



# Pomodoro Smart Connected Design Framework

@RONY ORMANDY ORTÍZ ALVAREZ

Carnet: 201807328

@TOMÁS ALEXANDER MORALES SAQUIC

Carnet: 201900364

@JHONATAN JOSUÉ TZUNÚN YAX

Carnet: 201900831

Nery Oswaldo Jiménez Contreras

Carnet: 201700381

Elder Fernando Andrade Foronda

Carnet: 201700858

## Introducción

Una de las técnicas de administración del tiempo más eficientes es el pomodoro. Esta técnica divide el tiempo del trabajo y de descanso en cuatro intervalos cada uno, siendo el más común 25 min de trabajo y 5 min de descanso, por cuatro ciclos.

La propuesta de Pomodoro Smart Connected es brindarle al usuario un dispositivo que le permita concentrarse en sus actividades y dejar por un lado la procrastinación. Este dispositivo IoT es capaz de generar reportes de rendimiento laboral, mostrar penalizaciones a los parámetros de cada pomodoro y replicarlas en un sistema PWA vía móvil.

## Infraestructura del producto

### Listado de materiales físicos

- Madera (case)
- Cables
- Micro-controlador de tamaño pequeño recomendable.
- Modulo Wifi
- 2 push botton
- Dimmer
- LCD

- Buzer
- Tornillos

## Listado de materiales digitales

- APP

## Sensores

### Sensor de asiento (Push botton)

Tamaño	Lectura sensor	Instalación	Rango de medición	Unidad Métrica
6.2 mm	Continuidad	Dentro de asiento para silla.		true/false

#### Módulo Inalámbrico Wifi ESP8266 Esp-01 40

Sensor de peso

bateria 9v

adaptador para arduino 9v

Link:

<https://uelectronics.com/producto/sensor-ir-rebote-smd-qre1113gr/>

Imagen:

Precio: Q 30.00



## Conectividad

## ENTORNO Y BOCETOS

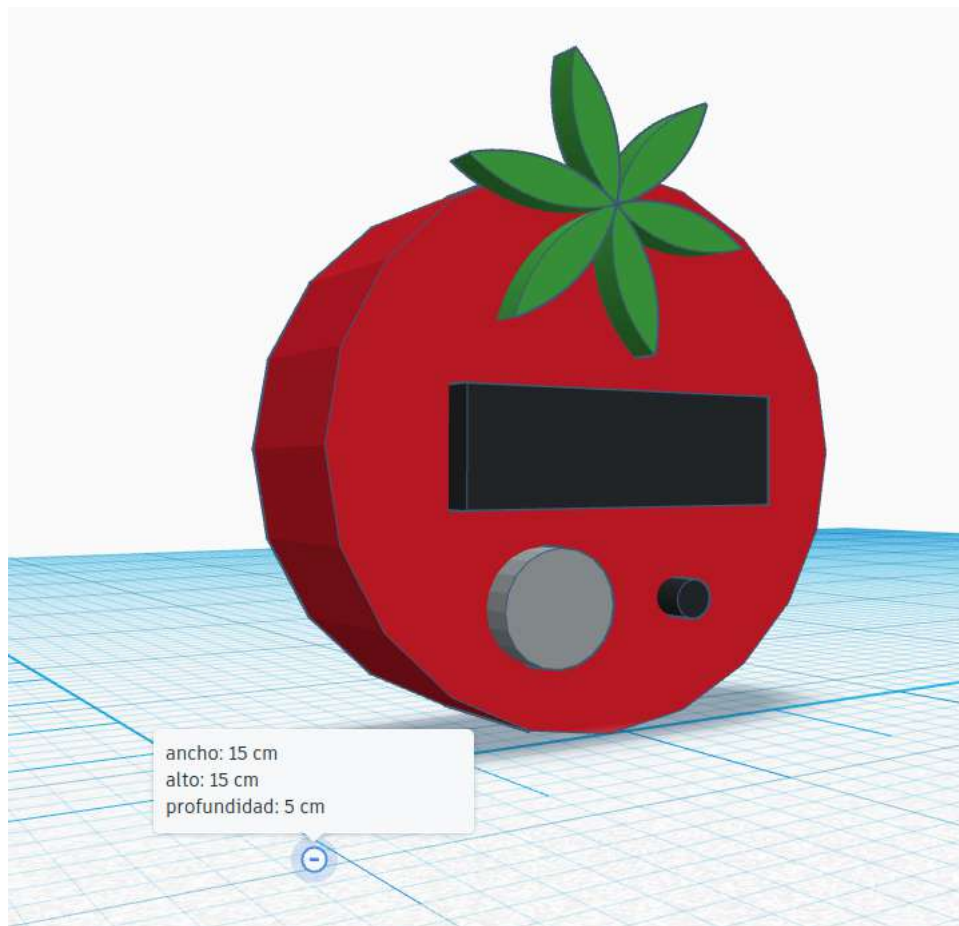
### Entorno del objeto

Interiores, tales como:

- Casa
- Oficina

## Tamaño del objeto

La estructura del pomodoro será similar a un tomate. Este tendrá un dimmer para regular el tiempo de trabajo y descanso, un botón de reset, y una pantalla LCD que nos mostrará el tiempo restante.



Esquema de modelo del producto

Al pomodoro se conectará un cojín vía cable, el cual será nuestro sensor de asiento ocupado.

