

# Documentación

Integrantes

Estación Meteorológica IOT

Objetivos

Bocetos del prototipo

Construcción del prototipo

API

Requisitos para el funcionamiento

Preparando el entorno

Iniciando el funcionamiento de la API

Diagrama

Interfaces

Pantalla de interfaz web

Interfaz en processing

Capas de framework IoT

Product Infrastructure

Sensors

Connectivity

Analytic

Smart App

Enlace del repositorio.

Enlace de video del prototipo.

Prueba en exterior

Prueba en interior

## Integrantes

- Rony Ormandy Ortíz Alvarez - 201807328
- Tomás Alexander Morales Saquic - 201900364
- Jhonatan Josué Tzunún Yax - 201900831
- Elder Fernando Andrade - 201700858
- Nery Oswaldo Jimenez Contreras - 201700381

# Estación Meteorológica IOT

Se solicita la creación de un dispositivo meteorológico inteligente para pronosticar y tomar las debidas precauciones acorde a las condiciones meteorológicas

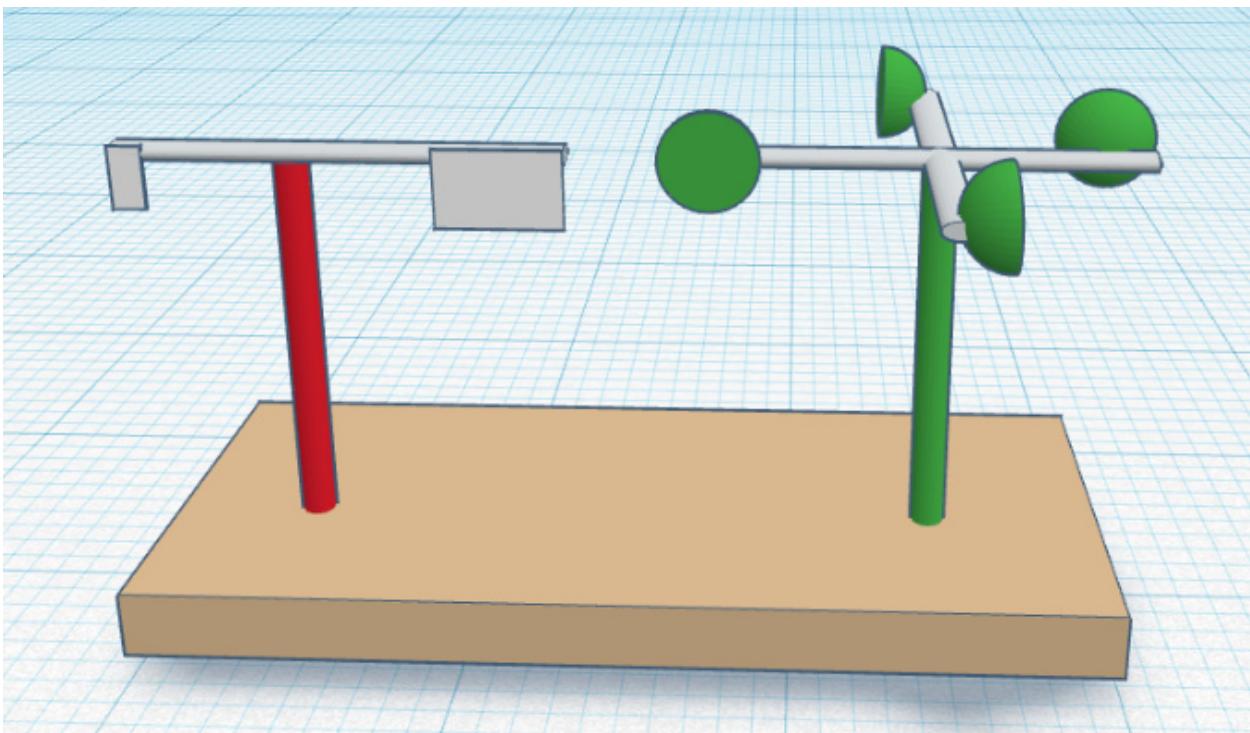
Como ya es de costumbre hoy día, los datos generados y almacenados por cualquier dispositivo deben poder ser monitoreados, visibles y de fácil comprensión para cualquier tipo de usuario, desde el más experto hasta el más inexperto, por ello se decidió también integrar una interfaz que permita interpretar las lecturas de una forma gráfica y animada.

El proyecto tiene como objetivo diseñar un dispositivo para medir y registrar periódicamente diversas variables meteorológicas como temperatura, humedad, velocidad del viento, dirección del viento y presión barométrica. El proyecto también involucra el desarrollo de una aplicación en procesamiento que proporcione un tablero en tiempo real para visualizar e interpretar las lecturas de datos de manera gráfica y animada. El dispositivo podrá medir y almacenar la temperatura ambiente, la humedad relativa, la humedad absoluta, la velocidad del viento, la dirección del viento y la presión barométrica, que se enviarán a una computadora a través de un cable y se almacenarán en una base de datos. La aplicación de Procesamiento mostrará los datos en un tablero, incluidos gráficos que muestran el comportamiento de las variables meteorológicas a lo largo del tiempo. El proyecto incluirá un experimento para calificar la precisión del dispositivo, incluido un experimento para aumentar la temperatura y la humedad ambiental, así como otro experimento para medir la velocidad y dirección del viento.

