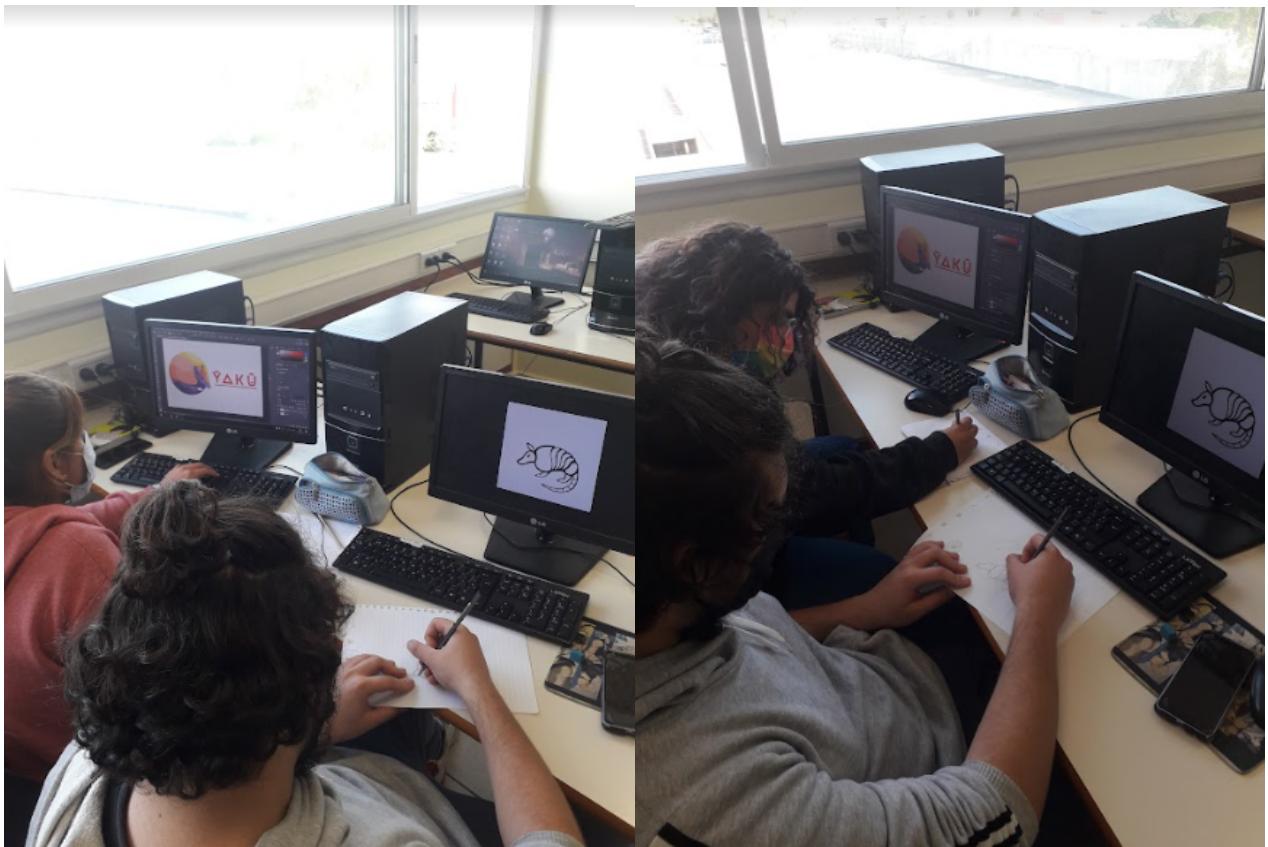
- Humedad en tierra.
- Luz solar.
- Volumen de agua utilizado.
- Humedad y temperatura ambiente.

Los datos obtenidos serán reportados a una nube permitiendo un monitoreo del estado actual del sistema y además consultar los valores históricos con el fin de obtener conclusiones sobre el cultivo.

Informe Yaku - ET 36 "Alte. Guillermo Brown"

7 de jul. de 2021 - 5 de ago. de 2021

Investigación para diseño: Se encargaron de diseñar un boceto del logo.



8 de jul. de 2021

Se establecieron los temas para una próxima reunión el 01/09, para la organización de las actividades subidas a Trello.

15 de jul. de 2021 - 6 de ago. de 2021

Pidieron información sobre el jacarandá y paletas de colores para las plataformas.

9 de ago. de 2021

Se aceptó YAKU como nombre del proyecto (antes llamado Smart Watering)

11 de ago. de 2021

Se realizó la reunión en la que se dividieron los equipos de todo el proyecto (y se pidió que vayan decidiendo los responsables), se pusieron todos al tanto y de cómo funcionaba. Se

habló del uso de Trello como medio de herramienta. Se habló de si alguien quería proponer un nombre distinto para el proyecto, y se puso en práctica la consigna de hacer un logo.

Se instaló Eclipse para C y C ++ - Firmware de ESP.

Creamos un grupo de Whatsapp para difusión de información.

Se puso a punto el tablero de Trello.

11 de ago. de 2021 - 31 de ago. de 2021

Hicimos ejercicios para el Bot de Python - Utilizamos una API, aprendimos formas de manejar un diccionario, y el manejo de JSON.

11 de ago. de 2021 - 22 de oct de 2021

Pensamos cómo va a funcionar el "help desk" de traducción para la aplicación de Android.

Dejando contacto para quienes necesiten.

11 de ago. de 2021 - 11 de sept. de 2021

Instalación de los plugins de Espressif para Eclipse.

Se subieron el programa Blinky de los ejemplos de Espressif a la ESP32, probando por primera vez su funcionamiento.

11 de ago. de 2021 - 18 de ago. de 2021

Investigaron el código realizado de la App de Android

11 de ago. de 2021 - 18 de ago. de 2021

Se realizaron las votaciones del Logo de YAKU, ya que hicimos varias opciones para encontrar la adecuada y la imagen que a todos nos agradara.

11 de ago. de 2021 - 22 de sep. de 2021

Se tradujeron los strings para App de celular, para poder tenerlos en el idioma español e inglés.

```

221      -      <string name="dialog_title_login_failed">Sign-in failed</string>
222      -      <string name="dialog_title_sign_up_failed">Sign up failed</string>
223      -      <string name="dialog_title_sign_up_success">Sign up successful!</string>
224      -      <string name="dialog_title_conf_code_sent">Confirmation code sent</string>
225      -      <string name="dialog_title_conf_code_req_fail">Confirmation code request has failed</string>
226      -      <string name="dialog_title_password_changed">Password successfully changed!</string>
221      +      <string name="dialog_title_login_failed">Inicio de sesion fallido</string>
222      +      <string name="dialog_title_sign_up_failed">Registro fallido</string>
223      +      <string name="dialog_title_sign_up_success">Registro exitoso!</string>
224      +      <string name="dialog_title_conf_code_sent">Codigo de confirmacion enviado</string>
225      +      <string name="dialog_title_conf_code_req_fail">Falló la solicitud del codigo de confirmacion</string>
226      +      <string name="dialog_title_password_changed">Se cambio correctamente la contraseña!</string>
```

18 de ago. de 2021

Se dividieron los responsables de cada uno de los equipos.

Instalaron Android Studio

Descargaron fuentes de app rainmaker (Android)

18 de ago. de 2021 - 19 de agosto 2021

Nos aseguramos de que cada equipo tenga un medio de comunicación propio. Como podría ser Discord, Whatsapp, Telegram o el que les pareciera más cómodo. Se terminó eligiendo un grupo de WhatsApp para la difusión de los contenidos, un servidor de Discord para tener un espacio de comunicación entre alumnos de diferentes equipos, y finalmente el Trello donde se plasman todas las actividades.

18 de ago. de 2021 - 27 de sep. de 2021

Pensamos y citamos una reunión con EII - Realizamos preguntas y ayuda con diversos temas del proyecto.

19 de ago. de 2021

Visitamos la EII, y tomamos medidas para la zona de la huerta.





19 de ago. de 2021 - 2 de sep. de 2021

Nos pusimos en contacto con el organizador de Android para la paleta de colores de la App de Android.



Se realizó un promedio de opiniones para modificar el logo de YAKU.

25 de ago. de 2021

Se explicó como realizar la rendición de cuentas para el presupuesto del proyecto, ya que para la compra de cada componente disponíamos de un presupuesto y era necesario cumplir con los requisitos en las facturas.

5 de ago. de 2021 - 11 de ago. de 2021

Se revisó y definió la lista de materiales.

Cantidad	Artículo	Descripción	Rubro	Precio unitario	Importe	Link
1	20m. Cable UTP		Conectividad	\$610,00	\$610,00	shorturl.at/oVHP1
3	Jack RJ45 autocrimpeable Cat5e		Conectividad	\$312,00	\$936,00	shorturl.at/uKUZ8
1	Router/AP/Switch		Conectividad	\$1.300,00	\$1.300,00	shorturl.at/mBC27
1	x10 Fichas RJ45 Macho		Conectividad	\$490,00	\$490,00	shorturl.at/cdtyR
2	5x Bornera 3 terminales		Electrónica	\$200,00	\$400,00	shorturl.at/lmsN1
10	Bornera 2 terminales		Electrónica	\$98,00	\$980,00	shorturl.at/uIW45
1	Display LCD 20x4		Electrónica	\$839,20	\$839,20	shorturl.at/emezx1
1	ESP-CAM	Módulo WiFi + Cámara	Electrónica	\$1.600,00	\$1.600,00	shorturl.at/hxyN8
2	ESP32	Microcontrolador + WiFi/Bluetooth	Electrónica	\$1.390,00	\$2.780,00	shorturl.at/kxCU3
1	Interfaz I2C - Display LCD		Electrónica	\$344,00	\$344,00	shorturl.at/bNRTU
1	Kit panel fotovoltaico		Electrónica	\$8.909,00	\$8.909,00	shorturl.at/pDEL3
1	Memoria microSD (al menos 4 GB)	Para almacenado de imágenes de ESP-CAM	Electrónica	\$543,00	\$543,00	shorturl.at/dfmGI
1	Módulo 4 relé		Electrónica	\$714,00	\$714,00	shorturl.at/abzY7
3	Switch 2 posiciones		Electrónica	\$53,00	\$159,00	shorturl.at/bcBI6
2	Válvula solenoide		Electrónica	\$1.335,00	\$2.670,00	shorturl.at/dZJL9
5	1m. Cable plano Dupont		Electrónica	\$421,00	\$2.105,00	shorturl.at/devEK
1	x2 Placa Virgen PCB (doble faz)		Electrónica	\$1.390,00	\$1.390,00	shorturl.at/kgMO3
1	x20 Terminales Dupont Macho		Electrónica	\$180,00	\$180,00	shorturl.at/iorn1
1	x25 Terminales Dupont Hembra		Electrónica	\$132,00	\$132,00	shorturl.at/pvFO8
1	x5 Leds 5mm	Colores varios	Electrónica	\$42,00	\$42,00	shorturl.at/hK137
1	x5 Placas experimentales		Electrónica	\$251,00	\$251,00	shorturl.at/bipIM
1	Pinza crimpeadora Dupont		Herramientas	\$4.100,00	\$4.100,00	shorturl.at/ksuAL

1	x5 Tee para microtubos		Jardinería	\$150,00	\$150,00	shorturl.at/hiP
1	10m de microtubo		Jardinería	\$250,00	\$250,00	shorturl.at/uLRY1
2	Adaptador de 3/4" a 1/2"		Jardinería	\$126,00	\$252,00	shorturl.at/dftMY
2	Bomba de agua R835		Jardinería	\$1.290,00	\$2.580,00	shorturl.at/gn369
2	Adaptador de 3/4" a Microtubo		Jardinería	\$700,00	\$1.400,00	shorturl.at/LYZ49
1	Microaspersores		Jardinería	\$390,00	\$390,00	shorturl.at/fnoB6
2	Sensores de presencia de agua		Jardinería	\$647,00	\$1.294,00	shorturl.at/eKMQ6
2	Caudalímetro		Sensores	\$2.500,00	\$5.000,00	shorturl.at/aikP0
1	DHT22	Sensor de temperatura y humedad ambiente	Sensores	\$419,00	\$419,00	shorturl.at/cmrCT
8	Higrómetro para suelo	Sensor de humedad en suelo (capacitivo)	Sensores	\$380,00	\$3.040,00	shorturl.at/dntCO
1	Módulo INA 219	Sensor de corriente y tensión	Sensores	\$697,00	\$697,00	shorturl.at/loL89
1	x10 Sensor LDR		Sensores	\$141,00	\$141,00	shorturl.at/sFHNX
2	Caja estanca 10x10x6		Varios	\$231,00	\$462,00	shorturl.at/dKGJ5
1	Caja estanca 30x38x12		Varios	\$2.421,00	\$2.421,00	shorturl.at/bgsA4
				TOTAL	\$49.970,20	

6 de ago. de 2021

Reunión por Meet con Maximiliano Lopez (mentor de INTEL), duró algo más de una hora y nos sirvió para confirmar decisiones ya tomadas, plantear nuevos requisitos y establecer prioridades.

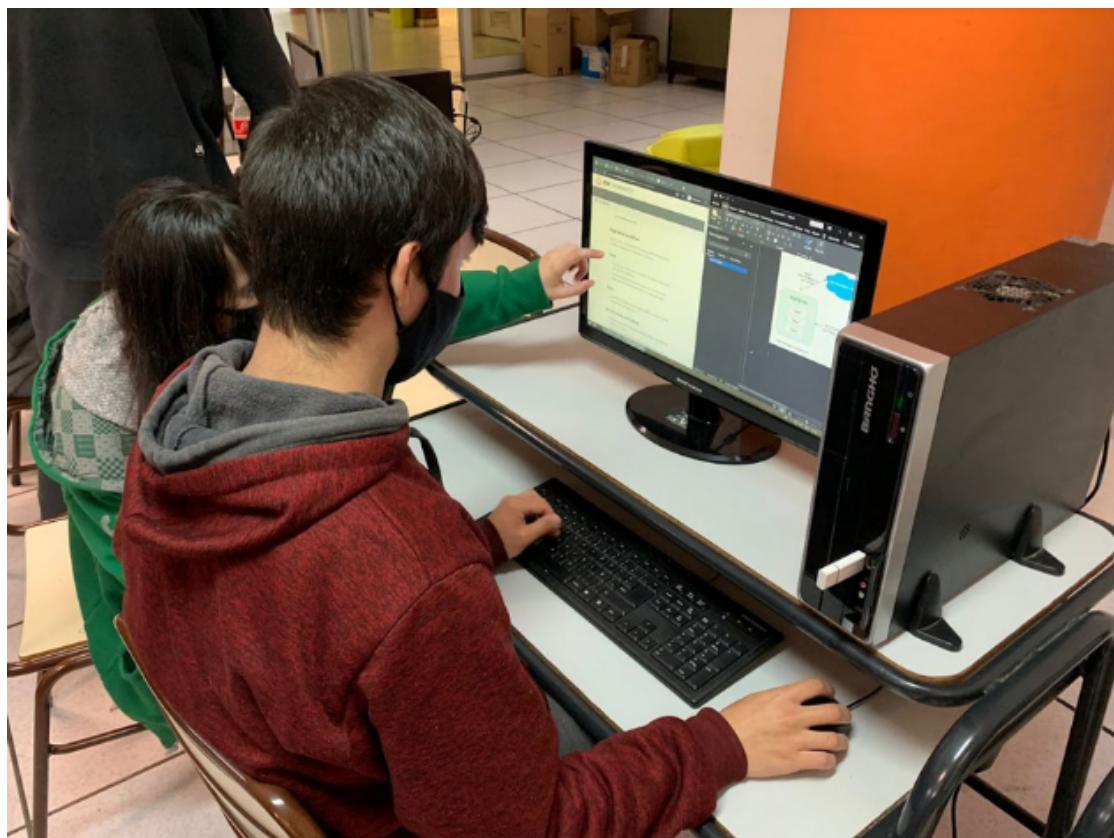
- Darle prioridad al desarrollo en detrimento del hardware (principalmente por costos).
- Sacar la unidad de control de nuestra placa y el display LCD. Controlaremos desde la app/plataforma web (reducimos gastos y no perdemos capacidades)
- Segmentar la interfaz de usuario pensando en dos perfiles con el objetivo de brindar una mejor experiencia de usuario. Los perfiles serán: modo básico (lo mínimo indispensable para el control de la huerta) y modo experto (aquí se incluirán el resto de las funcionalidades).
- Investigar sobre la existencia de sensores biodegradables.
- Conectar el dispositivo a una página web con pronósticos climáticos para avisar al usuario que va/no va a llover al usuario y así influir en la decisión de regar o no.
- Medir otros parámetros para darle info al usuario (simil estación climática)

1 de sep. de 2021

Se estableció una reunión en la que se habló acerca de las actividades ya subidas, como les había ido, si había algún inconveniente o algo que agregar. Se intentaba tener una reunión y contacto con los miembros de cada grupo para no perder el progreso del proyecto y para tener un seguimiento actualizado.