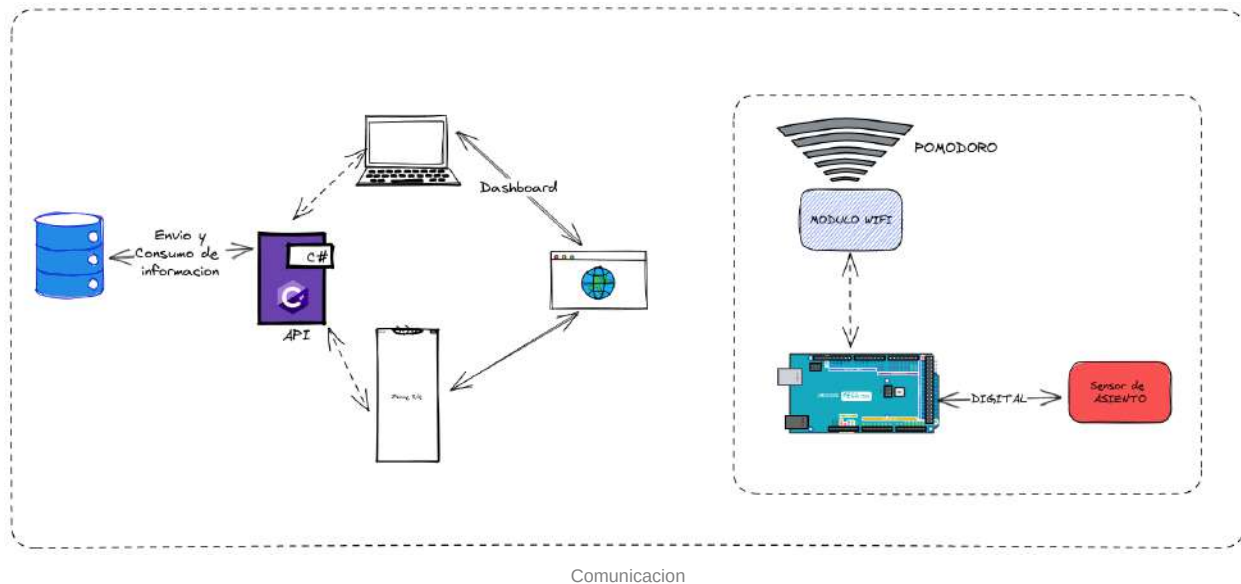
### ¿Cómo voy a enviar mis datos al servidor?

Los datos se enviarán de manera dependiendo el estado nuestro sensor indiscreto este será el encargado de funcionar como un actuador y dar inicio a la parte de envío de datos a la API.

La API tendrá un endpoint de tipo POST por el cual nosotros podremos enviar datos del estado actual del sistema, es decir la hora de la toma de datos, el estado del usuario, el número de pomodoro actual, indicador de descanso como otros puntos que nos sirvan para el análisis.

La API estara guardando los registros de nuestro sistema en una base de datos la cual consumiremos usando una ORM, entity framework, esta ORM nos ayudara a poder manejar nuestros registros como si fueran instancias de clases lo cual es mas practico para el desarrollador, pues podra utilizar el conocimiento de POO ( programacion orientada a objetos) que tiene para consumir la base de datos y poder mostrar los resultados en el respectivo dashboard.

## Gráfica de modelo de comunicación



## Analítica

### Base de datos

#### Métricas

- Estado del usuario ( basándonos en el estado del sensor )
- Tiempo de registro
- Descanso?
- Fecha
- Numero de pomodoro

### Para el modelo de 4 espacios:

GUID	Descanso	#Pomodoro	Fecha	Mes	Dia	Hora	Minuto
1	True	1	17:02:2023:11:20	02	17	11	20
2	False		17:02:2023:11:21	02	17	11	21
3	True	2	17:02:2023:11:22	02	17	11	22
4	True	3	17:02:2023:11:25	02	17	11	25

### Análisis Descriptivo

En que estado esta el sensor?

### Análisis de diagnostico

Sumatoria de penalizacion por no descansar.

Estamos en un descanso?  
Nos toca un pomodoro?  
Ya terminamos descanso?  
Ya terminamos pomodoro?

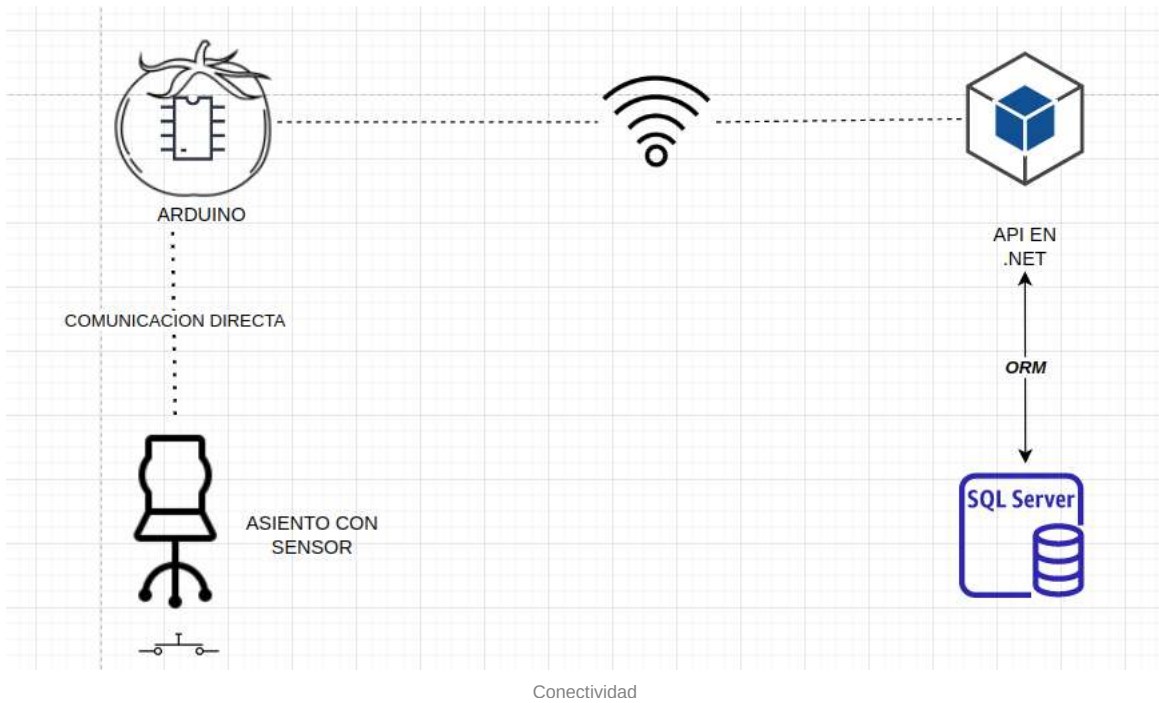
Sumatoria de penalizacion por no trabajar en un pomodoro  
especifico.