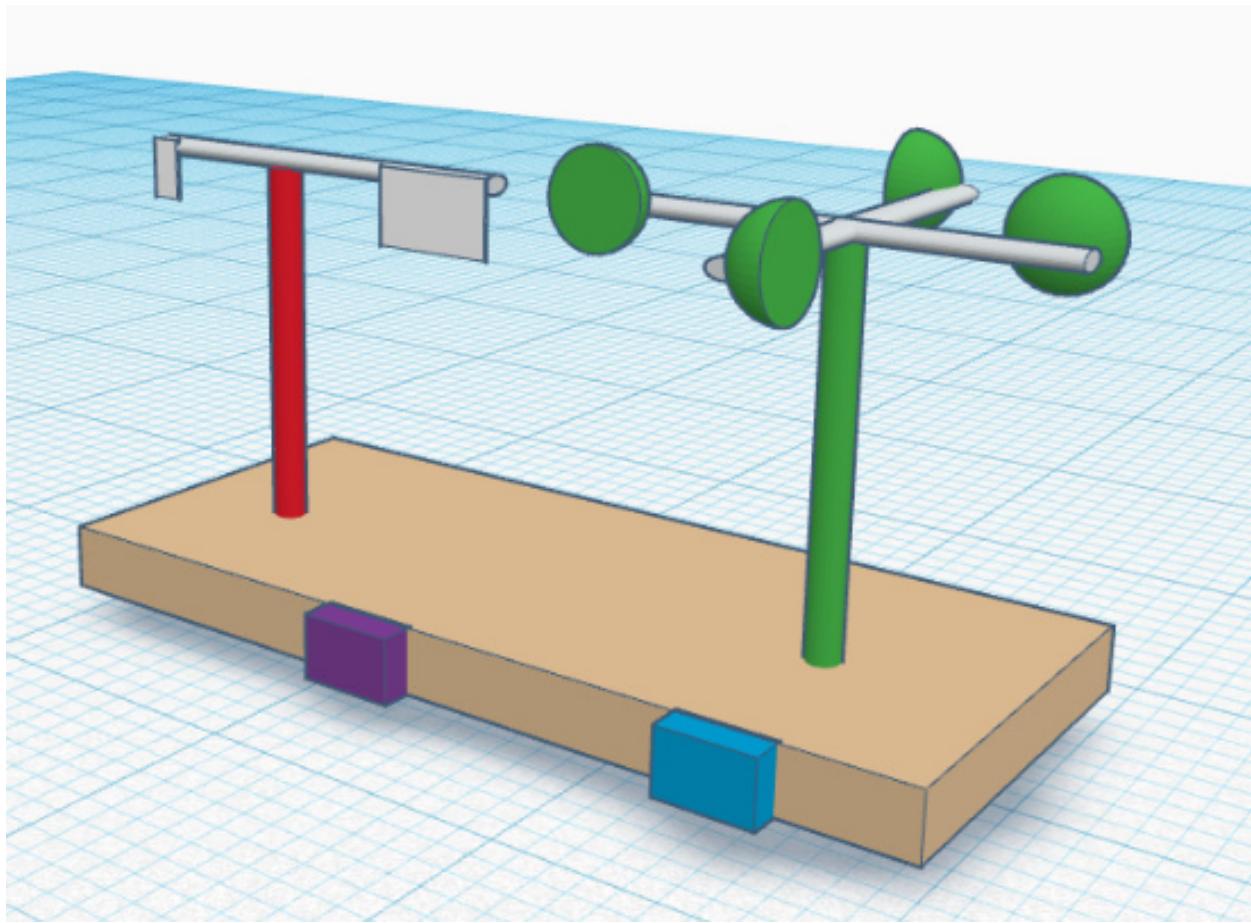
## Objetivos

- Diseñar un dispositivo destinado a medir y registrar regularmente, diversas variables meteorológicas.
- Implementar una aplicación en Processing que permita visualizar mediciones y observaciones puntuales de los diferentes parámetros meteorológicos.
- Aprender a desarrollar una manera correcta de visualización de datos mediante la implementación del framework de IoT.