1 de sep. de 2021 - 20 de oct. de 2021

Se tradujeron y divulgaron conceptos relevantes del documento de ESP-Rainmaker. Ya que su documentación estaba en inglés, y para el uso en común y entendimiento de todos los miembros del grupo, se tradujo al español.



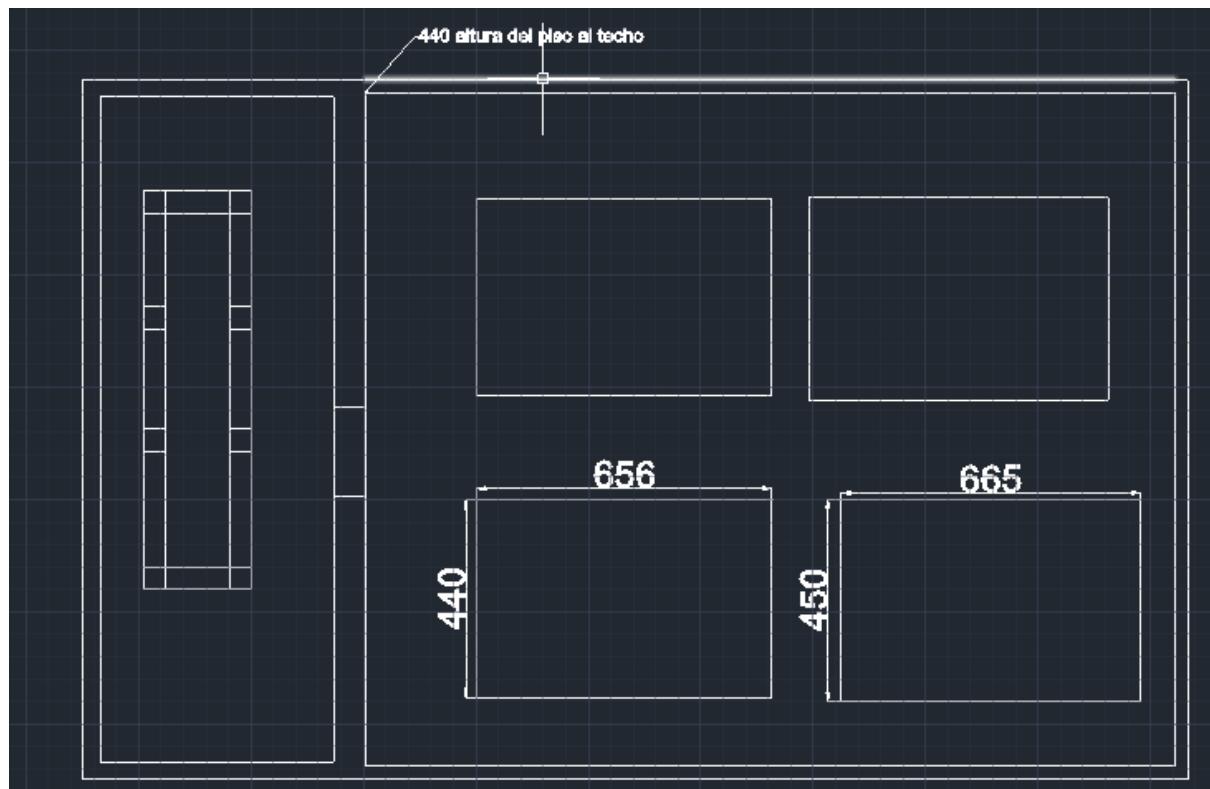
1 de sep. de 2021 - 13 de sep. de 2021

Se realizó el ícono de la app.



2 de sep. de 2021 - 9 de sep. de 2021

Se realizaron los planos de la huerta en Autocad, con la idea de implementarlo en la EII..



8 de sep. de 2021

Reunión al aire libre: conocimos a los integrantes de cada grupo en persona para mejorar la relación de los grupos y poder generar nuevas amistades dentro del colegio, además tuvimos la oportunidad de compartir algo para comer entre todos.

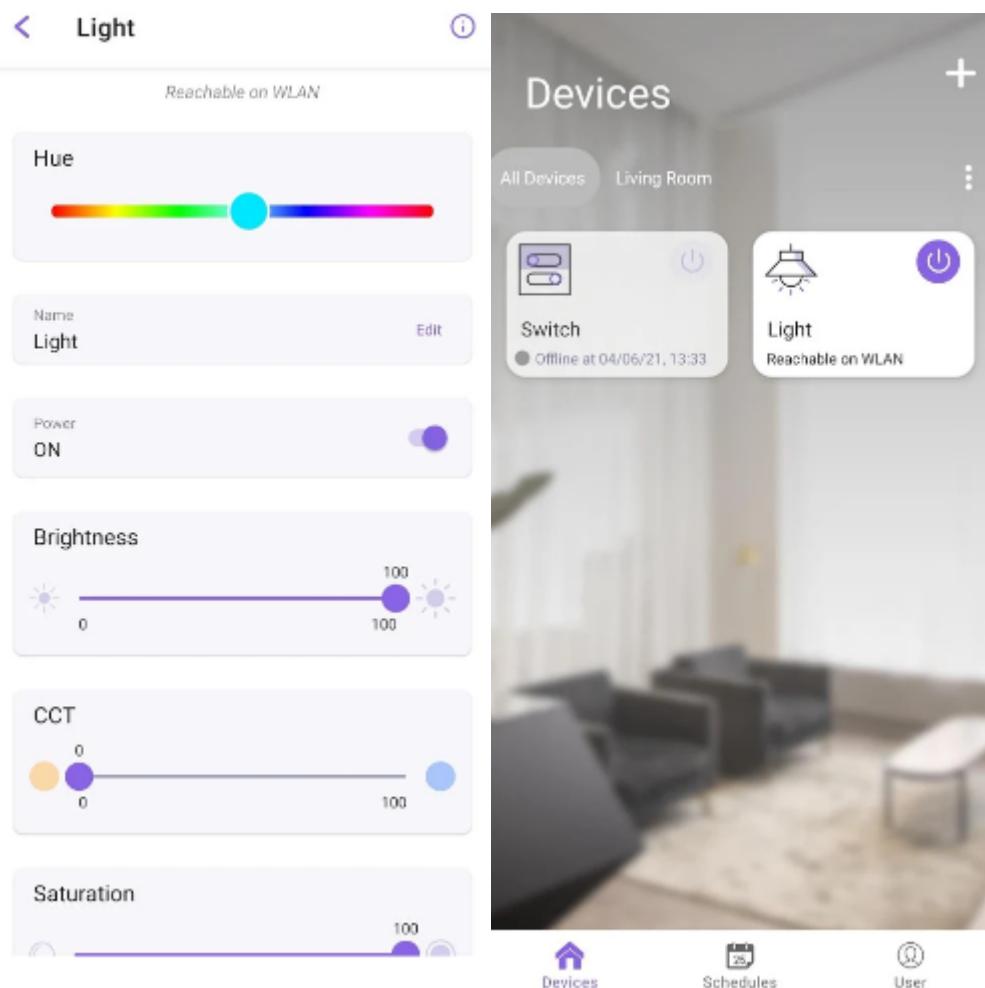
Se confirmó finalmente YAKU como el nombre del proyecto.

2 de sep. de 2021 - 10 de sep. de 2021

Aprendimos sobre las variables de entorno que fueron esenciales para la realización del bot de Python, ya que en el código necesitábamos declarar contraseñas de APIs y logins, por lo tanto fue de gran ayuda este concepto.

6 de oct. de 2021 - 18 de oct. de 2021

Creamos una guía de vinculación y led GPIO para el equipo de Firmware de la ESP32.



27 de sep. de 2021

Se investigó acerca de PowerUps para sumar al trello. Se terminaron agregando PowerUps para poder visualizar todas las tarjetas en un calendario y otro para darle un cronometraje a las tareas.

13 de oct. de 2021

Nos reunimos con el EI con la intención de que todo el resto de compañeros tenga la oportunidad de informarse, y hacer preguntas de lo que necesiten, para encontrar nuevas soluciones o mejoras.

18 de oct. de 2021 - 25 de oct. 2021

Aprendimos sobre los entornos virtuales en el bot de Python, esto fue de gran ayuda para poder descargar los paquetes necesarios para el programa ya que usábamos el programa en distintas computadoras.

18 de oct. de 2021 - 18 de nov. de 2021

Trabajo final del equipo de Python - Unificación de todas las prácticas para poner el bot en función, y dejando espera del equipo de Firmware para implementarlo en su totalidad de nuestra etapa dentro del proyecto. La imagen muestra parte del código.

```
def main():
    data = accounts.data
    response = requests.post(url='https://api.rainmaker.espressif.com/v1/login', data=json.dumps(data)) #Devuelve
    cargado = json.loads(response.content) # Carga el contenido de los tokens

    headers={"Authorization": cargado['accesstoken']}
    #nodesList = requests.get(url='https://api.rainmaker.espressif.com/v1/user/nodes', headers=headers)
    #nodeList = json.loads(nodesList.content)

    nodoUno = json.loads((requests.get(url="https://api.rainmaker.espressif.com/v1/user/nodes/params?node_id=1")
    #nodoUno_device = nodoUno['Temperature Sensor'][Name']
    #nodoUno_val = nodoUno['Temperature Sensor'][Temperature]
    #nodoUno_val = nodoUno['Temperature Sensor'][Temperature]
```

9 de dic. de 2021

Se utilizó una [herramienta web](#) del sitio batterystuff.com para definir la cantidad de celdas solares y baterías necesarias. Arrojando como resultado dos paneles solares de 10 Watts.

Calculation Type	Value
Estimated Watt demand	
3 Total Watts Per Hour (DC) DC Amps x System Voltage	50 Watts
Hours per day	
6 Hours Equip is expected to run (24hr) as per application	1 Hrs d ⁻¹
Watt-Hours per day	
9 Total daily usage Watts x Hours	50.00 Watt-Hrs d ⁻¹
Amp-hour calculation	
10 Total watts Daily requirements	50.00 Watt-Hrs d ⁻¹
11 Corrected for battery losses Assumes static average loss	51.000 Watt-Hrs d ⁻¹
12 System voltage DC voltage only	12 Volts
13 Amp-hour per day Watts divided by Volts	4.250 Amp-Hrs d ⁻¹
Battery bank calculation	
14 # of days backup power required Average 24 hour periods	1 days
15 Amp-hour storage Raw capacity you need	4.2500 Amp-Hrs
16 Depth of discharge Assumes 50%	0.5 fraction (enter decimal)
17 Required amp backup Prevents excessive discharge	8.5000 Amp-Hrs
18 Battery Amp Rating (20 hr) Battery Capacity in Amps	12 fraction
19 Actual # batteries wired in parallel Raw number	0.71
20 Batteries wired in series Relates to system voltage	1.00
21 Rounded number of Batteries Always rounded up	1
Solar Panel Array calculation	
22 Sun hours per day (Direct only) Be realistic!	8 Hrs
23 Worst-weather multiplier* 1.55 default	1.55 fraction
24 Total sun hours per day Assumes average sun	5.161 Amp-Hrs
25 Select panel size (Watt rating) Watt hour rating	10 Watts
26 Nominal Panel Voltage Approximate Solar output	16 Volts
27 Amps required from solar panels Total daily consumption	4 Amps
28 Peak amperage of solar panel Watts divided by Volts	0.625 Amps
29 Number of solar panels in parallel Raw Number	1.318
30 Number of panels in series (12 V) it is 1 for 12v, 2 for 24v, etc	1
31 Rounded number of solar panels Always rounded up	2

3 de jun. de 2022 - 9 de jun. de 2022

Creamos un repositorio de GitHub general.

 Feduski	Update README.md	f46003f yesterday	 12 commits
	Android	Second Commit	19 days ago
	Diseño Grafico	Second Commit	19 days ago
	Documentacion	new files	yesterday
	Planos	Plans and diagrams	2 days ago
	Python	First Commit	19 days ago
	Traducion	translation files	2 days ago
	README.md	Update README.md	yesterday

Buscamos alternativas para el alojamiento del programa en Python, pero quedamos en la decisión de usar AWS para ejecutar nuestro programa.

3 de jun. de 2022 - 13 de jun. de 2022

Organizamos la próxima reunión con nuevos integrantes. En ella se iban a tratar los siguientes temas:

- Presentación del proyecto y su situación actual.
- Presentación del Trello y su uso.
- División de grupos de los nuevos integrantes.
- Aviso de reactivación.
- Método de trabajo.
- Incorporación de los nuevos participantes a WhatsApp, Trello y Discord.
- Página Web

3 de jun. de 2022 - 27 de jun. de 2022

Se recuperaron archivos, teniendo en cuenta que hubieron miembros que se bajaron del proyecto, y que algunos archivos se habían quedado fuera de alcance de todos.

4 de junio de 2022

Se asignaron tareas para el grupo de ESP32. Entre ellas:

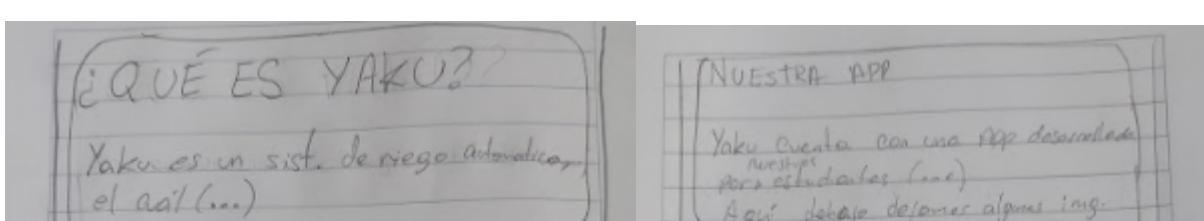
- 07A - Probar uno de los ejemplos que trae la placa para el uso de entradas analógicas
- 07B - Probar uno de los ejemplos de esp-rainmaker (switch p. ej)
- 08A - Modificar el ejemplo para que sea capaz de leer el sensor de humedad en suelo
- 08B - Probar un ejemplo de esp-rainmaker que reporte datos (temperature_sensor)
- 09AB - Modificar un ejemplo de esp-rainmaker para que reporte el valor de humedad en suelo

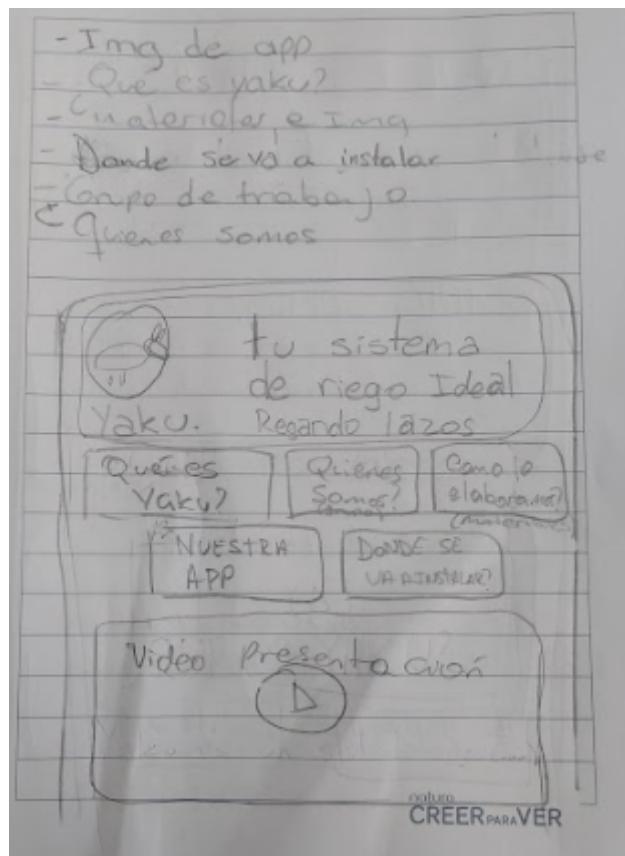
22 de junio de 2022

Se hizo una reunión con los nuevos miembros, se determinaron los integrantes del grupo de desarrollo web, se explicó todo el contenido del proyecto, y se agregaron a los miembros a los medios de comunicación.

22 de jun. de 2022 - 27 de junio de 2022

Diseñamos la Página Web.