## JSON



Ejemplo de json

## DIAGRAMA



## CONSTRUCCIÓN DEL PROTOTIPO

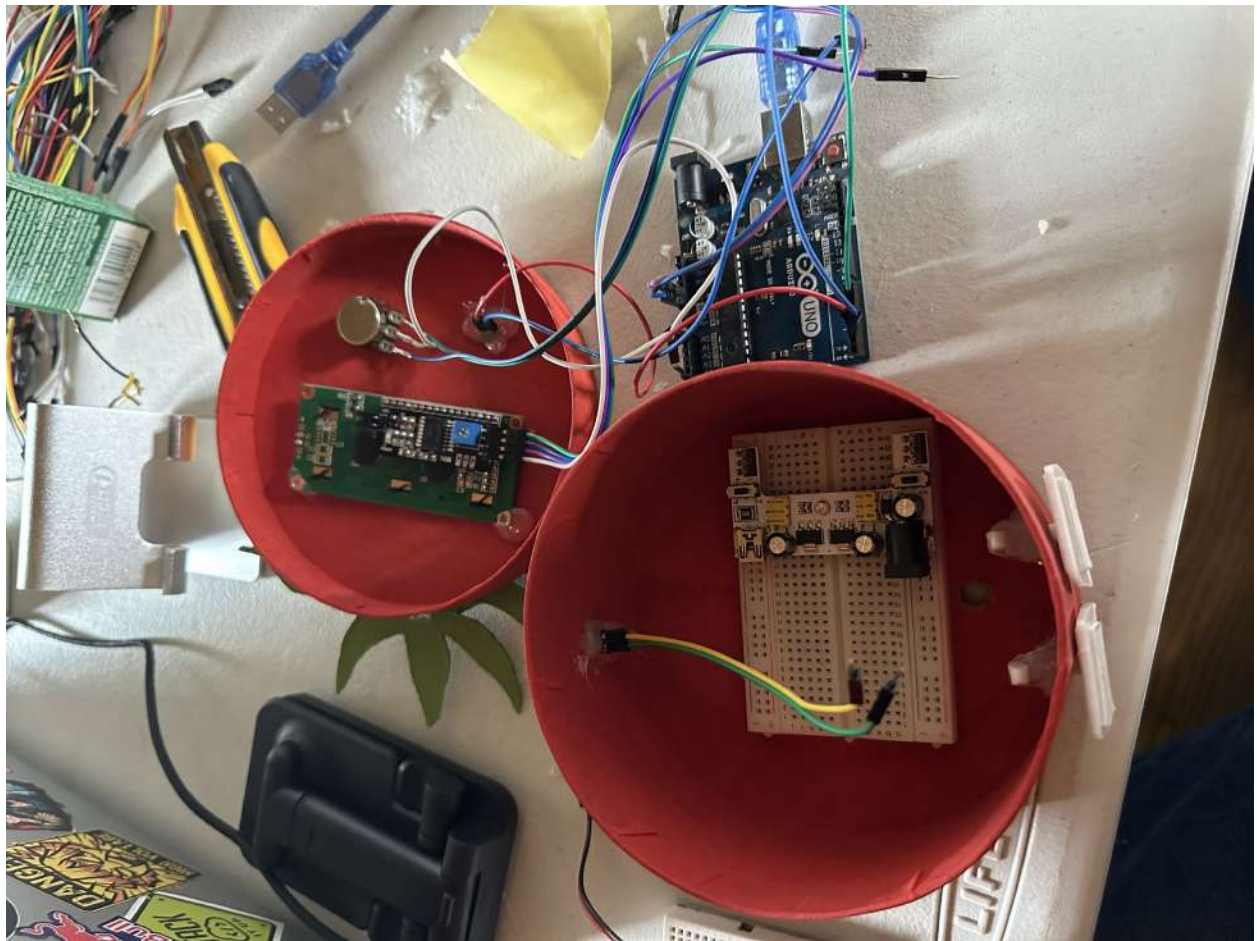


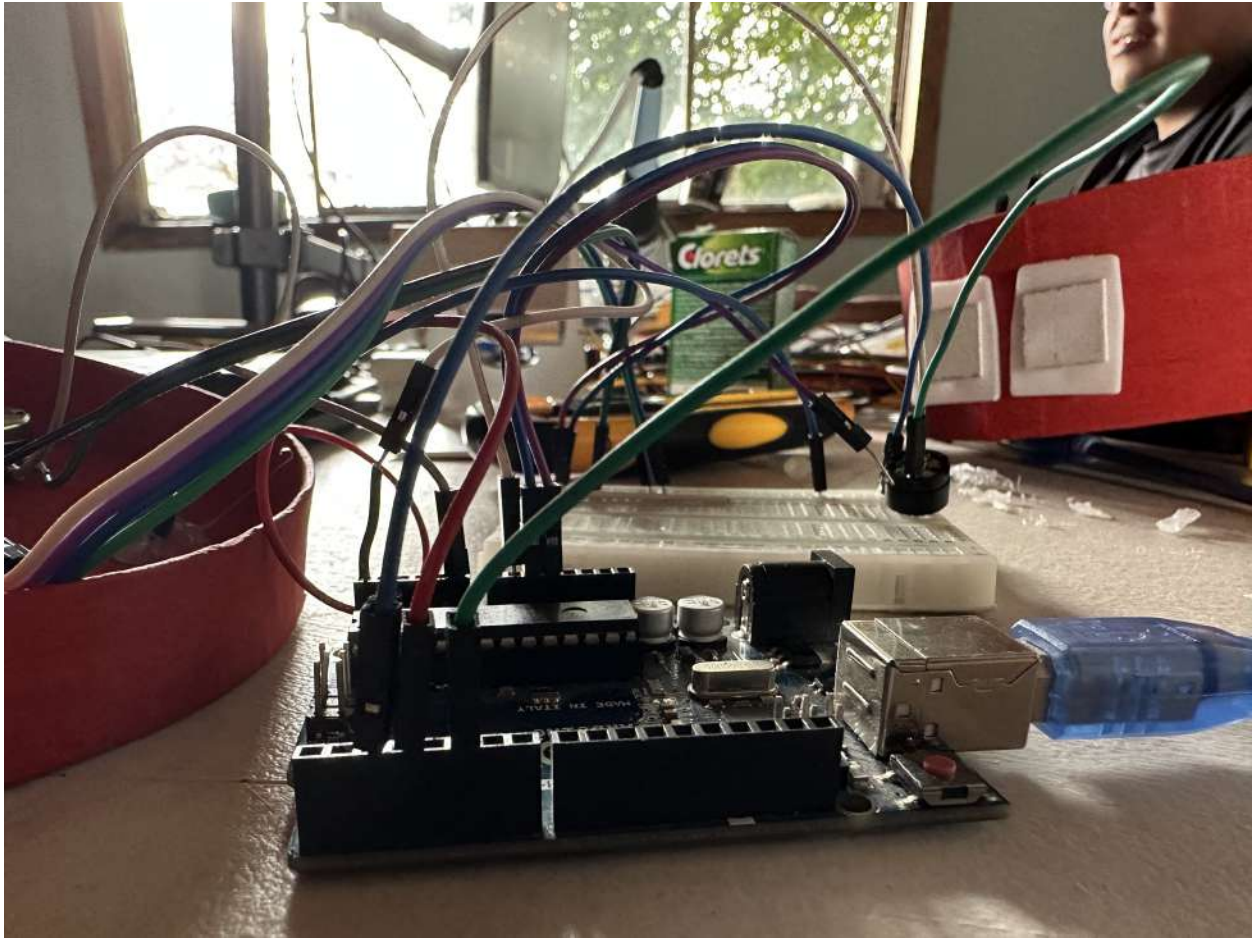




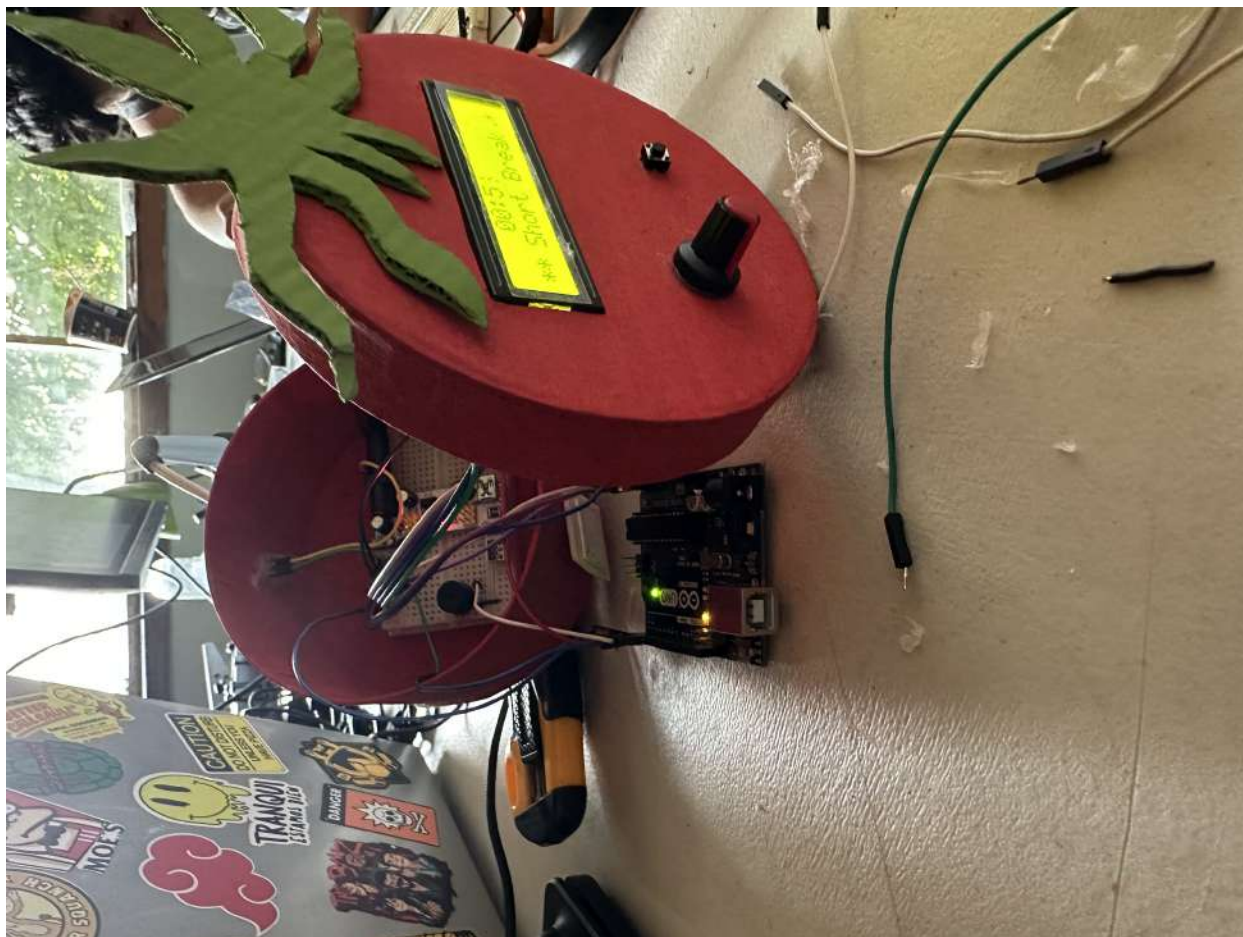












## API

A continuación se muestra una descripción de algunos endpoints de la API que utilizamos para poder desarrollar el proyecto.

/crearBD

Tipo:

Get

Parametros:

Ninguno

Descripcion:

Este endpoint lo que hace es verificar la instancia de la BD, en caso de no tener una instancia el programa creara la instancia de

Ejemplo de consumo :

`http://localhost:5012/crearBD`

/datos

Tipo:

Get

Parametros:

Ninguno

Descripcion:

Este endpoint nos retorna todos los datos de los registros existentes.

Ejemplo de consumo :

<http://localhost:5012/datos>

Ejemplo de datos que podriamos recibir:

```
[
  {
    "id": "0d894059-5532-439c-b36c-b7769f041670",
    "fecha": "20/03/2023 7:00:00",
    "fecha_corta": "20/03/2023",
    "descansoLargo": false,
    "descansoNormal": false,
    "inicio": true,
    "fin": false,
    "numeroPomodoro": 1,
    "numeroDescanso": -1,
    "dia": "20",
    "mes": "3",
    "anio": "2023",
    "hora": 7,