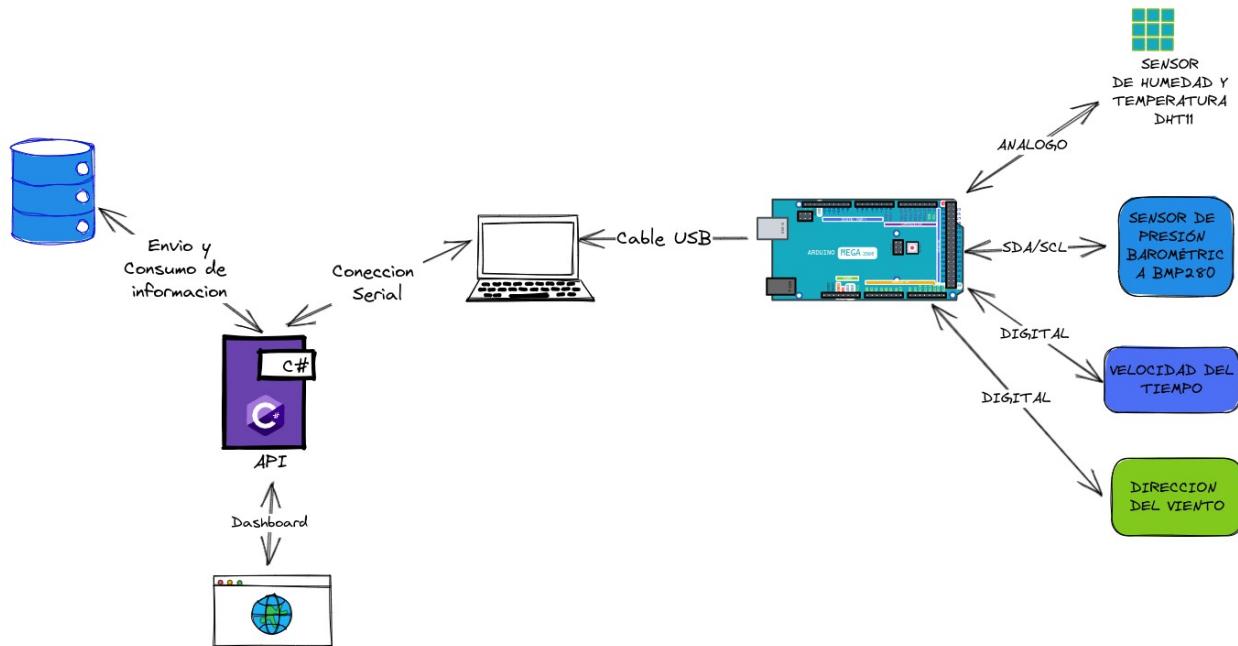
## Bocetos del prototipo



Modelo del prototipo en 3D para la toma de datos.

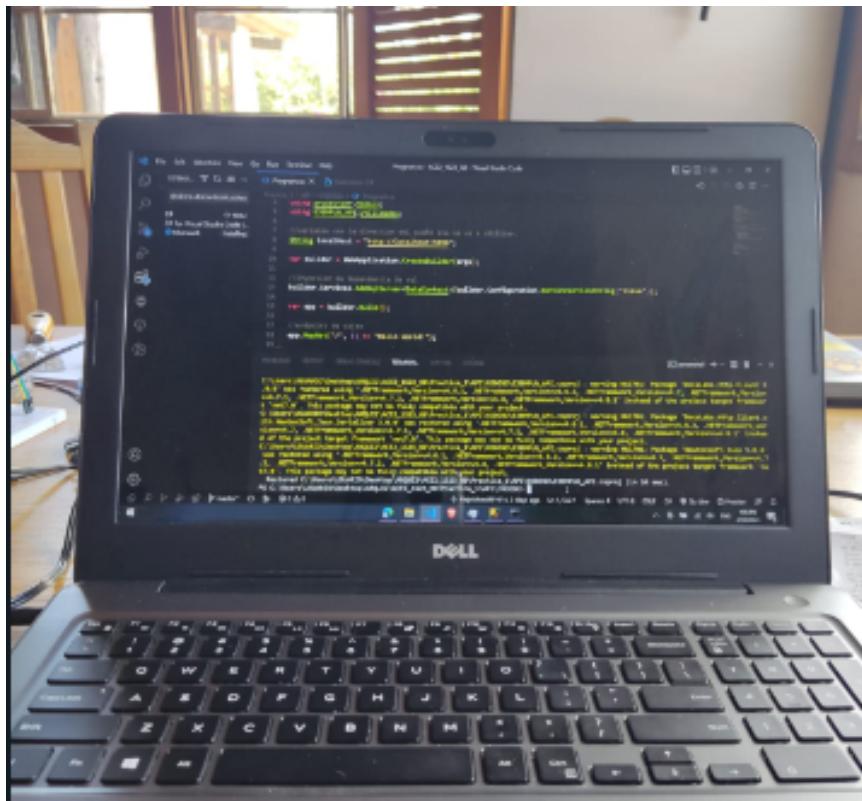


Modelo del prototipo en 3D para la toma de datos.

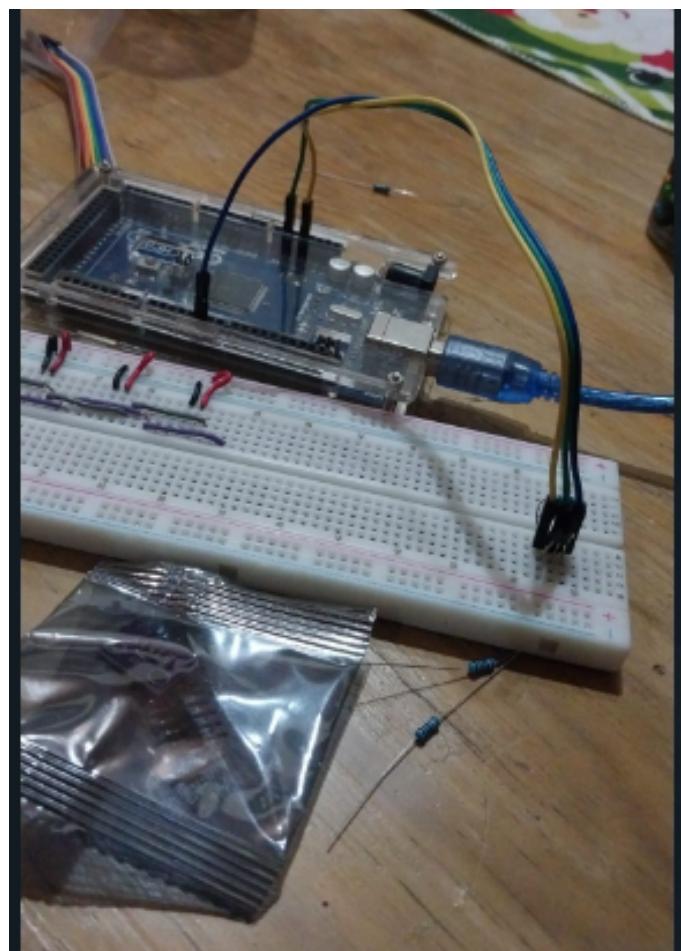


Modelo de trabajo para el funcionamiento del prototipo.