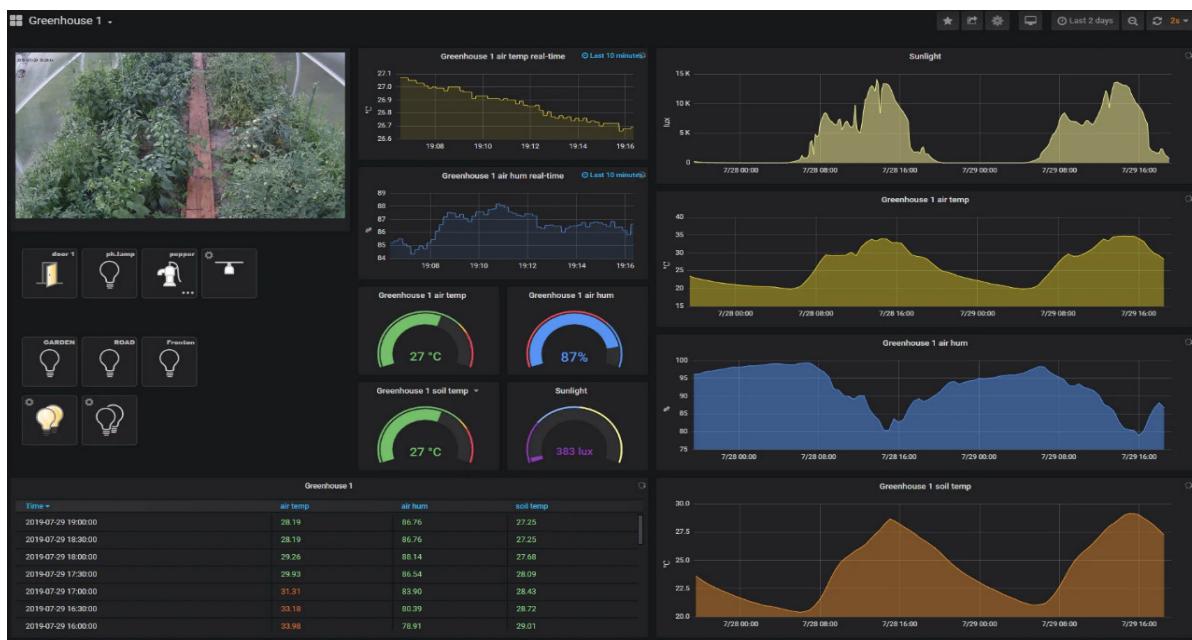
27 de junio de 2022 - En proceso

Se asignaron tareas para el grupo de desarrollo web. Entre ellas:

- 00 - Capacitación HTML.
- 01 - Capacitación CSS.
- 02 - Estructurado de la página en HTML.
- 03 - Darle estilo a la página.

27 de junio de 2022 - En proceso

Se asignó una tarea para el grupo de Métricas para la realización de gráficos con la aplicación Grafana. Esta mostrará las estadísticas de todos los datos recibidos por el dispositivo.



8 de julio de 2022

Concurrimos a la Feria de Ciencias, presentando de 8:30 a 12:00 y de 12:45 a 16:00. Recibimos consejos de muchas personas, como por ejemplo la idea de implementar un sistema capaz de generar energía eólica, para incentivar aún más el ahorro de energía. Tuvimos una devolución de parte de nuestros evaluadores, con la cual nos aportaron las siguientes cosas:

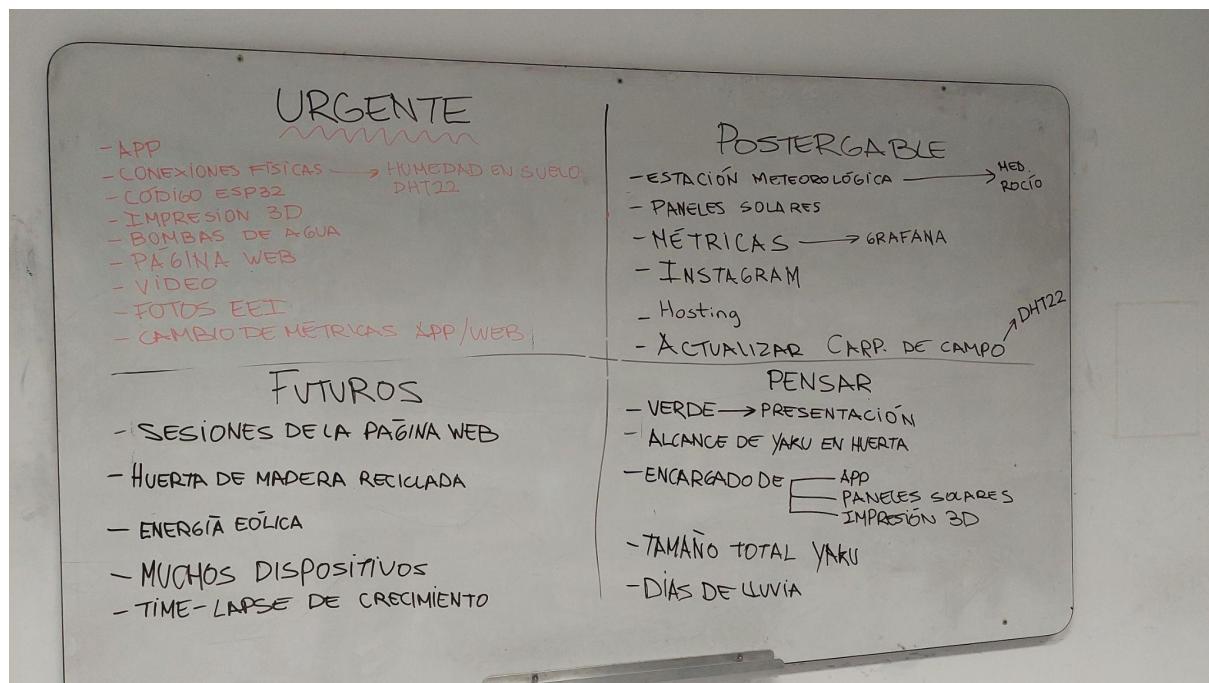
- La idea de implementar una estación meteorológica aprovechando el espacio a ocupar de nuestro proyecto.
- Intentar reducir el tamaño del dispositivo (lo cual ya teníamos en cuenta)
- Buscar un cliente específico (en nuestro caso la escuela EEI)
- Propusieron que para la siguiente feria haya al menos un video, documentación o muestra en persona que demuestre el funcionamiento de YAKU.
- Comentaron que las bombas de agua generaban mucho consumo eléctrico, y nos pidieron mejorarlo (pero por temas de infraestructura, le explicamos porqué era tan necesario su uso)
- Sugirieron realizar un presupuesto, evaluando costos de instalación y mantenimiento.
- Plantearon también que en la bitácora o carpeta de campo debíamos incorporar parte de los fracasos/errores que surgieron en el desarrollo, y evidencia de los mismos.

14 de julio de 2022

Tuvimos una reunión con el grupo de desarrollo web, en la que se informó la situación actual del grupo, y se conversó de que se iba a seguir trabajando durante el receso escolar.

Realizamos un cuadro de prioridad de necesidades, el cual dividimos en ‘Urgente’, ‘Postergable’, ‘Futuros’, ‘Pensar’. Siendo urgente las cosas que propusimos como totalmente necesarias para la feria de ciencias de septiembre, postergable aquello que

puede esperar pero que igualmente es muy importante, futuros como detalles a agregar en un futuro, y pensar, para preguntas las cuales aún no teníamos respuesta (Varias de ellas se resolvieron en ese mismo día en una mini-reunión de Federico Garcia, Jeremias Davison y Sebastian Schachtner).



Agregamos más tareas para cumplir con todo lo plasmado en la imagen (zonas urgente y postergable).

18 de julio de 2022

Recibimos la devolución escrita de los evaluadores.

27 de julio de 2022

Se terminó el código y las conexiones necesarias para poder medir temperatura ambiente y humedad. Se determinó el cambio del sensor DHT11 por un sensor DHT22 que mide temperatura y humedad de una manera más precisa, y en más altos rangos.



31 de julio de 2022

Se terminaron las capacitaciones de HTML y CSS correspondientes al grupo de Desarrollo Web, las cuales habían sido asignadas el 27 de junio.

4 de agosto de 2022

Teniendo en cuenta el código y conexión del sensor DHT22, seguimos con el sensor BMP180 para poder llevar a cabo el consejo de los evaluadores de realizar una estación meteorológica. Con este sensor, se logró medir la presión atmosférica, y la altitud. Se completaron el código y las conexiones necesarias.

A su vez, se aprovechó para investigar sobre el uso de las bombas de agua, y realizar su respectivo código.

6 y 7 de agosto de 2022

El grupo de desarrollo web aprovechó el fin de semana para seguir con el estilo de la página. Se terminó la estructura básica e hicieron un cambio de colores, respetando la paleta de colores. Comentaron la idea de ir mejorando el diseño, que hasta el momento era solo de prueba.





8 de agosto de 2022

Se asignaron nuevas tareas para el grupo de Desarrollo Web. Entre ellas:

- 04 - Capacitación de JAVASCRIPT
- 05 - Login de usuario para administradores.

10 de agosto de 2022

Se habló de que el grupo de Diseño Gráfico haga una paleta de colores / indiquen ellos los colores para la página web. El grupo de desarrollo web avanzó con el código.

Visual Studio Code interface showing two tabs:

- index.html - yaku-web - Visual Studio Code**: Contains the HTML code for the landing page. It includes a header, a section with five buttons (botones), and a video player.
- styles.css - yaku-web - Visual Studio Code**: Contains the CSS styles for the website. It defines classes for buttons (.botones), buttons (.boton), and a responsive container (.carrusel).

```

index.html - yaku-web - Visual Studio Code
EXPLORADOR          index.html
YAKU-WEB
  CSS      # styles.css
  imágenes
  index.html

index.html > html > body > section.botones
  <!-- CARRUSEL YAKU -->
  <link rel="stylesheet" href="CSS/styles.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Nunito&display=swap" rel="stylesheet">
</head>
<body>
  <section class="arriba">
    <header class="encabezado">
      <h1>Tu sistema de Riego Ideal</h1>
      <h3>Regando Lazos</h3>
    </header>
  </section>
  <section class="botones">
    <a class="boton" href="#one">¿Qué es Yaku?</a>
    <a class="boton" href="#two">¿Quiénes somos?</a>
    <a class="boton" href="#three">¿Cómo lo elaboramos?</a>
    <a class="boton" href="#four">Nuestra App</a>
    <a class="boton" href="#five">¿Dónde se va a instalar?</a>
  </section>
  <div class="video">
    <video src="#" poster="#" controls></video>
  </div>
  <section>
    <div id="one" class="respuestas">
      <h2>¿Qué es Yaku?</h2>
      <p>Yaku es un sistema de riego automático el cual(...)</p>
    </div>
    <div id="two" class="respuestas">
      <h2>¿Quiénes somos?</h2>
      <p>Somos chicos de la ET36 D.E. 15, ubicada en Saavedra CABA</p>
    </div>
    <div id="three" class="respuestas">
      ...
    </div>
  </section>

```

```

Lín. 26, col. 15 Espacios:4 UTF-8 CRLF HTML Go Live R Q
Windows Taskbar: Símbolos, Buscador, Navegador, Explorador, Ejecutar, Salir
Lín. 26, col. 15 Espacios:4 UTF-8 CRLF HTML Go Live R Q
Windows Taskbar: Símbolos, Buscador, Navegador, Explorador, Ejecutar, Salir

```

```

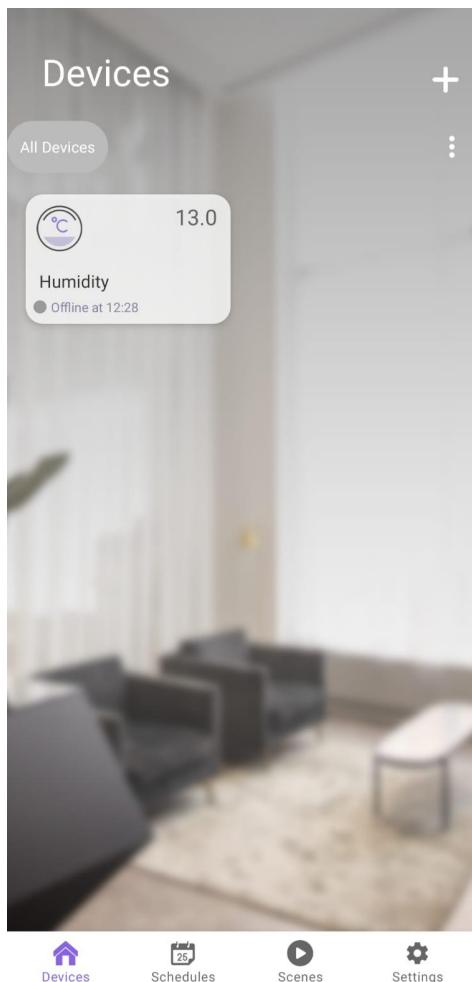
styles.css - yaku-web - Visual Studio Code
EXPLORADOR          index.html
YAKU-WEB
  CSS      # styles.css
  imágenes
  index.html

# styles.css > # styles.css > .botones
  .botones{
    text-align: center;
    margin-bottom: 70px;
  }
  .boton{
    border: 1px solid black;
    padding: 15px;
    background-color: #rgb(99, 33, 129);
    color: black;
    text-decoration: none;
    text-transform: uppercase;
    border-radius: 50px;
    margin: 20px;
    display:inline-flex;
  }
  .respuestas{
    margin: 190px;
    border: 5px solid #rgba(174, 0, 197, 0.281);
  }
  .carrusel{
    align-content: center;
    text-align: center;
    width: 240px;
    height: 270px;
    overflow-x: scroll;
    scroll-snap-type: x mandatory;
    display: flex;
    margin: 0 auto;
  }
  .carrusel img{
    flex: 0 0 100%;
    width: 100%;

Lín. 28, col. 13 Espacios:4 UTF-8 CRLF CSS Go Live R Q
Windows Taskbar: Símbolos, Buscador, Navegador, Explorador, Ejecutar, Salir

```

El equipo de ESP32 avanzó con su parte, y terminaron de conseguir datos del sensor de humedad en tierra, pudiéndose subir a la APP.

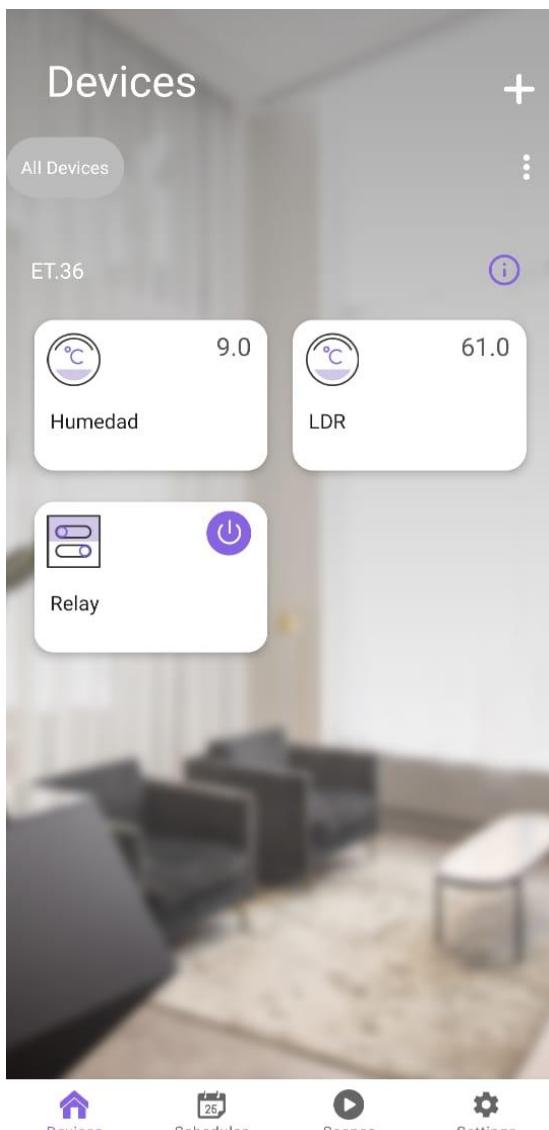
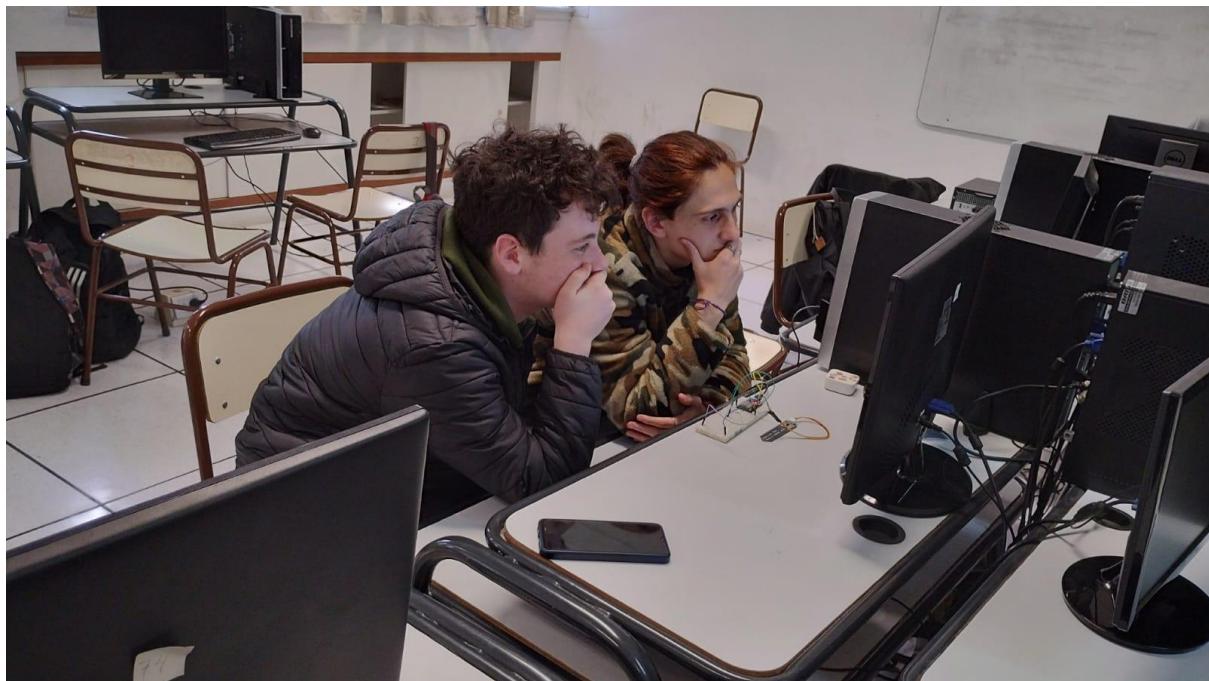