    "minuto": 0,
    "segundo": 0,
    "userName": "Batman",
    "sentado": false
  },
  {
    "id": "ad90f481-6039-4c28-b03f-07eb4d629e37",
    "fecha": "20/03/2023 7:25:00",
    "fecha_corta": "20/03/2023",
    "descansoLargo": false,
    "descansoNormal": false,
    "inicio": false,
    "fin": true,
    "numeroPomodoro": 1,
    "numeroDescanso": -1,
    "dia": "20",
    "mes": "3",
    "anio": "2023",
    "hora": 7,
    "minuto": 25,
    "segundo": 0,
    "userName": "Batman",
    "sentado": false
  },
  {
    "id": "552543e7-1385-4f1b-be70-6fdcf73699e5",
    "fecha": "20/03/2023 7:25:00",
    "fecha_corta": "20/03/2023",
    "descansoLargo": false,
    "descansoNormal": true,
    "inicio": true,
    "fin": false,
    "numeroPomodoro": -1,
    "numeroDescanso": 1,
    "dia": "20",
    "mes": "3",
    "anio": "2023",
    "hora": 7,
    "minuto": 25,
    "segundo": 0,
    "userName": "Batman",
    "sentado": false
  },
  {
    "id": "5c2bc8e8-15b6-4ef0-aa0f-ec51335500b7",
    "fecha": "20/03/2023 7:30:00",
    "fecha_corta": "20/03/2023",
    "descansoLargo": false,
    "descansoNormal": true,
    "inicio": false,
    "fin": true,