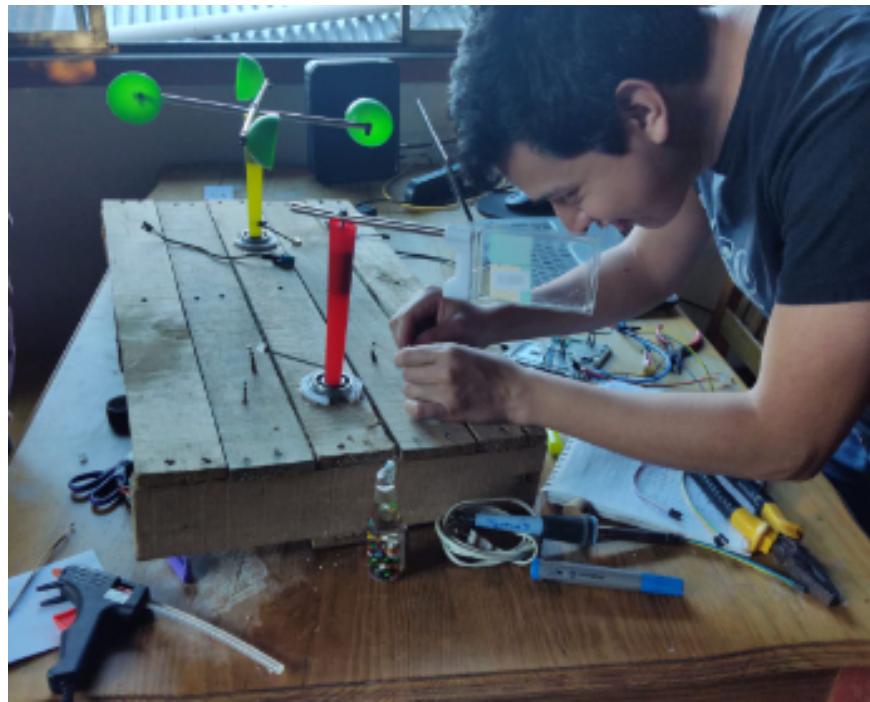
# Construcción del prototipo



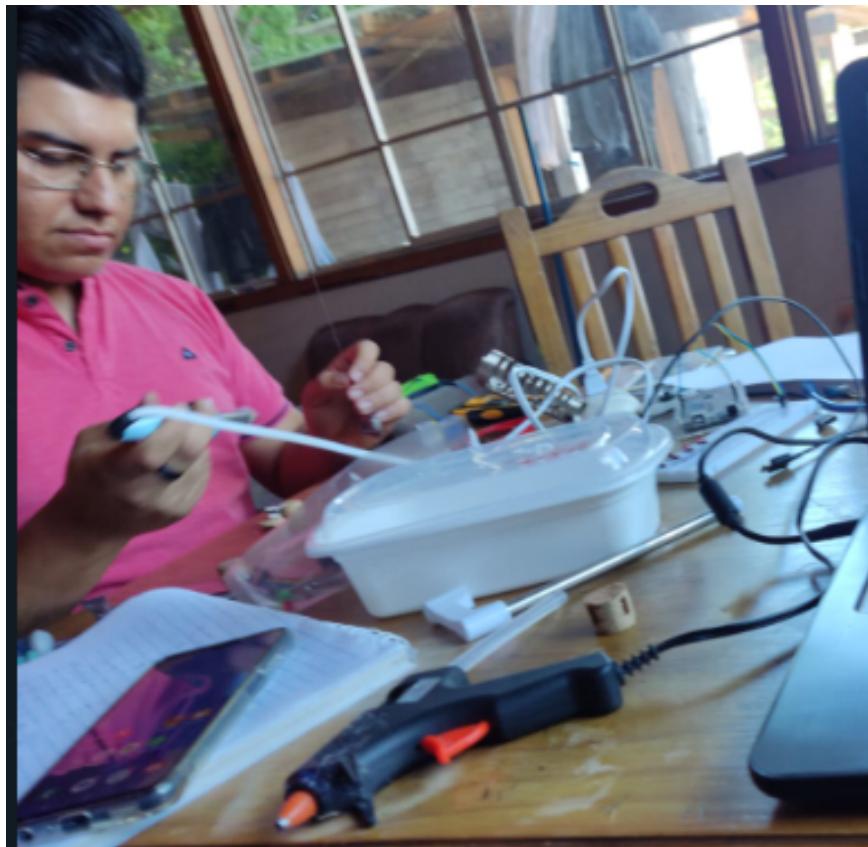
Proceso de codificación



Ensamblado en protoboard