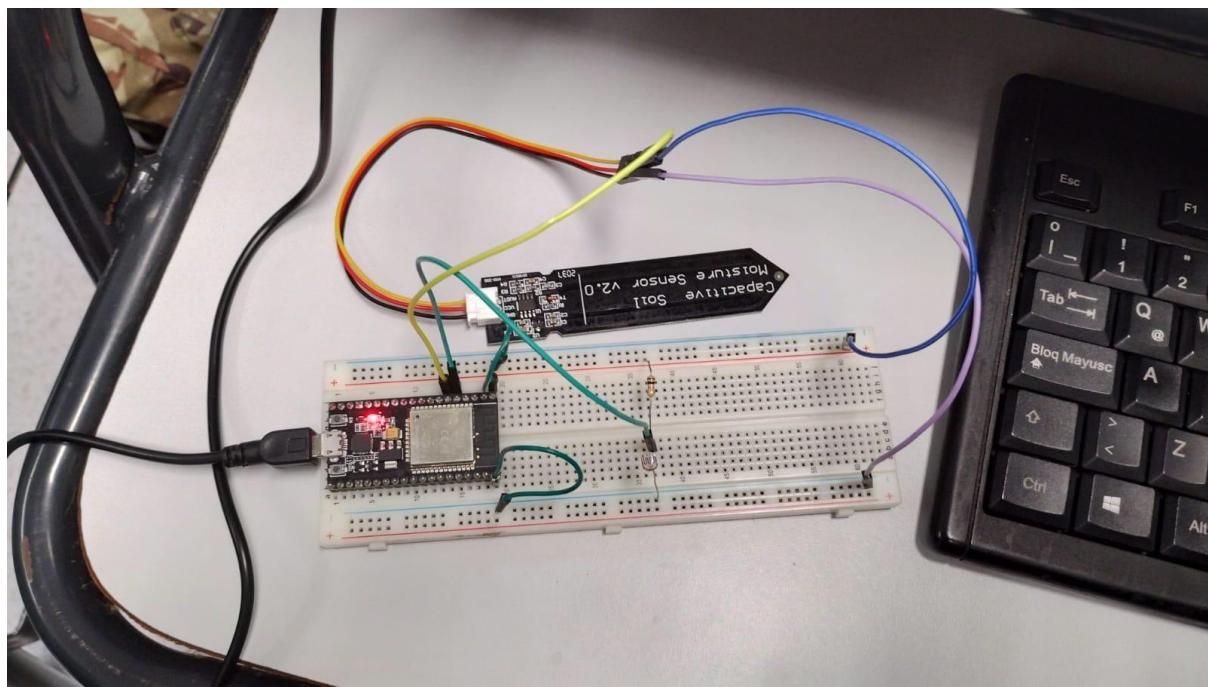
11 de agosto de 2022

Se agregó una nueva tarea para el grupo de ESP32: 'EXTRA-Investigación Rainmaker'

17 de agosto de 2022

El equipo de ESP32 logró subir a la aplicación los valores recibidos por el sensor LDR, sumado a un botón que próximamente activará un relay.



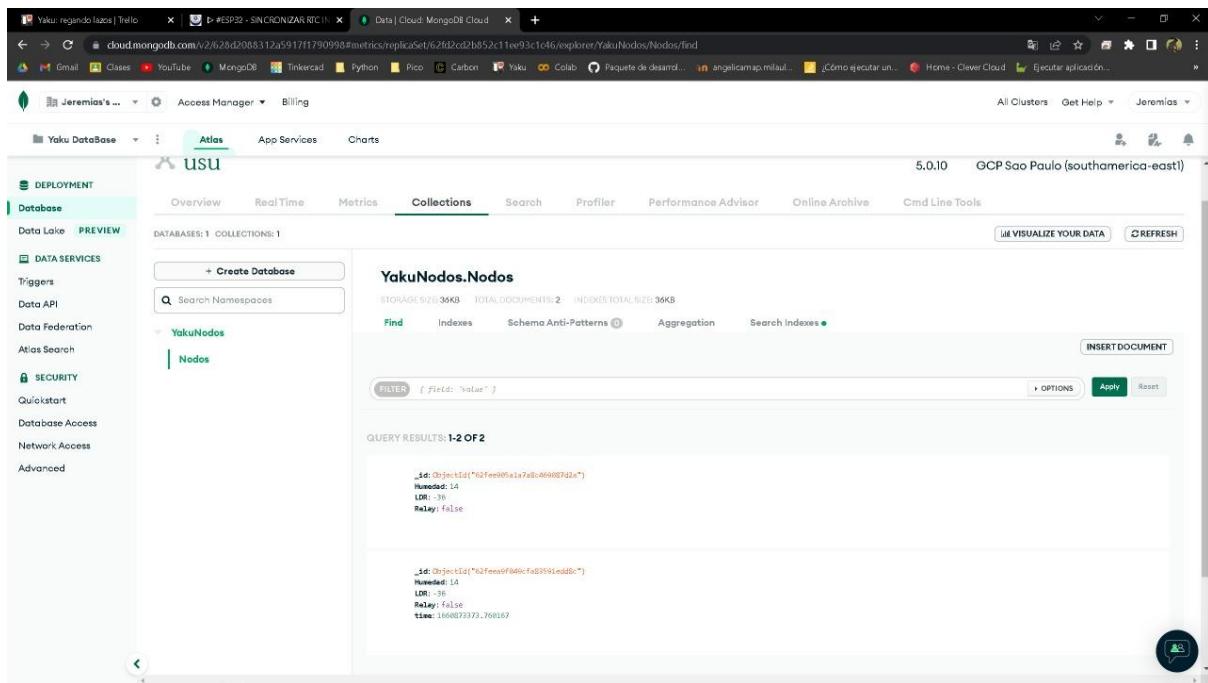


18 de agosto de 2022

Se asignaron nuevas tareas para el equipo de ESP32. Entre ellas:

- 11 - Armar git y dar permisos del código hecho al momento
 - 12 - Documentar lo hecho en la app de ESP32 que reporta datos de (al menos) 1 sensor y actuador
 - 13 - Automático de bomba de agua

Debido a que el equipo de ESP32 terminó con el envío de datos a ESP RainMaker, el equipo de python pudo terminar su parte enviando los datos recibidos a la base de datos de MongoDB (que anteriormente se estaban enviando datos dummies).



The screenshot shows the MongoDB Cloud interface with the YakuNodos.Nodos collection. The interface includes a sidebar with options like Deployment, Database, Data Lake, Data Services, Security, and Advanced. The main area shows the collection details: Storage Size: 36KB, Total Documents: 2, Indexes: 1, and Anti-Patterns: 0. Below this, there are tabs for Find, Indexes, Schema, Aggregation, and Search indexes. A query results table shows two documents:

```

_id: ObjectId("62f7e905ea17a0c690087d2a")
Humedad: 14
LDR: 36
Relay: false
time: 1660877373.798167

_id: ObjectId("62f7e905ea17a0c690087d2b")
Humedad: 14
LDR: 36
Relay: false
time: 1660877373.798167

```

En el equipo de Python también se decidió utilizar una API para poder tener más datos que mostrarle al usuario, y así mejorar su experiencia aún más.

Por otro lado, se establecieron las reglas de negocio sobre las cuales se va a regar o no en nuestro sistema.

regar SI

Es de noche/tarde/esta nublado (fijarse que valor da el LDR, que no debe ser mayor a X) para no ocasionar el efecto lupa

Y

La humedad en tierra es menor a (provisoriamente x defecto pongan un valor de referencia que consideren seco) y en el proceso ir fijándose nuevamente el valor.

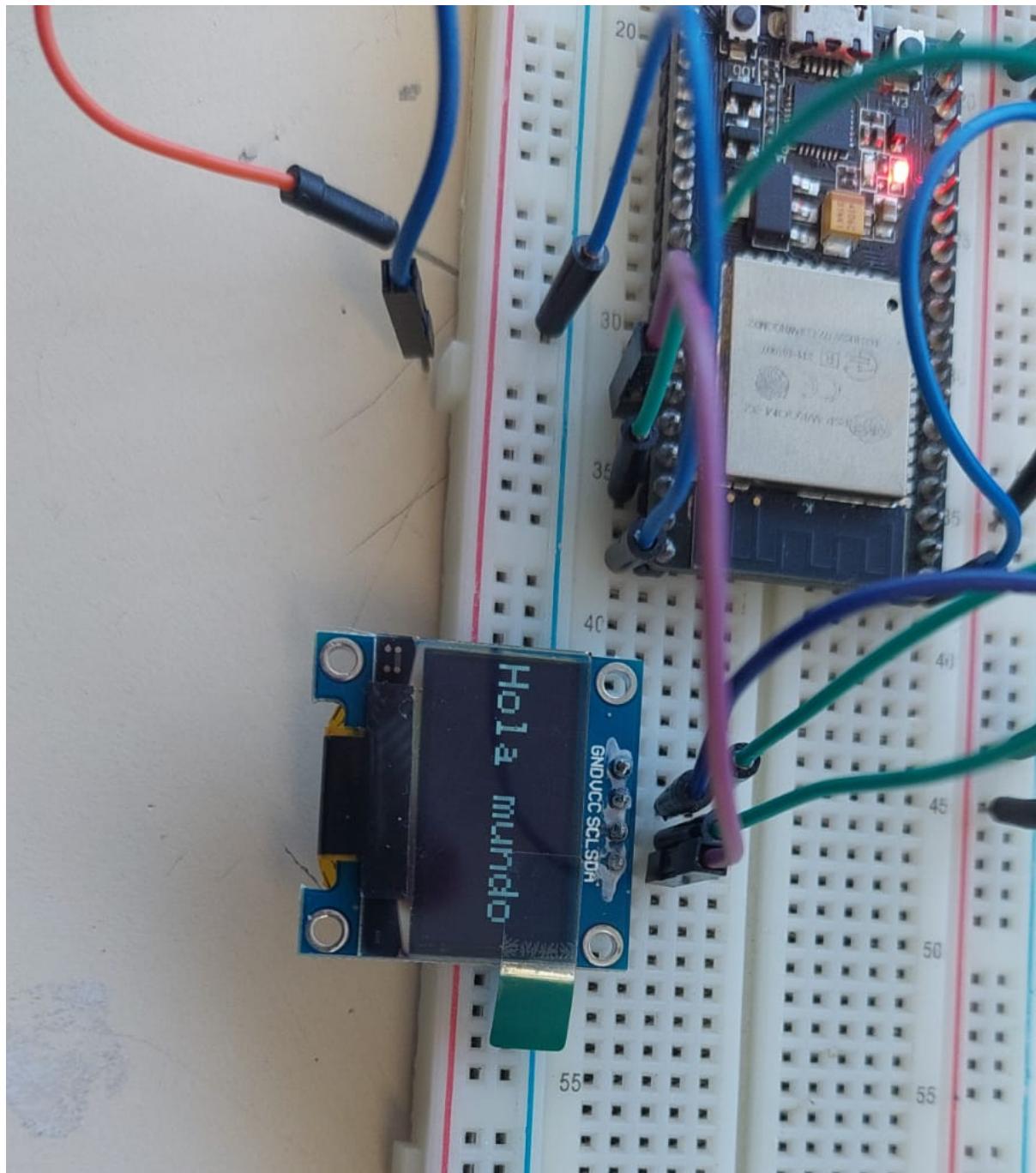
Quizás solamente por ejemplo, abrir la bomba 5 segundos, volver a consultar los valores, abrir 5 segundos de nuevo, hasta que llegue a la humedad deseada.

No hace falta tener en cuenta si regué antes si ya la humedad dejó de ser menor a X número, por eso es importante usar el Y (AND) (&&)

Hacer esta validación cada 15 minutos aprox creo que estaría bien, y para no saturar tampoco, pero no menos de 2 min al menos, porque sino va a terminar regando de más.

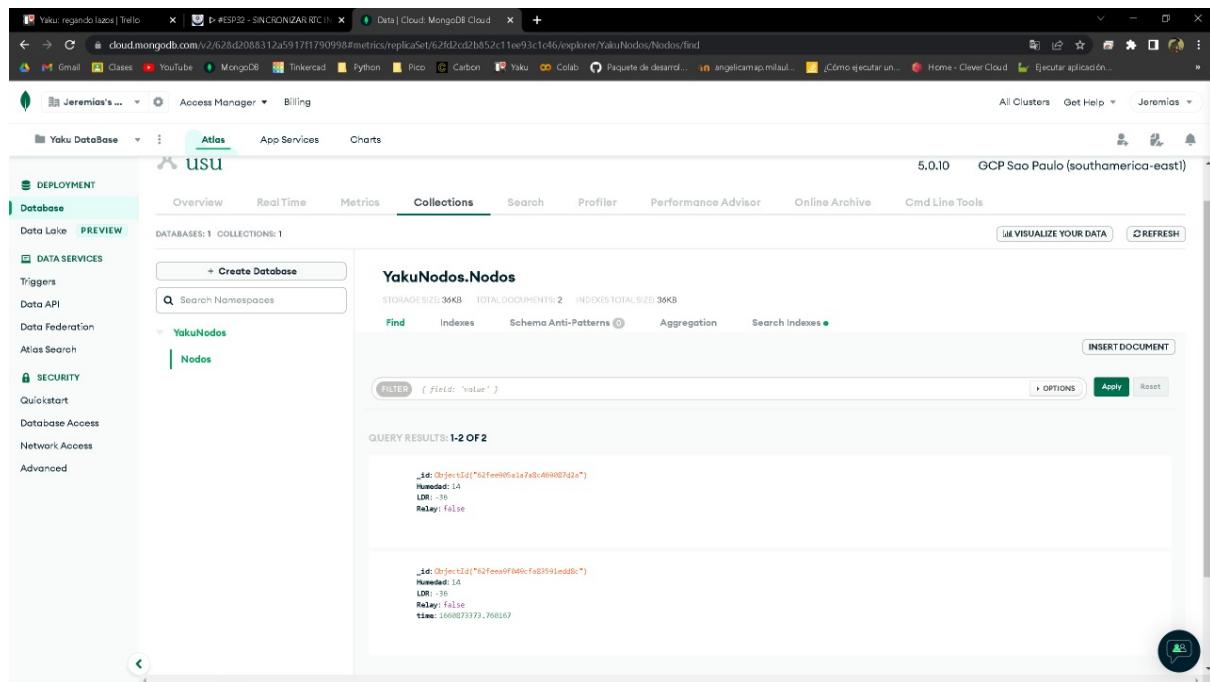
18 de agosto de 2022

Se realizó la conexión y el código para utilizar un panel OLED, el cual va a reflejar datos de importancia, en el momento.



Informe Yaku - ET 36 "Alte. Guillermo Brown"

También se subieron datos verídicos a la base de datos por primera vez.