    "numeroPomodoro": -1,
    "numeroDescanso": 1,
    "dia": "20",
    "mes": "3",
    "anio": "2023",
    "hora": 7,
    "minuto": 30,
    "segundo": 0,
    "userName": "Batman",
    "sentado": false
  }
]
```

```

        "anio": "2023",
        "hora": 7,
        "minuto": 30,
        "segundo": 0,
        "userName": "Batman",
        "sentado": false
    }
]

```

**/datosUser**

**Tipo:**

Get

**Parametros:**

Ninguno

**Descripcion:**

Este endpoint nos retorna los datos de la configuracion de la aplicacion, como el nombre de usuario y la configuracion de los pomod

**Ejemplo de consumo:**

`http://localhost:5012/datosUser`

**Ejemplo de datos que podriamos recibir**

```

[
  {
    "id": "a5408153-d0ca-4acc-838c-df985c4ad52f",
    "userName": "Batman",
    "valPomodoro": 25,
    "valDescanso": 5,
    "valDescansoLargo": 10
  }
]

```

**/actualizarParametrosApp**

**Tipo:**

Put

**Parametros:**

se necesita que se envíen los valores para actualizar.

nameUser -> indica el nuevo nombre del usuario

nuevoValPomodoro -> indica el nuevo valor del pomodoro, tiempo de trabajo.

nuevoValDescanso -> indica el tiempo de descanso.