Armado de prototipo



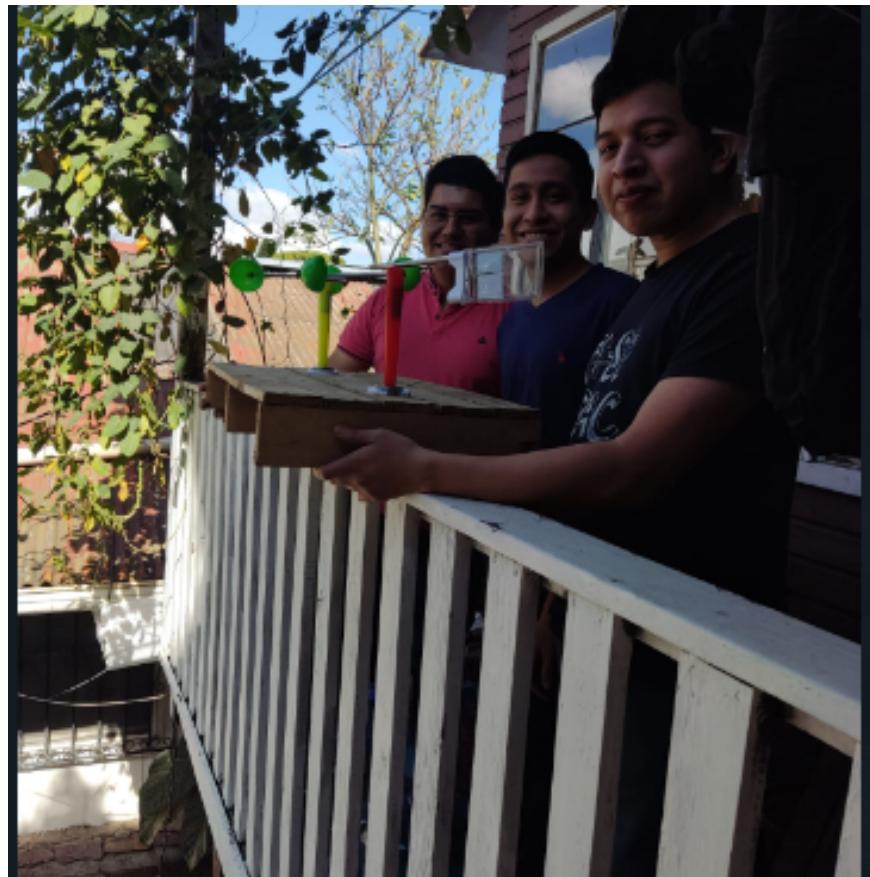
Soldadura de sensores.



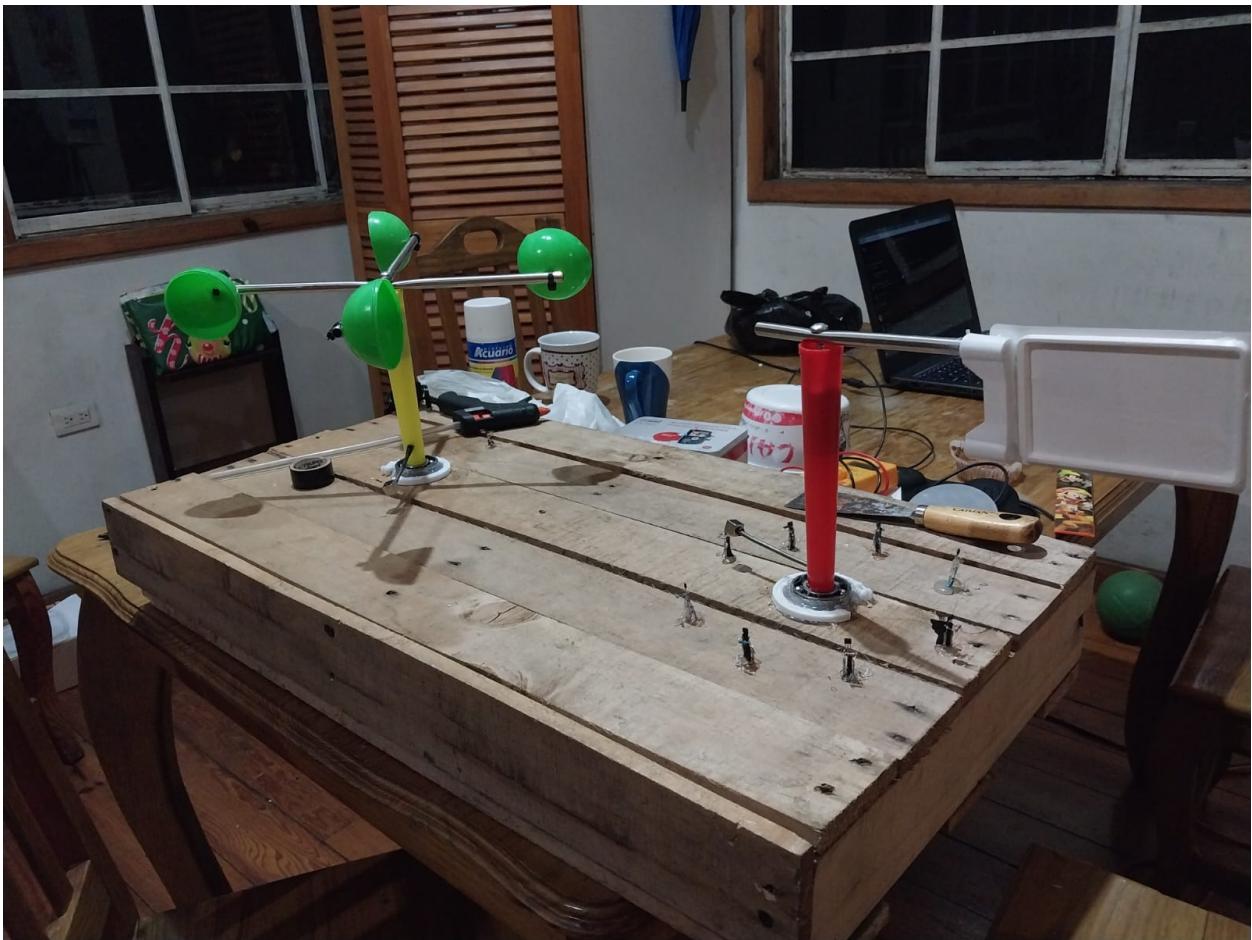
Test de toma de datos.