The screenshot shows the MongoDB Cloud interface. On the left, the sidebar includes 'Jeremias's ...' (selected), 'Access Manager', 'Billing', 'Database' (selected), 'PREVIEW', 'DATA SERVICES', 'SECURITY', and 'Advanced'. Under 'Database', there are 'Triggers', 'Data API', 'Data Federation', 'Atlas Search', and 'Nodos'. The main area shows the 'Atlas' tab selected. Below it, the 'Collections' tab is active, displaying 'YakuNodos.Nodos'. It shows 'STORAGE SIZE: 36KB' and 'TOTAL DOCUMENTS: 2'. The 'INDEXES TOTAL SIZE: 36KB' section is also visible. Two documents are listed:

```

_id: ObjectId("62fe9e05ea7a0460087d2a")
Humedad: 14
LDR: 36
Relay: false
time: 1660873373.760167

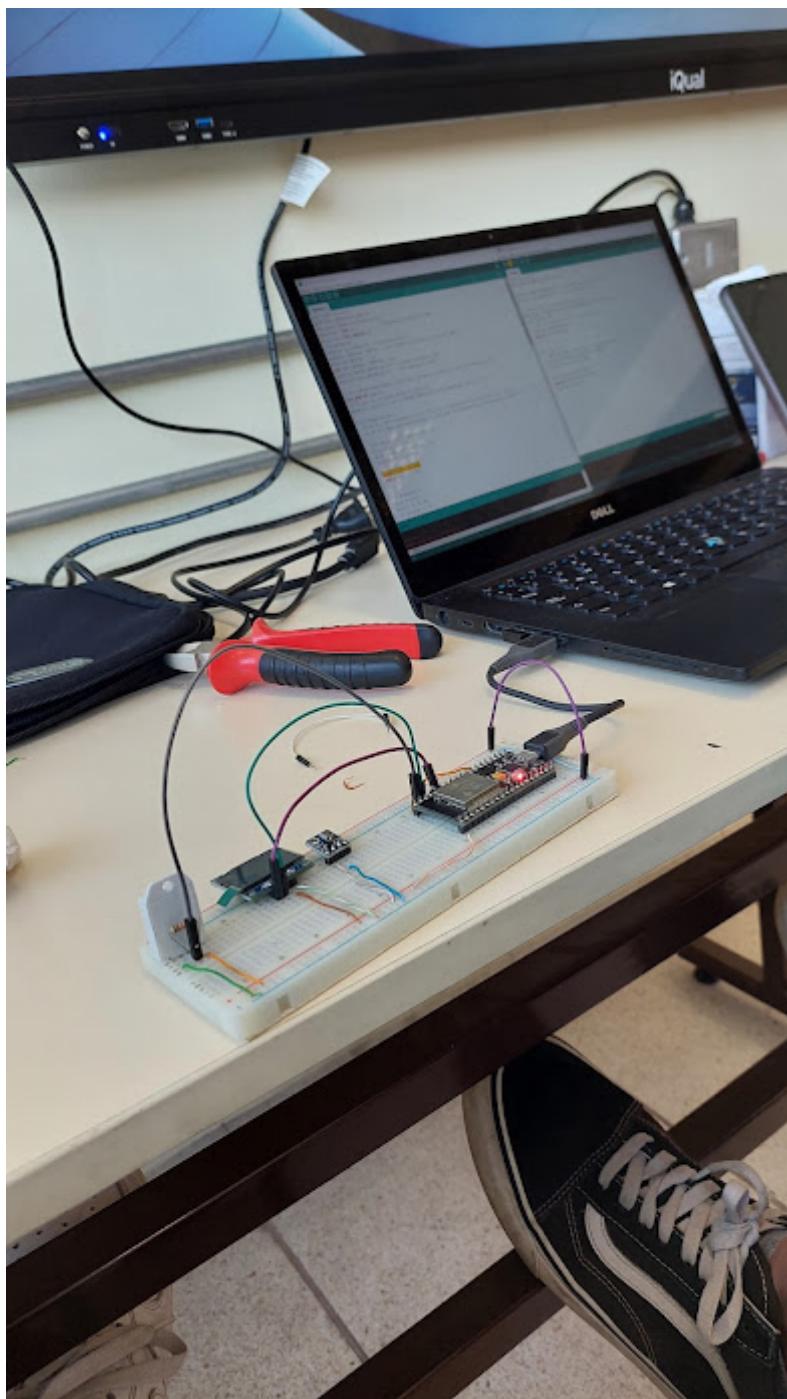
_id: ObjectId("62fe9e0f0ddcfa83591edd8c")
Humedad: 14
LDR: 36
Relay: false
time: 1660873373.760167

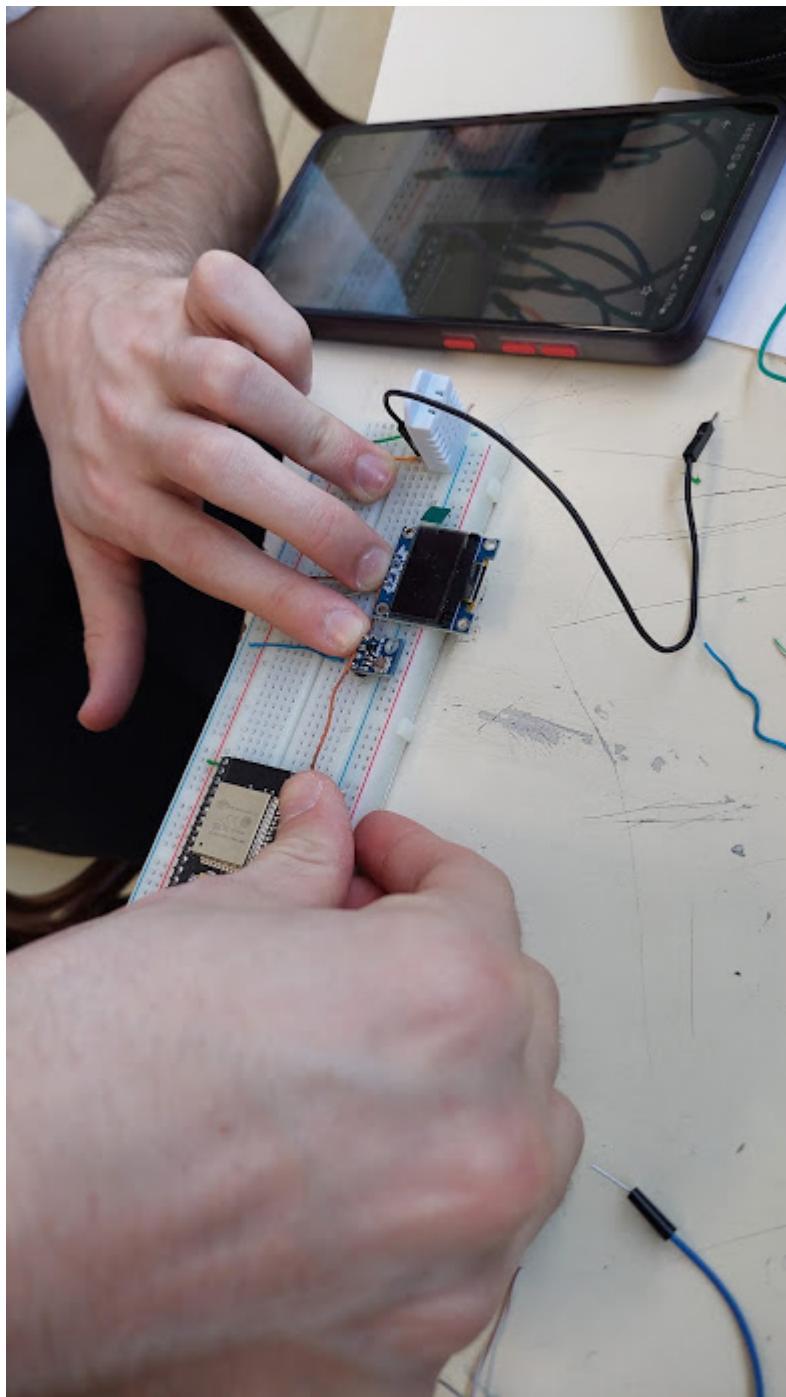
```

24 de agosto de 2022

Comenzamos a realizar recortes de cables pensando en la instalación completa de YAKU, para ya pasar todo a una placa, y dejar de depender de una protoboard.







Este día también se documentó el código que sube los datos recibidos a ESP Rainmaker, que posteriormente son mostrados en la aplicación. Se subió a un repositorio de GitHub.

25 de agosto de 2022

Se logró terminar exitosamente una primera versión del código necesario para el riego automático.

```

if(Water_bomb_Timer.isReady()){
    if(ldr_value < 1000 && humidity_value < 2250){
        digitalWrite(relay, HIGH); relay_state = true;

        // Falta encontrar una función para poner los 5s, esta no puede ser delay ya que esta función frena el programa.

        digitalWrite(relay, LOW); relay_state = false;
        Update_Sensor();
    }
    Water_bomb_Timer.reset();
}
  
```

26 de agosto de 2022

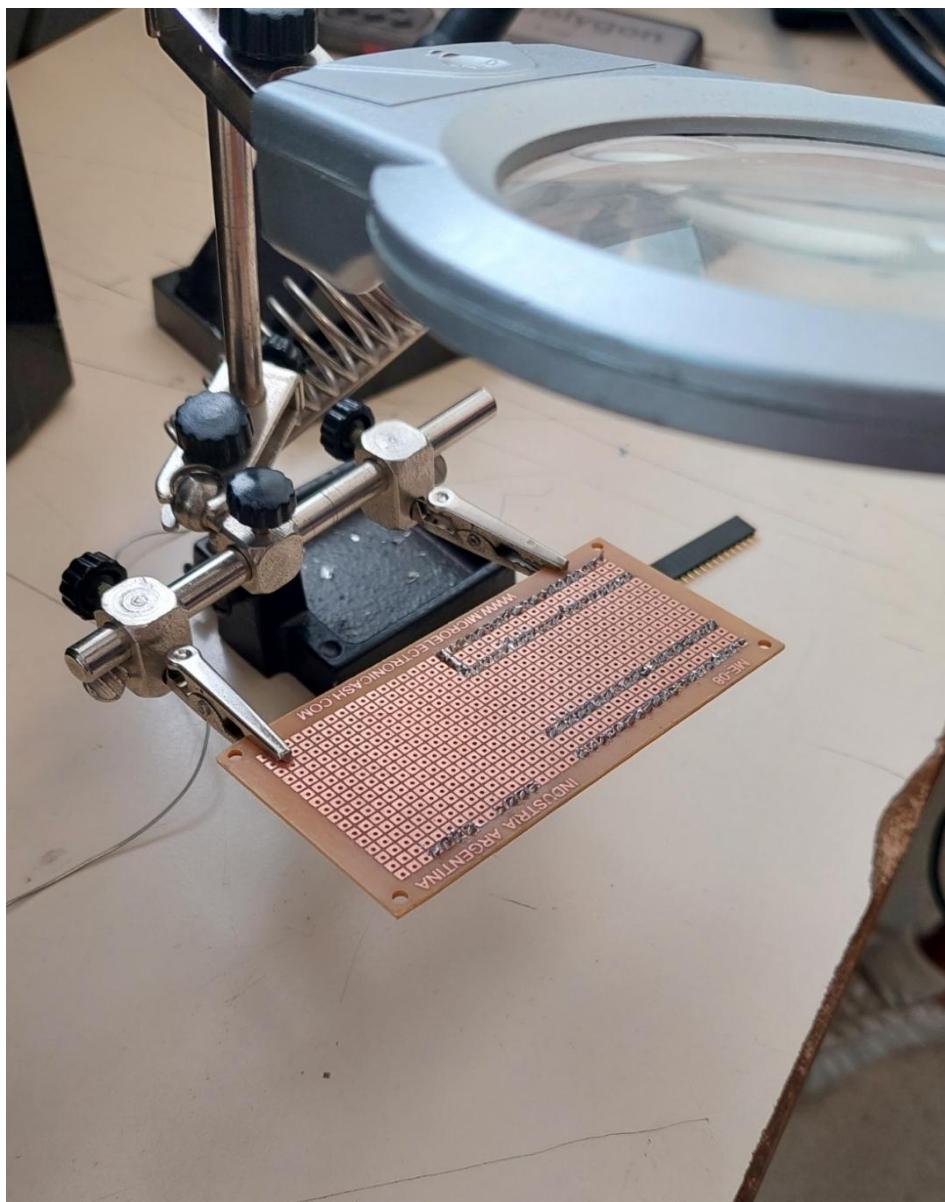
Se realizaron pruebas con los paneles solares que van a alimentar parte de la energía necesaria para YAKU. Se conectaron con el controlador de carga y la batería.

