<https://www.canva.com/design/DAFQJIhLd04/lp-ULs34ypE1d9U45GjjQw/view?utm_content=DAFQJIhLd04&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

Thiago:

**Leo la frase**

Justamente nosotros consumimos un montón de recursos y buscamos reducir esto mimos a través de la concientización.

**Objetivo (esto lo digo mas resumidamente)**

Comenzando por los recursos energéticos, se puede dar un ejemplo con el consumo energético en TIC, el cual es muy elevado.   
Usamos un montón de luces, hasta cuando hay iluminación natural de afuera. Justamente muchas veces quedan las luces prendidas de las aulas luego de irnos, consumiendo innecesariamente.   
Luego están las computadoras de los laboratorios, que muchas veces quedan prendidas.  
Sobra mencionar los carros de carga de las laptops, los cuales están prendidos y consumiendo todo el día, hasta cuando nos vamos del colegio.   
También teniendo los dispensers de agua prendidos todo el día, calentando y enfriando agua cuando no es necesario, cuando no hay nadie en el colegio.

Luego está la calidad del aire, ya que estamos mucho tiempo en espacio cerrados, y la mayoría de las veces teniendo una calidad de aire inadecuada, afectando el rendimiento de los estudiantes.

Finalmente está el consumo y los residuos generados, lo cual hace importante entender cuándo hay que vaciar los tachos de basura. También sirve para comenzar a reciclar y producir menos residuos, así concientizamos acerca de cuántos residuos generamos y como bajar la cantidad.

**Tareas**

Bueno, en mis tareas principales primero tenía los gráficos. Básicamente yo me comunico con el hardware a través del protocolo MQTT y recopiló los datos enviados en tiempo real. Esos datos los subo a una DB y al mismo tiempo los envió al FRONT-END, para que pueda mostrar esos valores al usuario.

Luego tengo un registro, hecho principalmente para poder crear una cuenta y a través de un sistema de mails, poder recibir mails cuando los valores sean inoptimos.

Finalmente tenemos una pagina para poder borrar el registro, que sirve para dejar de recibir mails si es que el usuario no quiere recibirlos más.

**Herramientas**

Use principalmente el framework nodeJS y los siguientes módulos: (explico los módulos)

Use también MYSQL con el gestor de bases de datos de PHPMYADMIN y explico.

Gonza:

Mini presentacion:

Como era el único que se encargaba del front-end, y al no haber ningún otro diseñador ni nada parecido, tuve que encontrar la manera de, en estos 3, 4 meses, poder plasmar distintos tipos de datos, en gráficos, y otros cosas más, pero, a la vez, poder hacer un diseño para que la página no sea, digamos, insufrible de ver. Es decir, tuve que balancear entre el diseño y las funcionalidades de la página, sin realmente saber cómo.

1.Herramientas:

Para esto, termine usando, aparte de usar VS code como editor de texto, algunas librerías. La primera siendo materializecss, solo para algunos detalles para hacer la aplicación responsive. Por otro lado, al momento de hacer los gráficos tenía que usar alguna librería que me permitiera hacerlos sin tener que hacerlos desde 0 cada segundo que se cargue un dato nuevo, para eso use tanto DyGraph.js cómo Chart.js, ambas librerías que facilitan la creación de gráficos. Más adelante voy a explicar porque use estas y no otras o porque tuve que usar dos distintas librerías si, en resumen, servían para lo mismo.

2.Evolución:

Estas imágenes son la evolución de cómo fueron cambiando algunas de las pantallas de la página, por lo menos las que más fueron cambiando. Las 3 de la izquierda son como estuvieron en su primera etapa, más como una especie de boceto para ver si la idea en mente estaba bien planteada, en el medio es como se vio la mayor parte del tiempo, esto porque era el momento en donde estaba programando todas las funciones y asegurandome de que estén bienal, y al final, como terminaron cuando empecé a centrarme lo mas que podia en el diseño.

3.Gráficos.

Los gráficos de los que me tuve que encargar, que recibian datos directamente del back-end a tiempo real por medio de los sockets,el primer tipo, los de los gráficos de línea, hechos con dygraph, porque, a diferencia de a otra framework que use, esta era más sencillo de usar y podía tener grandes cantidades de datos, en caso de necesitarlos. De los más destacable que puede configurar con estos gráficos fueron hacer dos líneas, una del valor del grafico y otra que diga cuales son los valores óptimos de la misma, con la opción de desactivar esta última si molesta, y, por default, que vaya borrando los valores más viejos, con una opcion tambien de hacer que esto no pase, si es que se quiere ver los valores en intervalos de tiempo más extensos. Después está este tipo de gráfico como un círculo que en realidad empezó como un gráfico de torta, que tuve que modificarlo casi con todas las opciones que me dejaba el chart.js. Después hice que el valor se vaya actualizando y que el círculo de alrededor sea representativo de ese valor. Porque use chart.js para este gráfico? Por es mismo de que me dejaba mucha variedad para hacer lo que quería.

4. Responsive design:

Por último, también quiero mencionar de que la pagina hecha es responsive, osea que todos los elementos se acomodan para que se vea bien en cualquier pantalla. En parte, use materialize para las interfaces que tengas muchos elementos, como en la imagen de la izquierda, que solo hace que las cosas que normalmente están una al lado de otra, las acomoda una abajo de otra. Por otra parte, también había algunas pantallas que tuve que diseñar una versión para pantallas más chicas que se vayan achicando a agrandando solas según la resolución de la pantalla, como en la imagen de la derecha.

Sofi: adafruit y electricidad.

¿Qué es el adafruit? Cuando hicimos funcionar el primer sensor de peso teníamos que mandar los datos, para eso usamos un bróker, Adafruit es la página web donde nosotros publicamos valores tomados, para que luego el back se pueda suscribir a este broker y utilizar estos datos para mandarlos a la página web.

¿Cómo funciona la conexión desde el sensor a la web? Con este código que subimos al nodemcu, la idea es que a través de wifi se conecte al adafruit. Para eso necesitamos que en el código figure la id y el password de la red que nos vamos a conectar. Así se pasan los datos del sensor al Adafruit.

Sensor de energía

Nuestro último sensor, el SCT-013 está capacitado para medir la intensidad de la corriente que circula. Nuestra idea principal era adaptarlo a una térmica y enchufar cosas para que pueda medir el consumo total. Pero lo terminamos adaptando a una zapatilla. Este sensor está compuesto por una pinza (el núcleo) que se debe conectar al cable fase, cuanto más corriente pase más grande es el campo magnético.

Como ven hay un circuito integrado, LM358 trabaja con polaridad positiva que es la que nos interesa, eliminando la parte negativa de la señal que nos es irrelevante.

Santi: sensor de aire y sensor de peso.

Sensor de peso, ¿cómo toma los datos? No me voy a meter en el tema físico de cómo toma los datos, pero cuando introducimos peso sobre un extremo de la celda de carga, se envían los datos de cuánto pesa ese objeto.. Se conecta a la placa a través de un adaptador de la información, el HX711. La idea es crear una base para colocar una bolsa de basura sobre la celda, asi omar cuantos gramos están siendo desperdiciados en cada bolsa de basura.

Calidad de aire, este sensor toma los datos midiendo el aire para ver qué tan limpio y respirable para el humano es. se mide en ica, qué es una unidad qué mide justamente la calidad del aire. Normalmente en un ambiente normal como la escuela los datos no suelen subir a 100 qué ya sería bastante malo, pero es una forma de medir qué tan ventilada esta una zona, lo ideal sería no subir de los 35 pero a veces es complicado llegar a eso.