Miembros del equipo en la toma de datos.



Miembros del equipo en la toma de datos.



Prototipo con sensores.



Toma de datos del prototipo.

## API

Para poder manejar los datos que se obtengan del dispositivo se decidió trabajar con una API (Interfaz de programación de aplicaciones) por la cual se pueda realizar la comunicación del dispositivo hacia una base de datos mediante comunicación serial y poder mantener la información obtenida.

La API se desarrollo en el lenguaje **C#** utilizando la tecnología del framework **.NET** utilizando el template de Minimal API, para la base de datos se utilizo SQL Server en dicha base de datos se almacenaran todos los datos leídos por el dispositivo, dichos datos se recolectaran mediante sensores, para la comunicación entre la base de datos y el framework se utilizo entity framework el cual es una *ORM*.

## Requisitos para el funcionamiento

- Descargar e instalar el SDK de .NET
- Descargar e instalar VSC para la manipulación del programa
- Postman o cualquier programa para manipular API's
- Descargar e instalar SQL Server 2019
- Descargar e instalar SQL Server Management Studio (SSMS)

## Preparando el entorno

Una vez tengamos todo instalado necesitaremos realizar una descarga de las dependencias del proyecto estas dependencias podemos instalarlas utilizando el siguiente comando en la terminal, tenemos que estar en la ubicación del proyecto.

```
dotnet restore
```

Si esto no funciona también podemos utilizar el archivo llamado "**dependencias.txt**" en el cual encontraremos los comandos para poder instalar las dependencias de nuestro proyecto.

```
//para EF
dotnet add package Microsoft.EntityFrameworkCore --version 7.0.2

//para conexion de sql server
dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 7.0.2

//para el puerto serial
dotnet add package System.IO.Ports --version 7.0.0

//newton soft para json
dotnet add package DocaLabs.Http.Client.with.NewtonSoft.Json.Serializer --version 3.0.0

//esto para la instalacion de dependencias
dotnet restore
```

# Iniciando el funcionamiento de la API

Para hacer que nuestra API comience a funcionar bastara con utilizar el comando

```
dotnet run
```