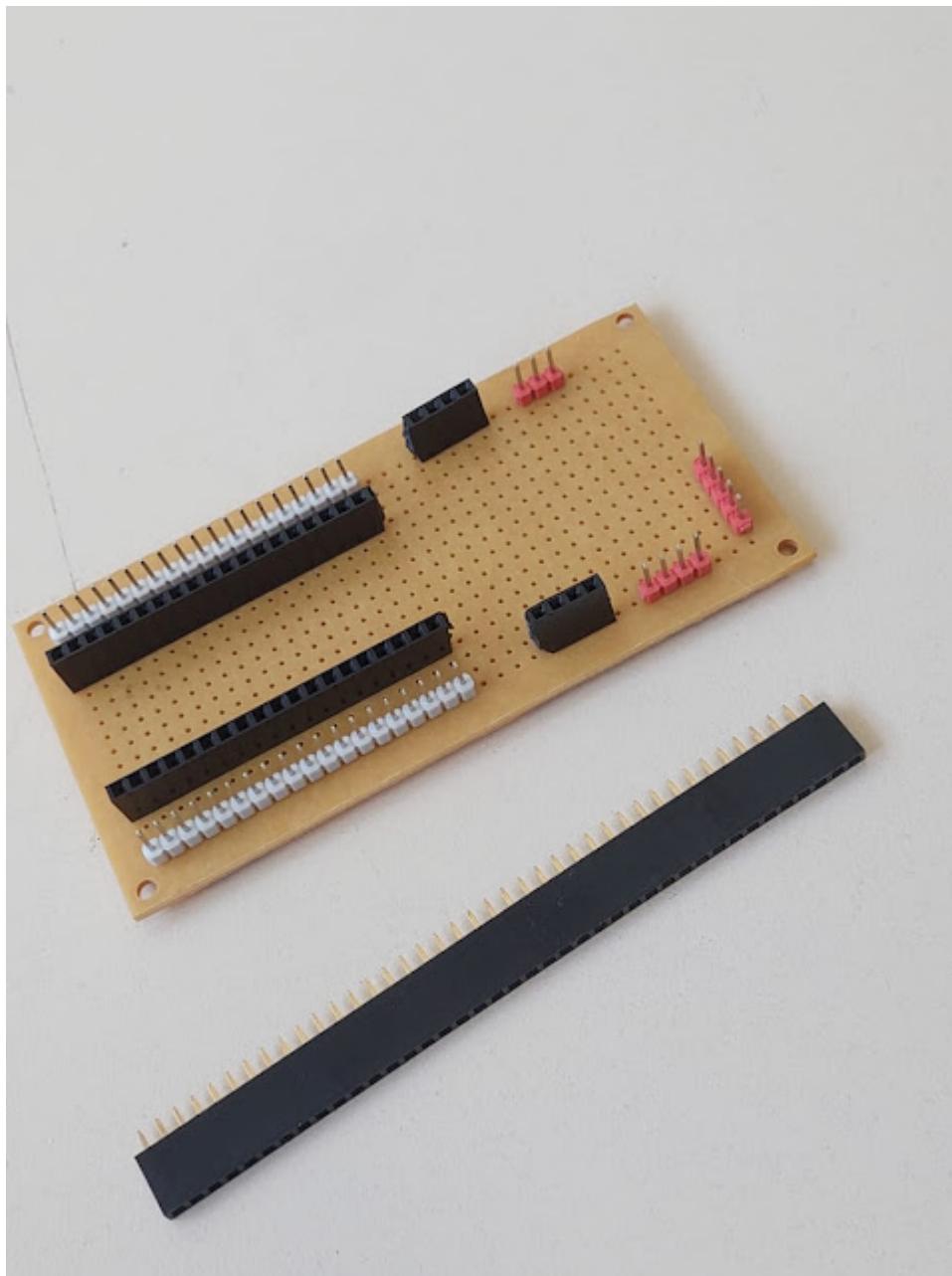




31 de agosto de 2022

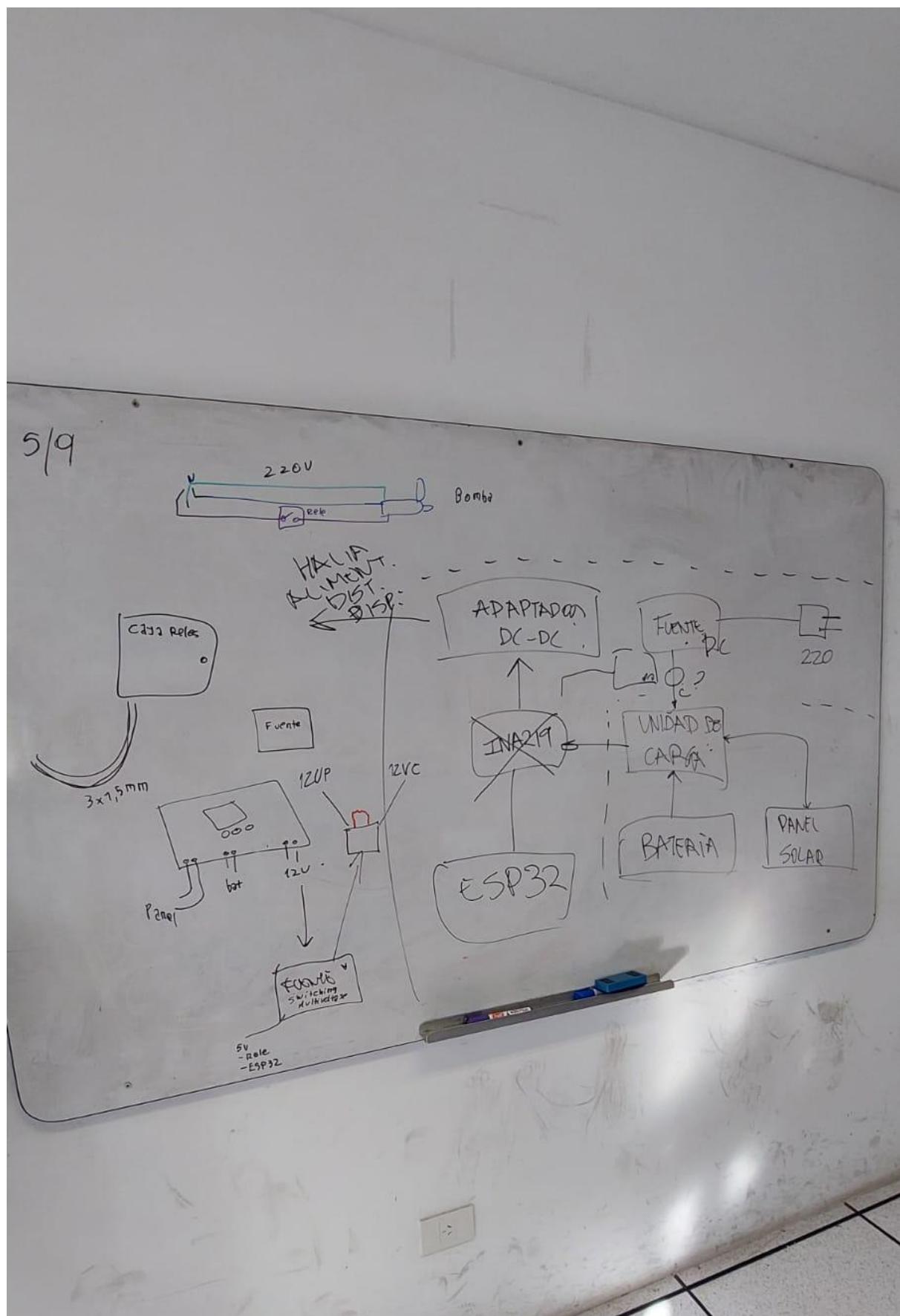
Se soldaron en la placa microperforada, pins macho - hembra, para posteriormente colocar los sensores y actuadores.



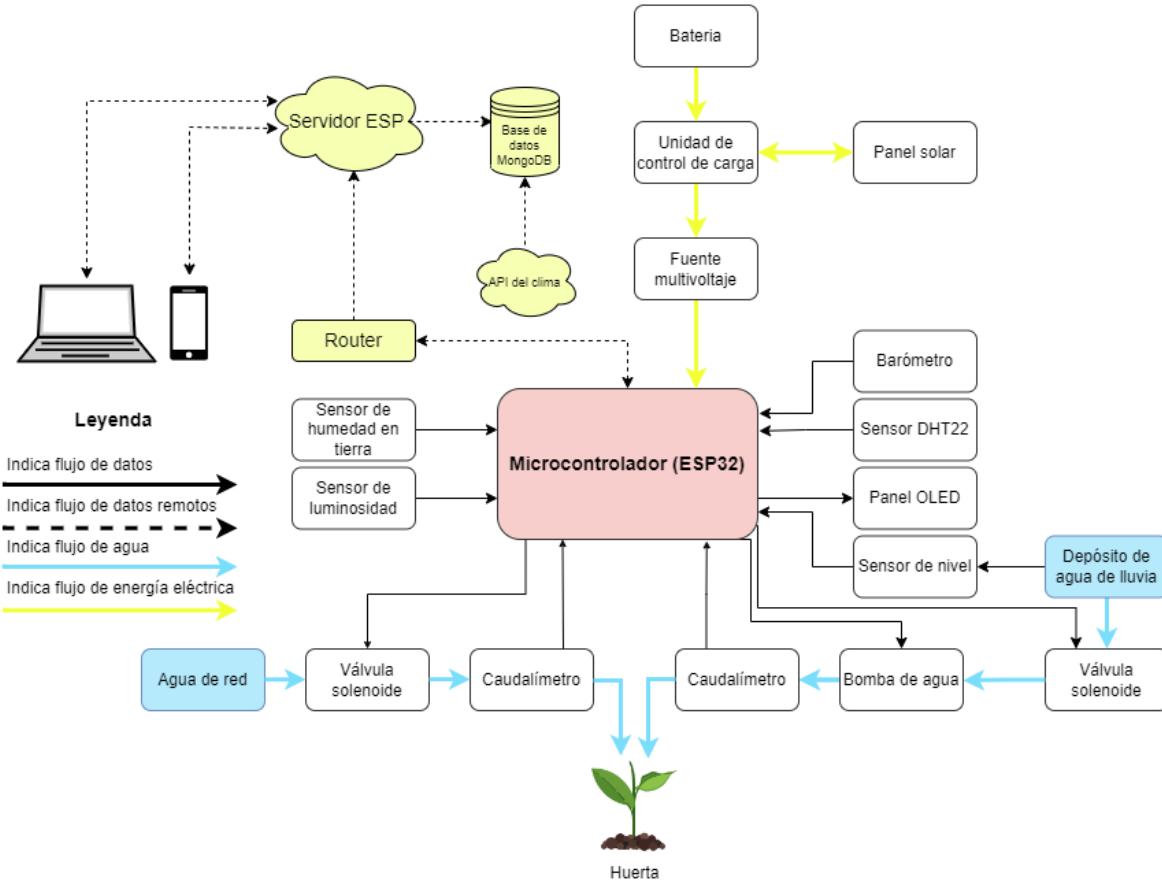


05 de septiembre de 2022

Se reformuló la manera en la que se alimenta YAKU, dejando todo en manos totalmente del panel solar y la batería.

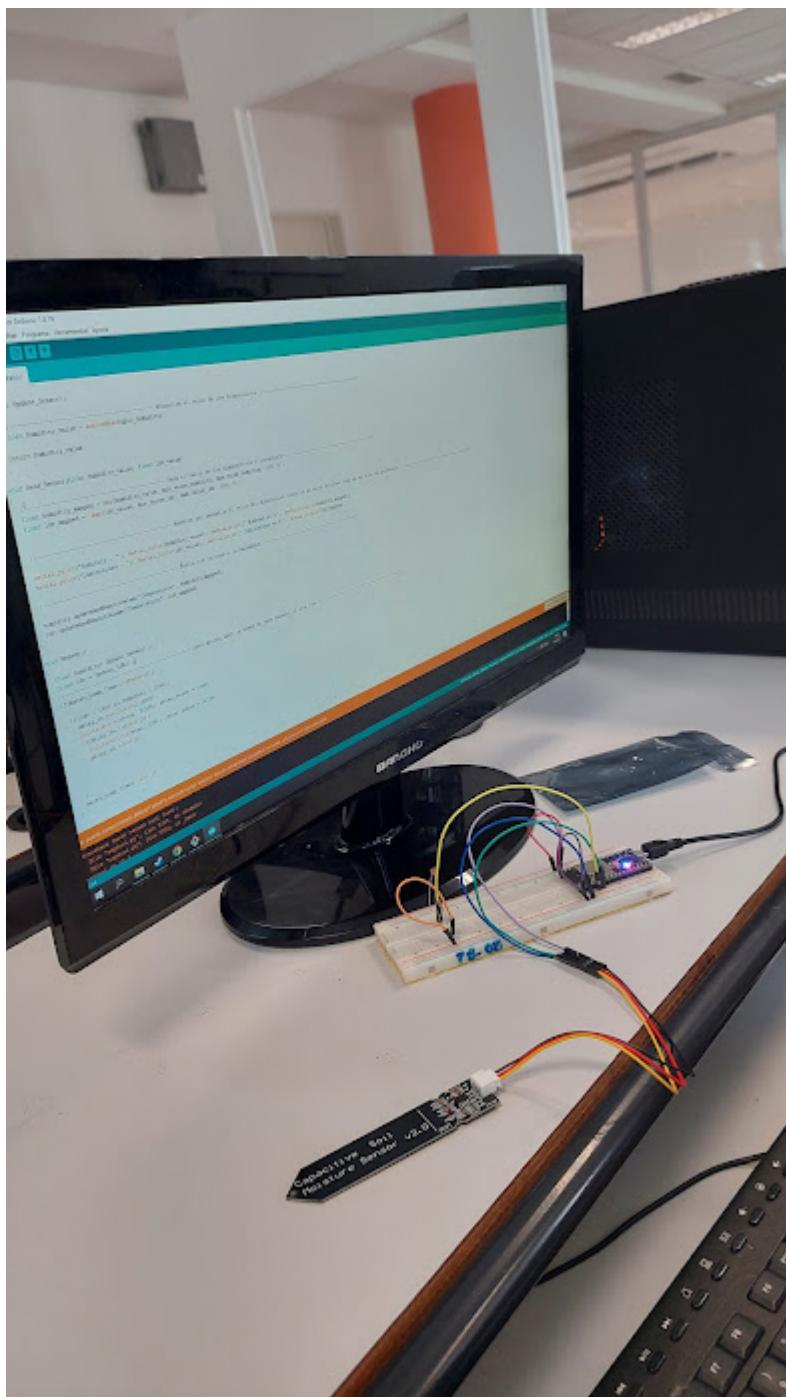


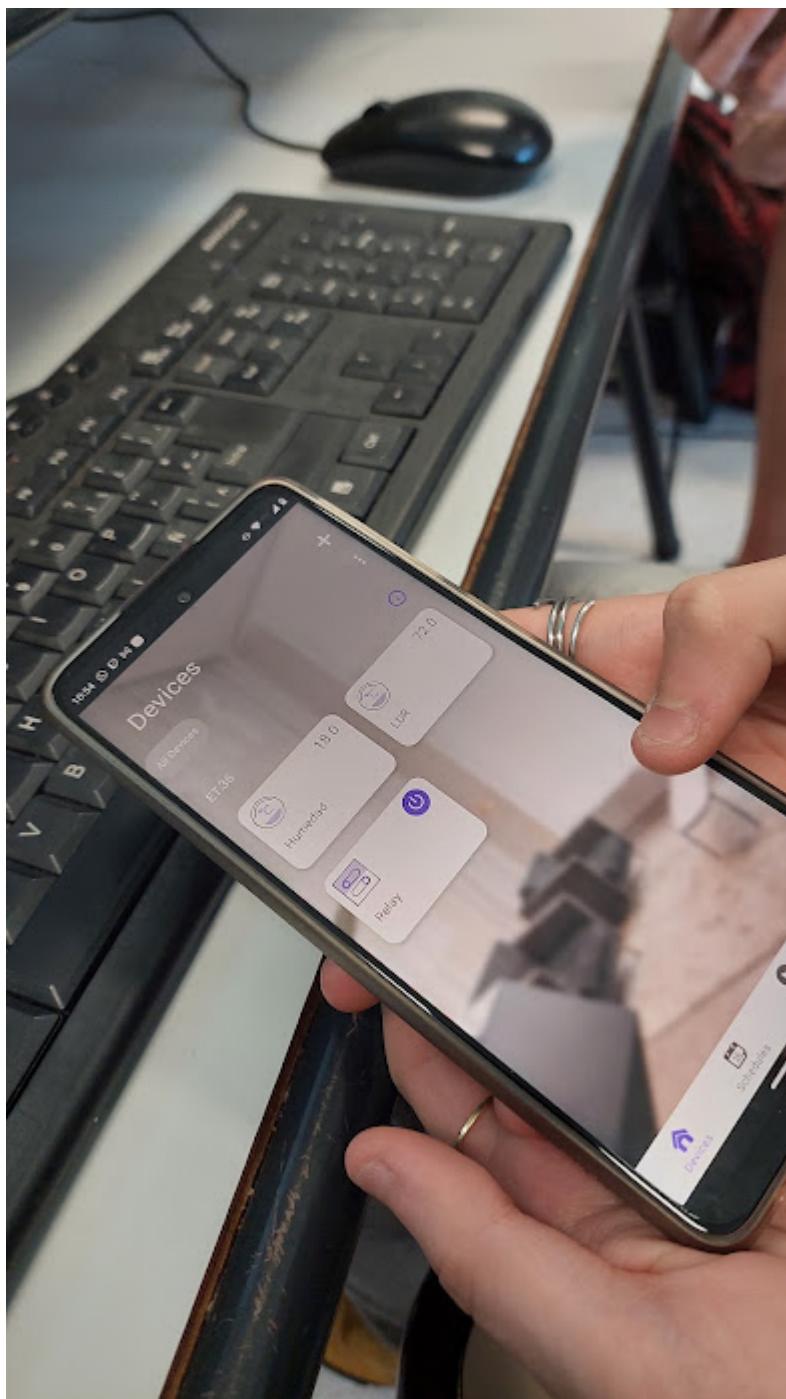
También se realizó nuevamente el diagrama general de bloques del funcionamiento de YAKU, agregando las nuevas implementaciones y eliminando las que se descartaron/se mejoraron.



07 de septiembre de 2022

Hubo una reunión de finalización de parte del grupo de ESP32, en la que se concretó parte del código final, sobre la actualización de los sensores en la aplicación y el regado efectuado por la bomba de agua.



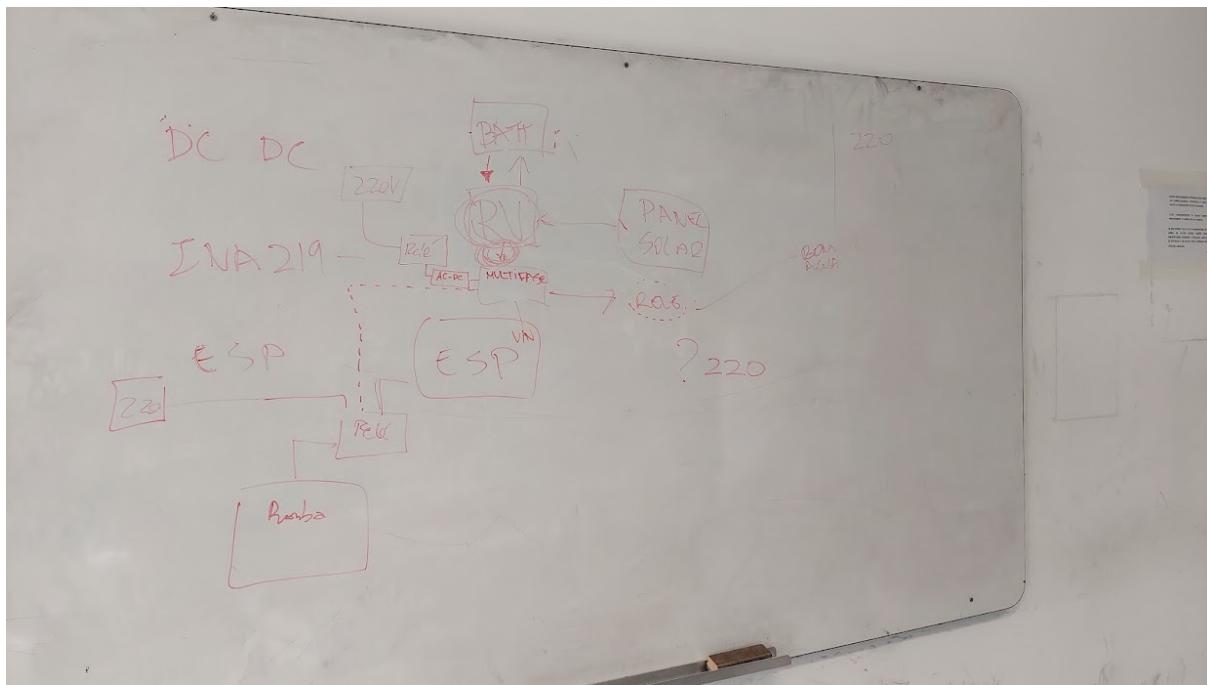


Parte del código realizado en la siguiente imagen:

```

if(Water_bomb_Timer.isReady()){
    if(analogRead(gpio_ldr) > 1000 && analogRead(gpio_humidity) < 250){
        delay_5s.setInterval(5000);
        digitalWrite(relay, HIGH); relay_state = true;
        Serial.println("REGO");
        if(delay_5s.isReady()){
            Serial.println("Dejo de regar");
            digitalWrite(relay, LOW); relay_state = false;
            delay_5s.reset();
        }
    }
    Water_bomb_Timer.reset();
}
    
```

Posteriormente, se analizaron algunas dudas acerca de cómo alimentar eléctricamente YAKU, cuando los paneles solares y la batería no pudiesen brindar la cantidad necesaria de energía.