**ejemplo:**

```

{
  "nameUser": "SuperMan",
  "nuevoValPomodoro": 9,
  "nuevoValDescanso": 10,
  "nuevoValDescansoLargo": 10
}

```

**Descripcion:**

Este metodo nos sera de utilidad al querer editar los datos del usuario, aca podremos enviar el nombre del usuario, el tiempo del p

**Ejemplo de consumo:**

`http://localhost:5012/actualizarParametrosApp`

**Ejemplo de datos a enviar al endpoint:**

```

{
  "nameUser": "SuperMan",
  "nuevoValPomodoro": 9,
  "nuevoValDescanso": 10
}

```

**/agregarRegistro**



Tipo:

Post

Parametros:

fecha -> Variable de tipo string, la fecha puede o no venir, en caso de enviar la fecha se solicita que siga el formato " dd/MM/yyyy

descansoLargo -> Variable booleana que indica si el registro actual es un descanso largo o si no lo es. Si no es un descansoLargo d

descansoNormal -> Variable booleana que indica si el registro actual es un descanso normal o si no lo es. Si no es un descansoNorma

inicio -> Variable booleana, nos indica si el registro actual es el inicio o si no lo es, esto nos sirve para validar donde comeinz

fin -> Variable booleana, nos indica si el registro actual es el fin o si no lo es, esto nos sirve para validar donde termina ya se

numeroPomodoro -> Variable entera que indica que numero de "pomodoro" es el que estamos trabajando, en caso de no estar registrando

numeroDescanso -> Variable entera que indica que numero de "descanso" es el que estamos trabajando, en caso de no estar registrando

userName -> Variable de tipo string, nos indica el nombre del usuario que esta posteando este registro, si no esta definido se busc

sentado -> Variable de tipo booleana que nos indica si el usuario esta sentado o si no lo esta. DEBE DEFINIRSE

Ejemplo de consumo:

http://localhost:5012/agregarRegistro

Ejemplo de datos a enviar al endpoint :

```
{
  "fecha": "20/03/2023 07:00:00",
  "descansoLargo": false,
  "descansoNormal": false,
  "inicio": true,
  "fin": false,
  "numeroPomodoro": 1,
  "numeroDescanso": -1,
  "userName": "usuario1",
  "sentado": true
}
```

/obtenerUsuarios

Tipo:

Get

Parametros:

Ninguno

Descripcion:

Es para obtener todos los usuarios que tengamos registrados

Ejemplo de consumo:

http://localhost:5012/parametrosFiltro

Ejemplo de la respuesta:

```
{
  "usuarios": [
    "tomasMorales",
    "usuario1"
  ],
  "fechas": null,
  "grupos": null
}
```

/obtenerFechasUsuario

Tipo:

Get

Parametros:

nameUser -> es el nombre de usuario, buscaremos las fechas que posea este usuario en la BD

Ejemplo de consumo:

`http://localhost:5012/obtenerFechasUsuario`

Ejemplo de respuesta:

```
{
  "usuarios": null,
  "fechas": [
    "20/02/2023",
    "20/03/2023",
    "21/03/2023",
    "22/03/2023",
    "23/03/2023"
  ],
  "grupos": null
}
```

`/obtenerGrupos`

Tipo:

Get

Parametros:

Necesita de un usuario y de una fecha lo que hara sera ver las coincidencias de grupos.

Ejemplo de consumo:

`http://localhost:5012/obtenerGrupos`

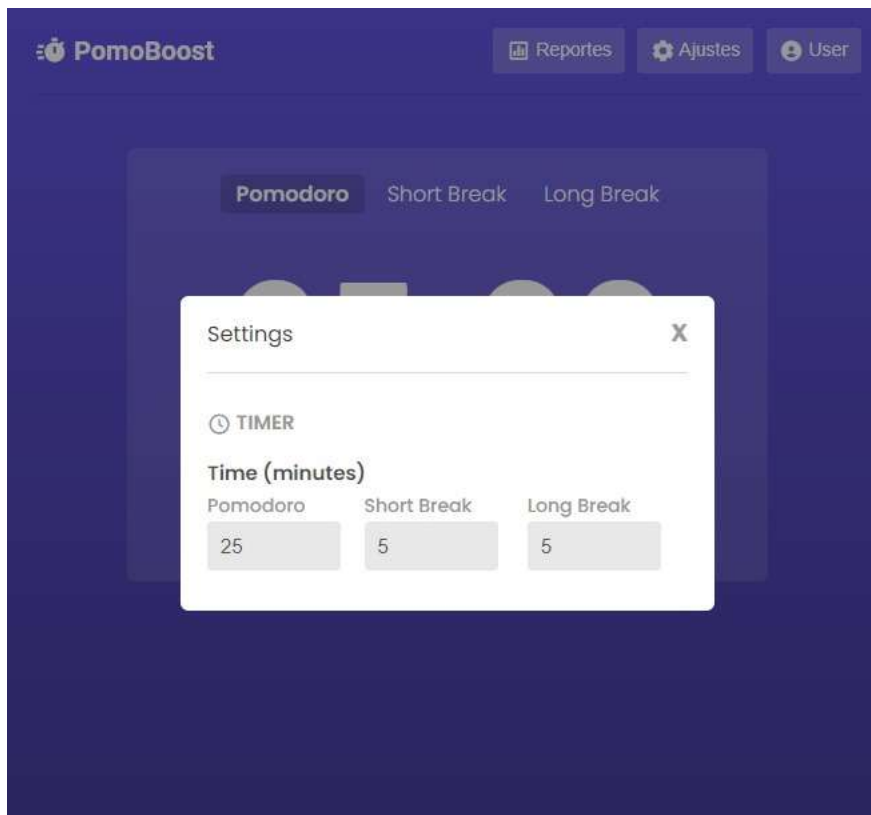
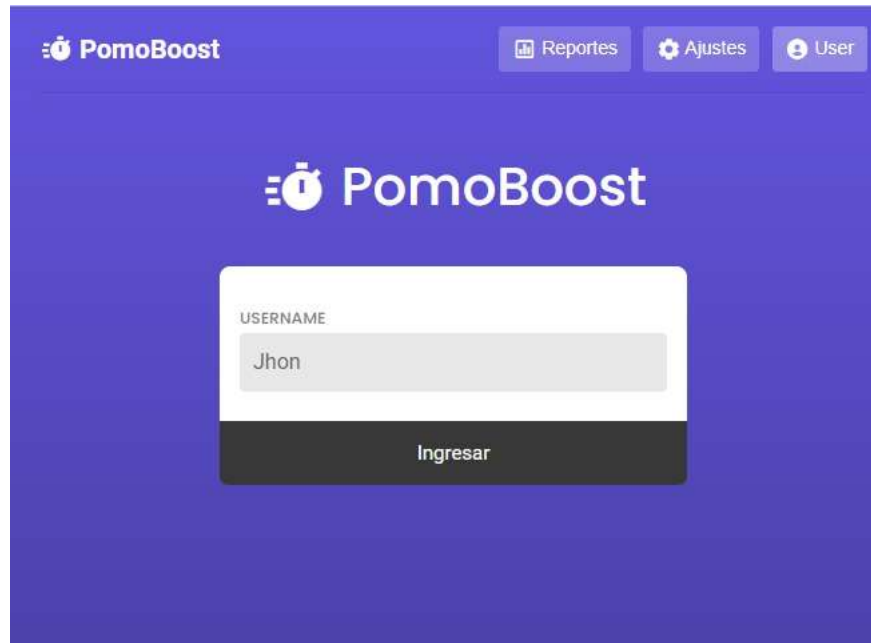
Ejemplo de peticion