## Diagrama

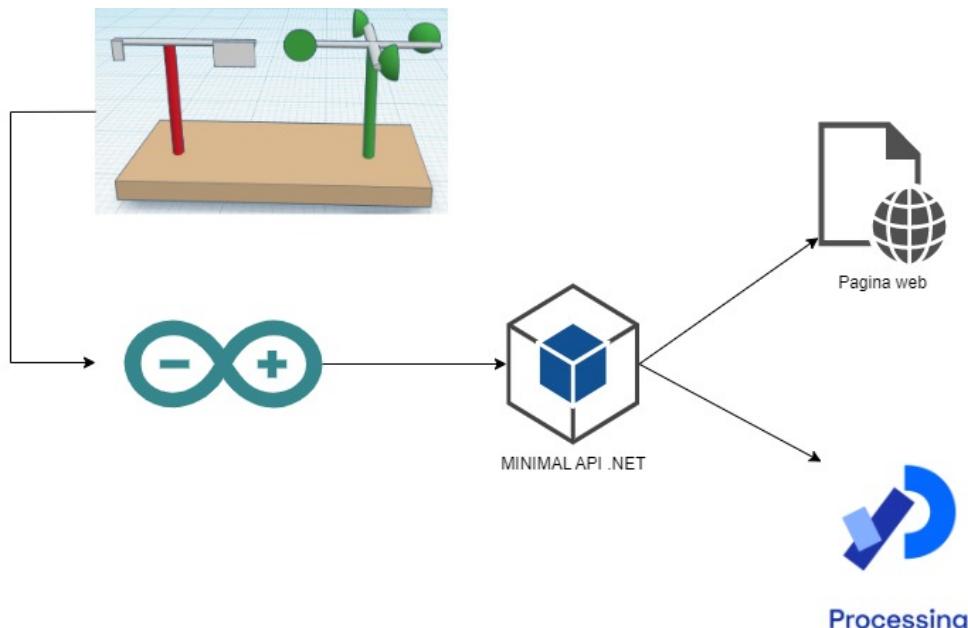
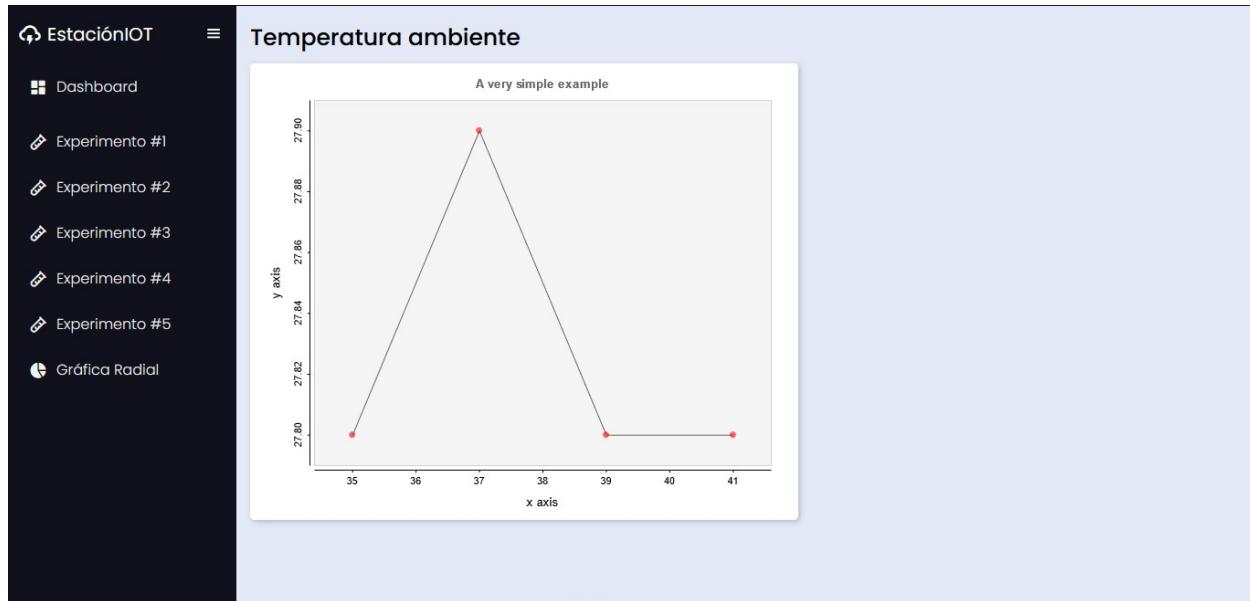


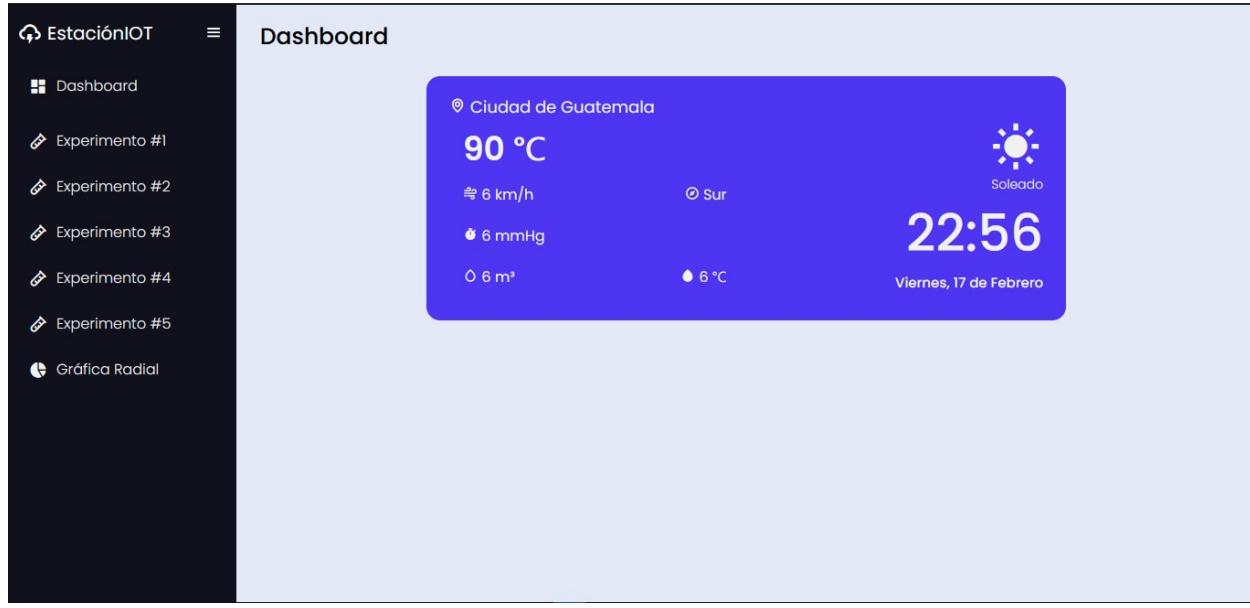
Diagrama que ejemplifica la comunicación entre los componentes