Evaluando cómo podríamos incorporar un rol suplente de la conexión a 220.

También, se realizó un modelo en Flask, para poder enviar los datos a la página web de YAKU. (Parte del código utilizado a continuación)

```

14     app = Flask(__name__)
15
16     @app.route("/", methods=["GET", "POST"])
17     def hello_world():
18         global last_callback
19         hum = last_callback['Humedad']
20         ldr = last_callback['LDR']
21         return render_template('home.html', hum=hum, Ldr=ldr)

```

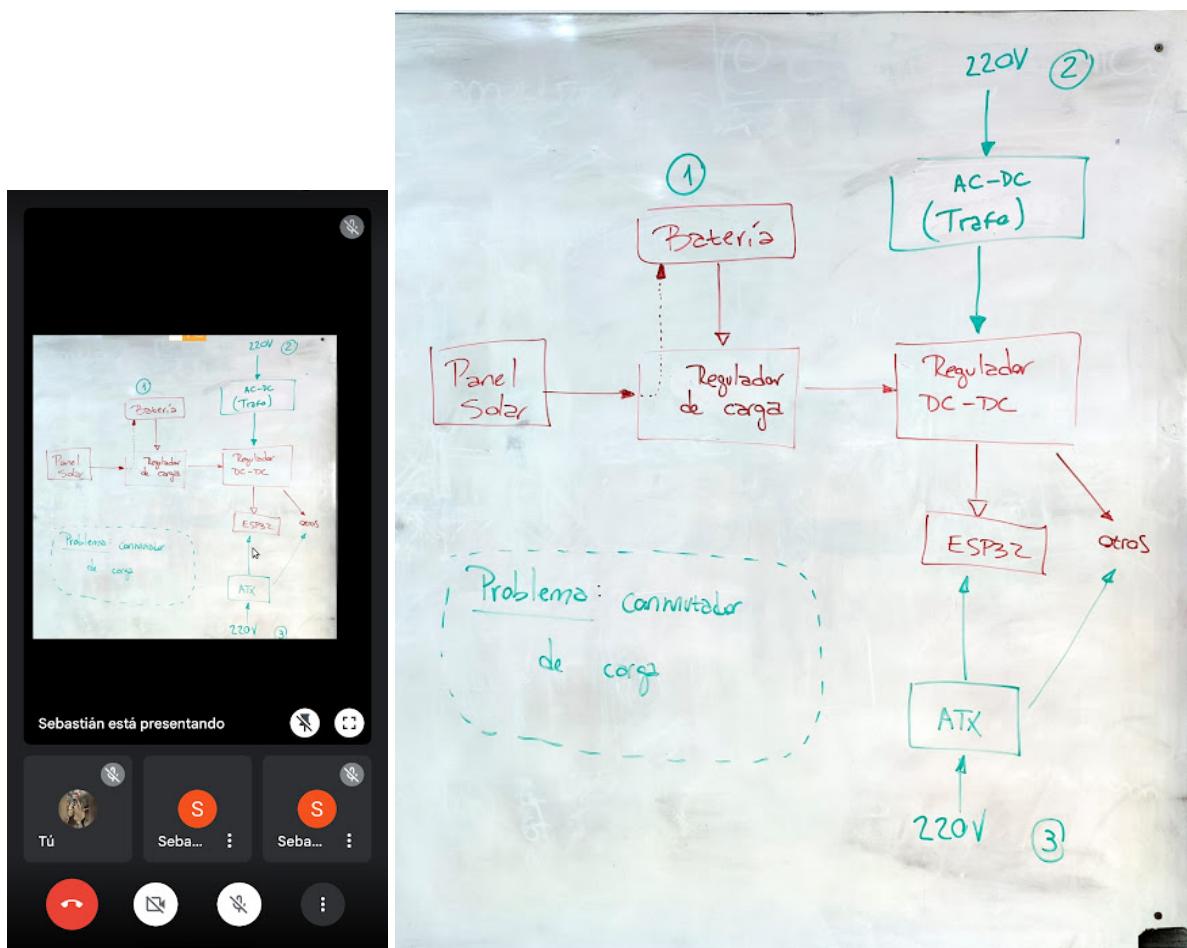
Más tarde, se terminó de unificar el código general de ESP32, faltando solo realizar unas últimas pruebas.

```

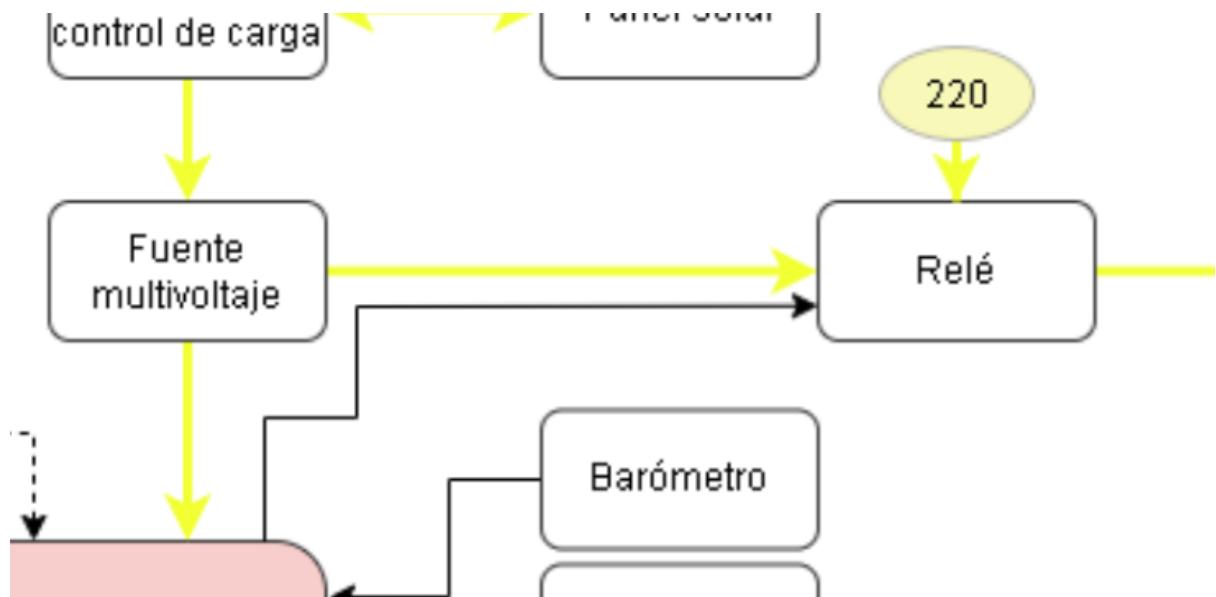
//----- Envía los valores a la Rainmaker -----
ground.updateAndReportParam("Temperature", ground_value);
ldr.updateAndReportParam("Temperature", ldr_value);
humidity.updateAndReportParam("Temperature", humidity_value); //No se mapea ya que es un porcentaje
temp.updateAndReportParam("Temperature", temp_value);
//pressure.updateAndReportParam("Temperature", P);

```

Se tuvo nuevamente una reunión virtual terminando de afianzar posibles soluciones al problema de qué hacer cuando se acabe la energía de la batería y el panel solar. Se decidieron dos alternativas, una reutilizando una fuente ATX de computadora, y otra usando un trafo ac - dc.



Se actualizó nuevamente el diagrama de bloques, agregando el relé, que antes no había sido plasmado, y que pensamos que era importante remarcar, ya que anteriormente no se notaba tan explícitamente como es que la bomba de agua se alimentaba.(Igualmente el relé es utilizado para demás partes, pero no creemos de total necesidad informar en el diagrama, ya que quedaría desprolijo)



08 de septiembre de 2022

Se sumó a la realización de la página un integrante de otro grupo, en este caso era de python, y realizó el home de la página de YAKU. Haciéndola totalmente responsive, para que se adapte a cualquier dispositivo.

YAKU

Home Page1 Page2 Page3

Regando Lazos

Regado automatico y ecologico.

Ahorra energía y sumate al regado automatizado y ecológico!

SEGUINOS EN INSTAGRAM CON EL LINK DE ACA ABAJO!

INSTAGRAM

Proyecto realizado por alumnos y docentes de la E.T. N°36 D.E. 15
"Almirante Guillermo Brown"

Dimensions: iPad Air • 820 x 1180 80% No throttling

YAKU Home Page1 Page2 Page3

Regando Lazos
Regado automatico y ecológico.

Ahorra energía y sumate al regado automatizado y ecológico!

¡SEGUINOS EN INSTAGRAM CON EL LINK DE ACA ABAJO!

[INSTAGRAM](#)

Proyecto realizado por alumnos y docentes de la ET. N°36 D.E. 15 "Almirante Guillermo Brown"



... w .divhome_container.container ...
 Styles Computed Layout ▾
 Filter show .cls + ⌂ ⓘ
 element.style {
 }
 @media screen and (min-width:
 767px){
 .home_container {
 styles.css@19
 height: 100vh;
 display: flex;
 align-items: center;
 justify-content: center;
 gap: 2rem; 5rem;
 }
 .home_container {
 styles.css@198
 position: relative;
 padding: 0 2rem;
 }
 }

Dimensions: iPhone SE • 375 x 667 100% No throttling

YAKU

Regando Lazos
Regado automatico y ecológico.

Ahorra energía y sumate al regado automatizado y ecológico!

¡SEGUINOS EN INSTAGRAM CON EL LINK DE ACA ABAJO!

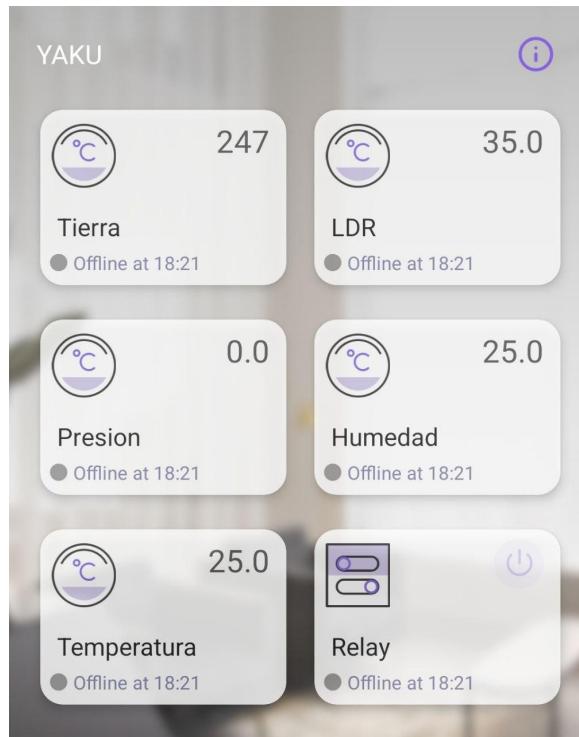
[INSTAGRAM](#)

Proyecto realizado por alumnos y



... w .divhome_container.container ...
 Styles Computed Layout ▾
 Filter show .cls + ⌂ ⓘ
 element.style {
 }
 @media (min-width:
 1024px){
 .home_container {
 styles.css@198
 position: relative;
 padding: 0 2rem;
 }
 .container {
 styles.css@69
 max-width: 1024px;
 margin: 0 auto; 1.5rem;
 }
 * {
 styles.css@42
 box-sizing: border-box;
 margin: 0;
 padding: 0;
 }
 div {
 user-agent stylesheet
 }
 }

Se logró compilar perfectamente el código unificado. Los valores de todos los sensores se subieron, solamente que se registran en valores fuera de rango, y se mandan de mala manera en el caso de la presión, temperatura y humedad general.



Al tener los sensores en el nodo de Rainmaker, se realizó el código faltante en python, para ahora sí, enviar todos los registros de los sensores a la base de datos.

```
dicc = {'Humidity': node['Humedad']['Temperature'], 'LDR': node['LDR']['Temperature'], 'Pressure' : node['Presion']['Temperature'],
'Soil Moisture': node['Tierra']['Temperature'], 'Temperature': node['Temperatura']['Temperature'],
'Relay' : node['Relay'][Power], 'Wind' : str(w.wind()['speed']) + 'm/s', 'Clouds' : str(w.clouds) + '%',
'Time': strftime("%d %b %Y, %H:%M") , 'Timestamp' : ts} #creates the dicc that we are sending to the database
```

```
_id: ObjectId("631a81c13c764b5858fafffc")
Humidity: 25
LDR: 35
Pressure: 0
Soil Moisture: 2477
Temperature: 25
Relay: false
Wind: "6.69m/s"
Clouds: "100%"
Time: "08 Sep 2022, 20:58"
Timestamp: 1662681536.996431
```

Se añadieron también a la página (actualmente provisional en blanco) con Flask.

```
return render_template('home.html', hum=last_callback['Humidity'],
ldr=last_callback['LDR'], soil_m=last_callback['Soil Moisture'],
temp=last_callback['Temperature'], press=last_callback['Pressure'])
```



Humidity: 25

LDR: 35

Soil Moisture: 2477

Temperature: 25

Pressure: 0

09 de septiembre de 2022

Se agregaron las siguientes divisiones en la página web:

The screenshot displays a mobile application interface titled "MOBILE APP". The main content area contains descriptive text about the Yaku system, mentioning its remote monitoring capabilities, relay functionality, and Google Assistant compatibility. To the right, a smartphone screen shows the "Devices" section of the app, listing various sensors and their current values: Tierra (247), LDR (35.0), Presión (0.0), Humedad (25.0), Temperatura (25.0), and Relay (Off). Below this, a summary for "ET.36" shows a temperature of 17.0 and a humidity of 64.0. The top navigation bar includes tabs for "Home", "Mobile App", and "Values".



Los valores aún son estáticos, cuando se termine la página, se le agregarán los verdaderos valores enviados desde Flask (cuyo código ya está hecho).

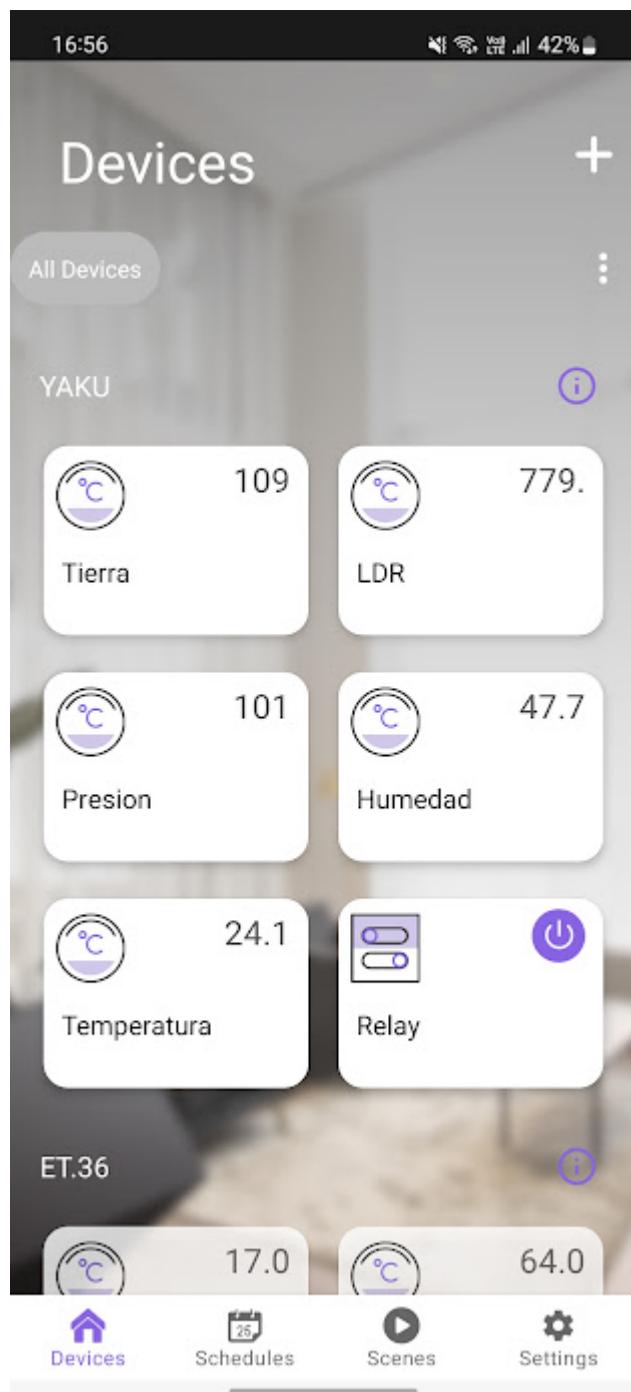
También se realizaron posts de instagram en canva. Posteriormente se publicarán.