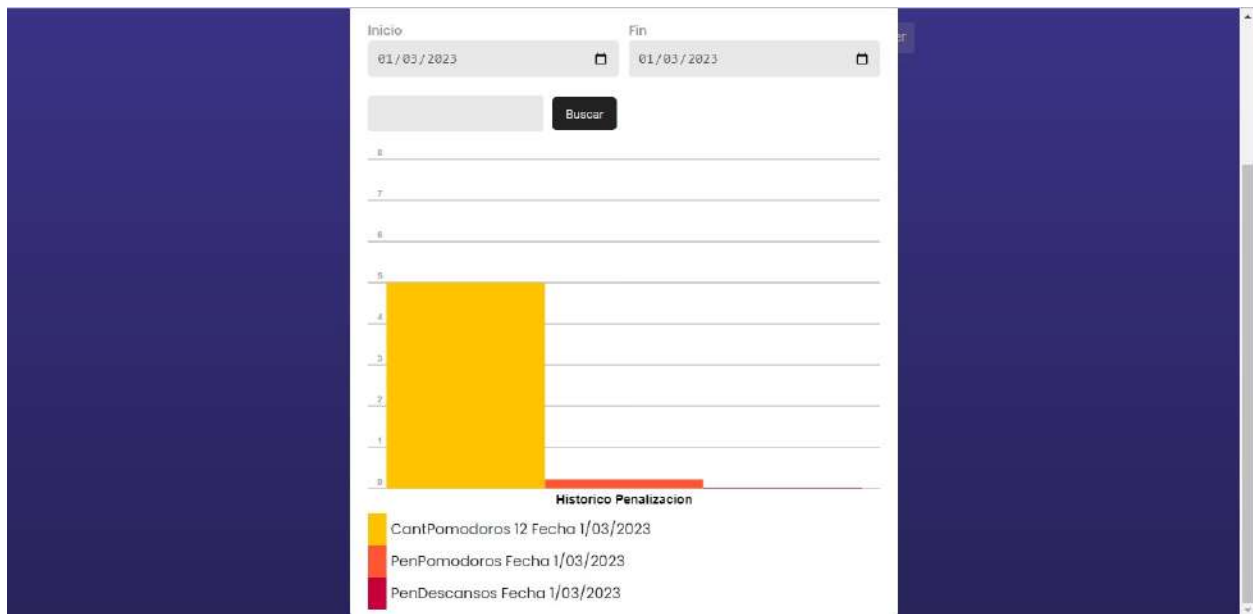
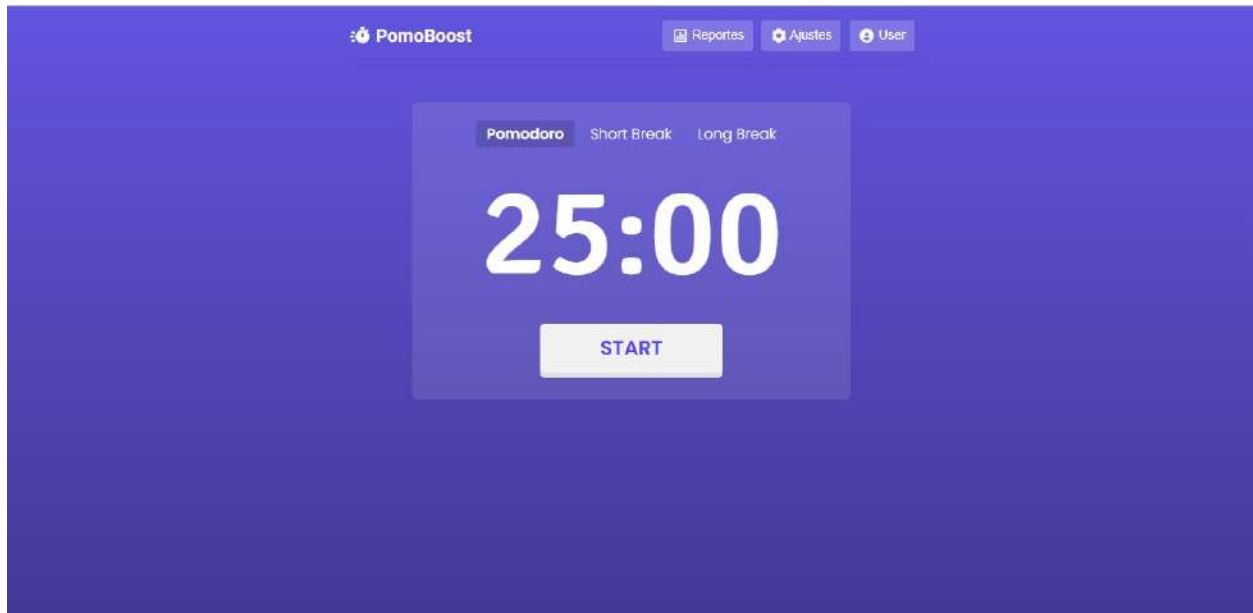
```
{
  "nameUser": "tomasMorales",
  "fecha1": "20/03/2023"
}
```