## Interfaces

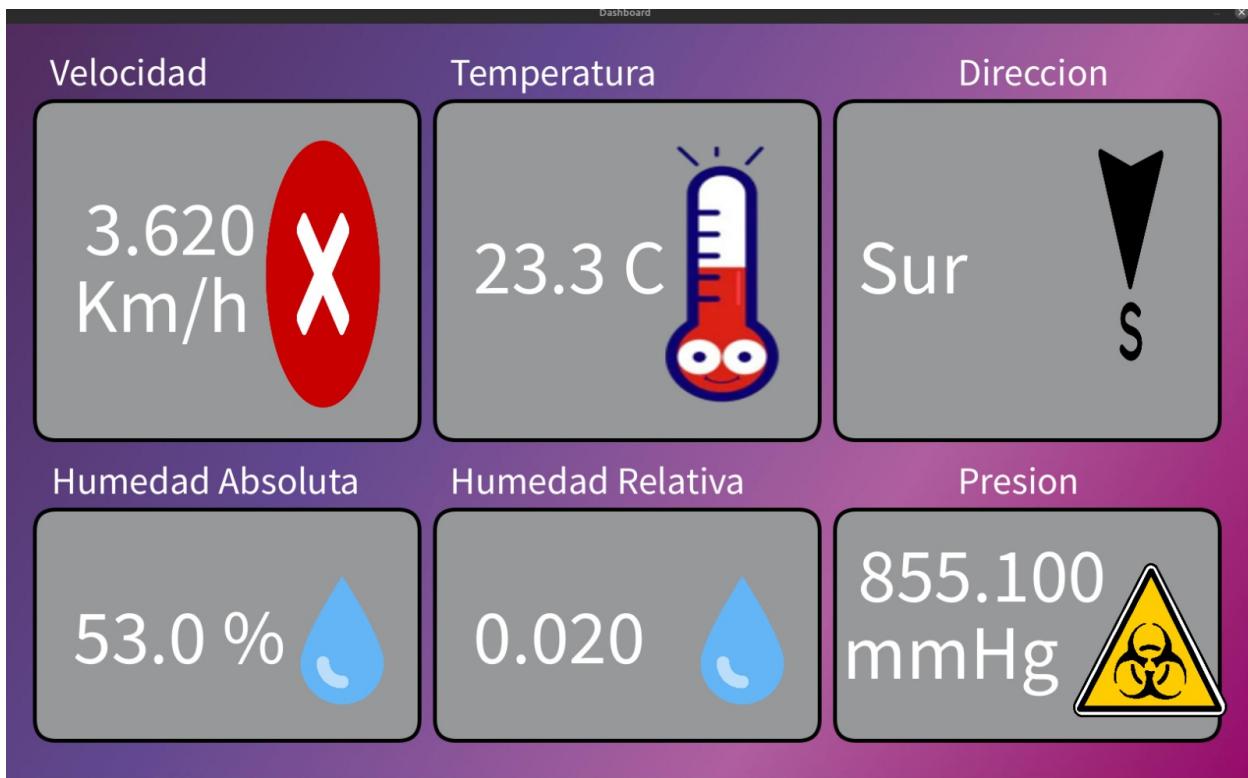
### Pantalla de interfaz web



Ejemplo de DashBoard de la aplicación para ver los datos recolectados.



## Interfaz en processing



Ejemplo de DashBoard en tiempo real realizado en processing,

Processing es un lenguaje de programación y un entorno de desarrollo integrado (IDE) diseñado para crear visualizaciones interactivas, gráficos y animaciones en 2D y 3D. Processing es un software de código abierto y está disponible para descargar de forma gratuita en su sitio web. También cuenta con una comunidad activa de usuarios que comparten sus proyectos y recursos en línea, lo que lo convierte en una herramienta popular para artistas, diseñadores, educadores y programadores. Processing también ha sido utilizado en numerosos proyectos académicos, de investigación y de arte y ha sido exhibido en museos y galerías en todo el mundo.

## Capas de framework IoT

### Product Infrastructure

La infraestructura de la estación meteorológica distribuye dos objetos mecánicos para el reconocimiento del viento sobre una plataforma que protege otros dos sensores y el

microcontrolador que procesará la información obtenida durante el experimento.

## Sensors

Se requiere la obtención de las magnitudes de la temperatura exterior y la humedad del ambiente, la cual proviene de un sensor de humedad y temperatura para arduino, DHT11.

Asimismo, se estará utilizando un sensor de presión barométrica BMP280.

Para obtener las mediciones de la velocidad del viento y la dirección, se implementarán sensores de efecto hall en conjunto con una veleta y un anemómetro.

## Connectivity

Los datos que hayan obtenido los diferentes sensores serán almacenados en un archivo JSON por nuestro microcontrolador ARDUINO, el cuál está conectado vía serial a una computadora donde serán procesados estos valores.

## Analytic

El análisis de los datos proporcionados del arduino, serán consumidos por una API. Ésta dirigirá la información hacia el dashboard principal que nos mostrará las magnitudes en tiempo real, y también hacia processing, donde nos retornará información a largo plazo de los cambios climáticos reconocidos por nuestro prototipo.

## Smart App

Para la visualización de los datos recabados se cuenta con un dashboard que muestra las magnitudes de la velocidad del viento y su dirección, la temperatura del ambiente así como la humedad y la presión barométrica. Este dashboard despliega las mediciones en tiempo real para cada magnitud, y contiene un apartado donde muestra gráficas de las magnitudes en función del tiempo a manera de tener un historial del comportamiento del clima donde se esté utilizando el prototipo.

## Enlace del repositorio.

```
Console.WriteLine("Repositorio Grupo Cero");
```

## **Enlace de video del prototipo.**

### **Prueba en exterior**

```
Console.WriteLine("Video");
```

### **Prueba en interior**

```
Console.WriteLine("Video");
```