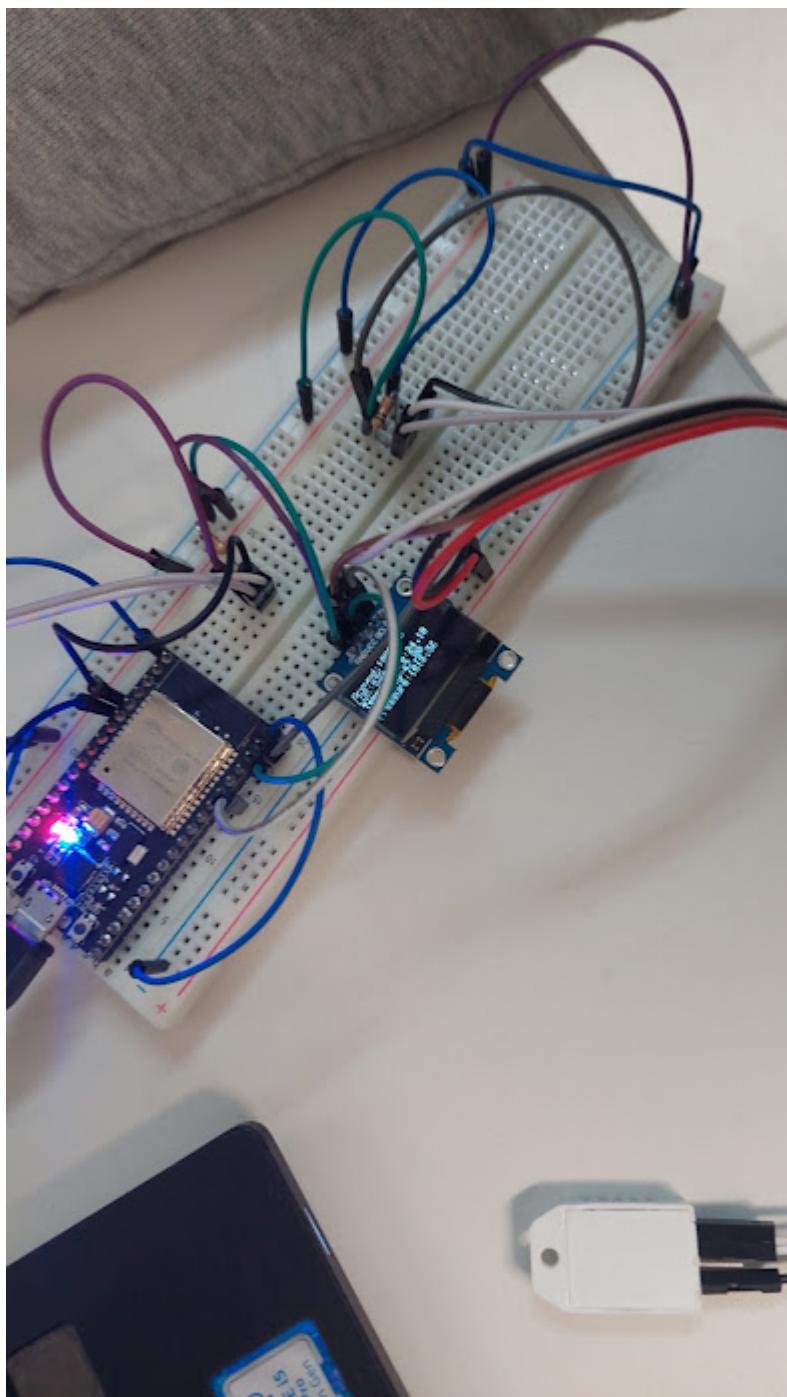
12 de septiembre de 2022

Se realizó el video para enviar la documentación y registrarnos en la segunda etapa de Feria de Ciencias.

13 de septiembre de 2022

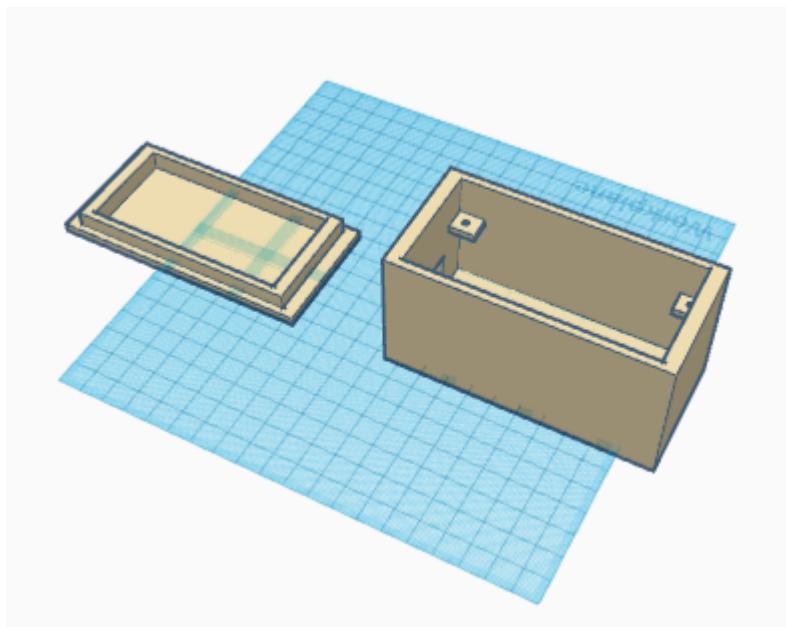
Se pudieron terminar correctamente las conexiones, que estaban fallando, y se pudieron enviar los datos de presión atmosférica (que antes faltaban), además de corregir los errores de los otros valores enviados.





13 de septiembre de 2022

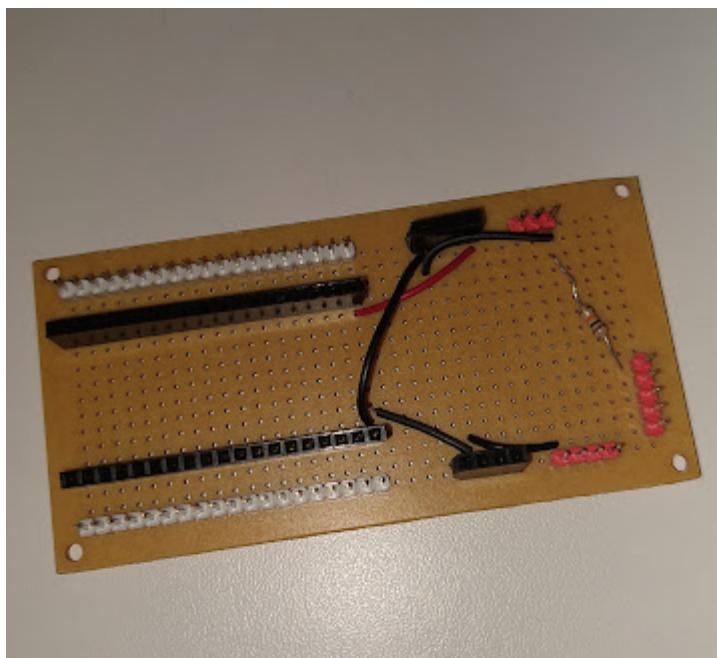
Se realizó un primer boceto de prueba, de la caja contenedora de la central de YAKU. Dicha caja va a ser impresa en 3D, por lo que su diseño se realizó en TinkerCAD.



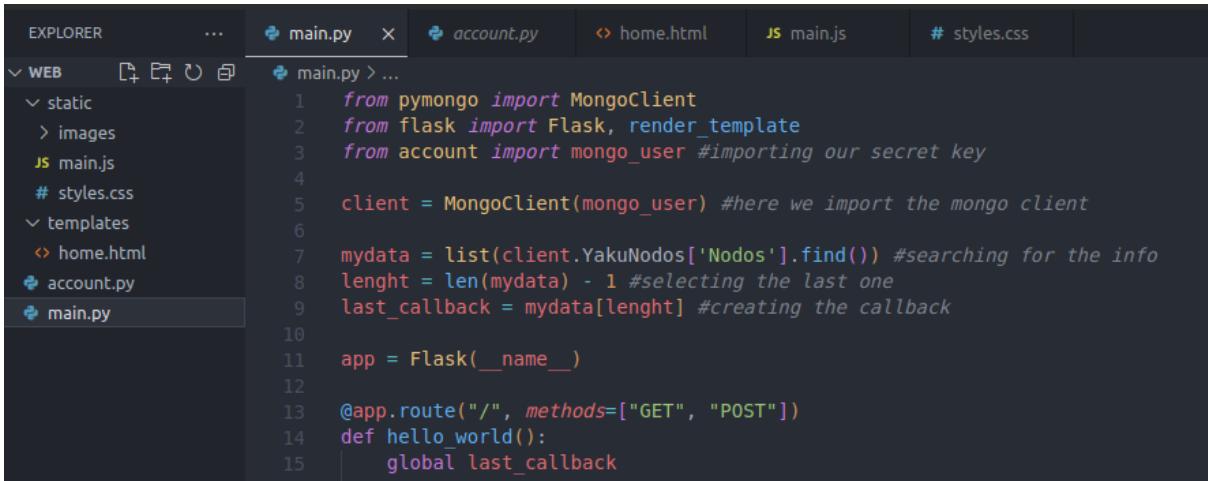
Se agregó el valor de la altitud a la aplicación.



Comenzamos a soldar las conexiones en la placa, terminando las conexiones GND y comenzando por las 5V.



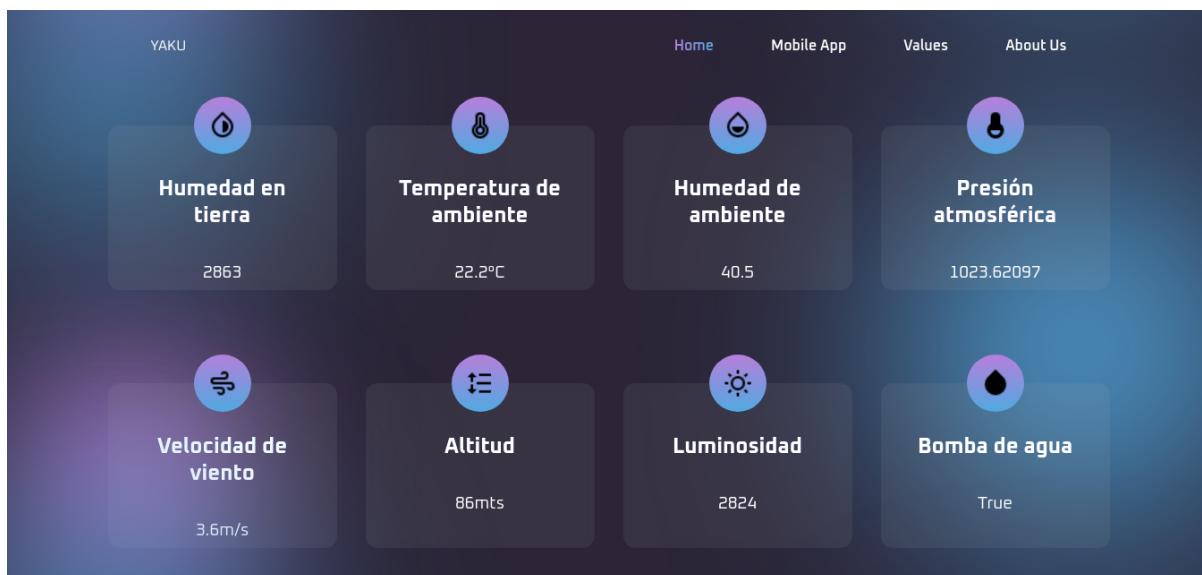
Se finalizó por completo la página web, abarcando la sección de ‘sobre nosotros’ al final, en la cual se puede acceder a instagram, github y al video, y se le agregaron valores al sector que ya estaba previamente hecho. Además, los datos mostrados, son recibidos desde la base de datos mediante Flask, ya que toda la página pasó a estar oficialmente (como se venía planeando) montada en dicho framework.


 A screenshot of a code editor showing a project structure and a Python file. The project structure on the left includes 'WEB' (static, images, main.js, styles.css), 'templates' (home.html), and 'account.py'. The 'main.py' file is open on the right, showing the following code:

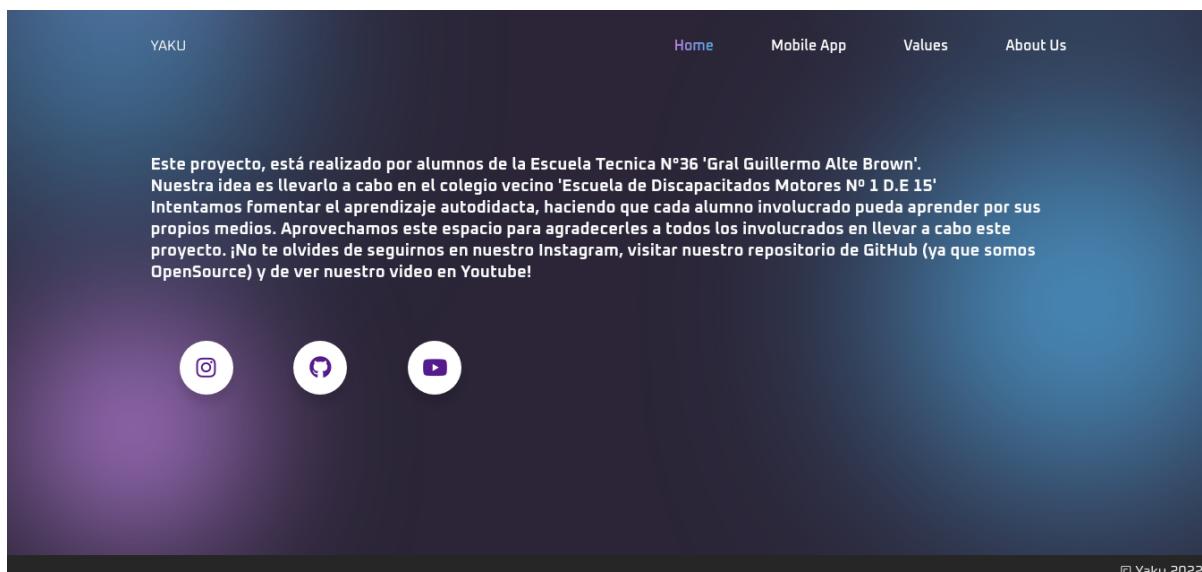

```

EXPLORER ... main.py account.py home.html main.js # styles.css
WEB static images main.js # styles.css templates home.html account.py main.py
  1 from pymongo import MongoClient
  2 from flask import Flask, render_template
  3 from account import mongo_user #importing our secret key
  4
  5 client = MongoClient(mongo_user) #here we import the mongo client
  6
  7 mydata = list(client.YakuNodos['Nodos'].find()) #searching for the info
  8 lenght = len(mydata) - 1 #selecting the last one
  9 last_callback = mydata[lenght] #creating the callback
 10
 11 app = Flask(__name__)
 12
 13 @app.route("/", methods=["GET", "POST"])
 14 def hello_world():
 15     global last_callback
  
```

(La imagen muestra la organización de carpetas utilizada para Flask, y todos los archivos por fin en conjunto.)



(La imagen muestra los valores reales recibidos desde la base de datos, y los nuevos agregados)



(Esta última imagen muestra la última página 'About Us')

Se actualizaron valores enviados a la base de datos para que sea más prolífico.

```

_id: ObjectId("63227db1b32d7e9fd8c98f0b")
Soil Moisture: 2863
LDR: 2824
Pressure: 1023.62097
Humidity: 40.5
Temperature: "22.2°C"
Height: "86mts"
Relay: true
Wind: "3.6m/s"
Clouds: "0%"
Time: "14 Sep 2022, 22:19"
Timestamp: 1663204783.790964
  
```