Ejemplo de respuesta:

```
{
  "usuarios": null,
  "fechas": null,
  "grupos": [
    "13aa63c6-1b1f-44f6-84be-8770eefc099b",
    "1ea6c04f-0ed9-4512-86e1-13dfcb1ca37f",
    "24394388-5935-42ce-92b5-b5e1bb8778cf",
    "b58a4137-3c0f-4f48-bad6-1b644d35e6af",
    "b9eda2ae-455d-488c-8747-e52a68de1069",
    "c69de59d-23f7-4eb5-ab4f-407463877265"
  ]
}
```

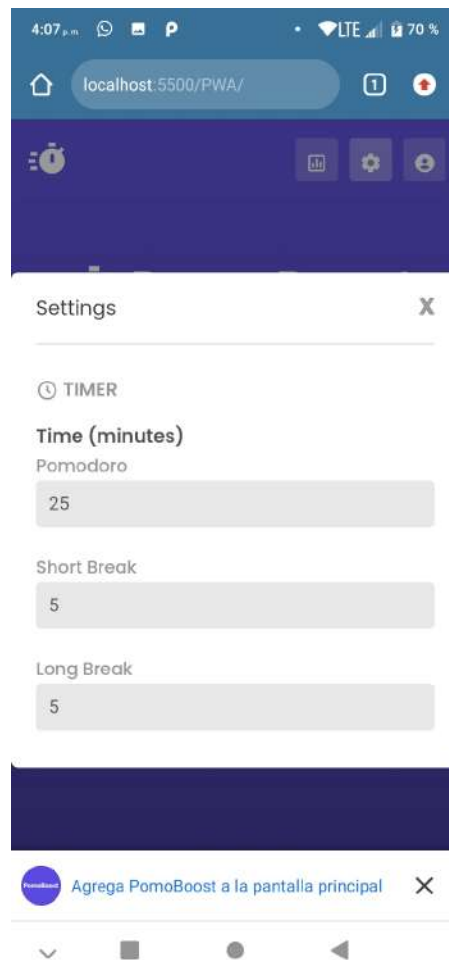
## Aplicación web



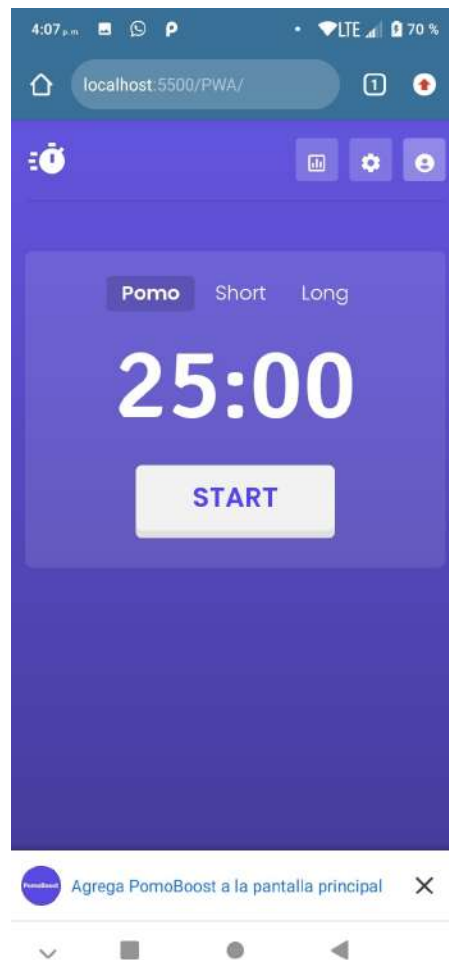


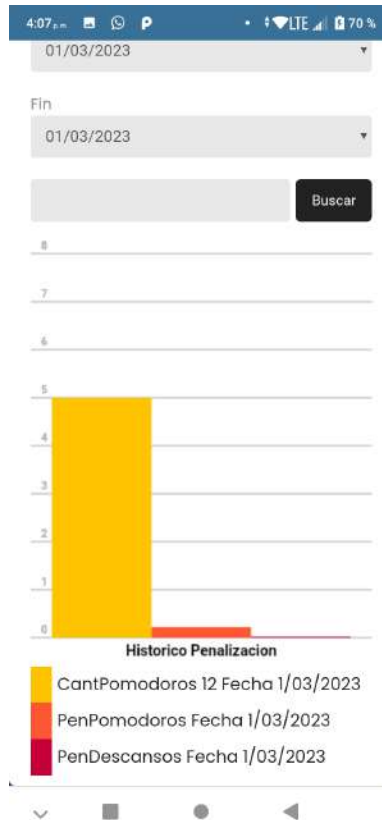
## Aplicación móvil











## Capas de framework IoT

### Product Infrastructure

La infraestructura del proyecto POMODORO distribuye un sensor de tipo discreto para el reconocimiento del estado actual del usuario, dicho estado es el que se vera evaluado en la transición de los datos.

### Sensors

Se requiere la obtención del estado actual del usuario el cual nos indicara cuando comencemos la estructura de los pomodoros, esta sera parte vital de todos los datos registrados.

### Connectivity

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

### Analystic

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 realizara el procesamiento de los registros de sql server, donde nos retornará información a largo plazo de los cambios segun el estado del usuario 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.**

`Console.WriteLine("Video");`