**Problema a resolver**

El problema que se busca resolver es la falta de acceso en tiempo real a información sobre el índice UV, temperatura y humedad.  
Este sistema permitirá a los usuarios obtener estos datos en tiempo real y descargarlos para su análisis o uso en otras aplicaciones.

**Alcance**

* Mostrar en tiempo real el índice UV, temperatura y humedad.
* Permitir el registro de usuarios para acceder a la descarga de datos.
* Ofrecer una API para obtener estos datos en formato JSON.
* Interfaz web para visualización de la información en tiempo real.
* Base de datos para gestionar usuarios y datos registrados.

**Casos de uso**

Usuario no registrado:

* Puede visualizar los datos en tiempo real desde la web.
* No puede descargar los datos históricos.

Usuario registrado:

* Puede descargar los datos históricos en distintos formatos.

Administrador:

* Puede gestionar usuarios.
* Puede administrar la base de datos.

**Casos de uso No Soportados**

* No se permitirá la descarga de datos sin registro.

**## Arquitectura**

**### Diagramas**

**### Modelo de datos**

* Tabla de usuarios (id\_usu, correo, contraseña hash, nombre, apellido, institución).
* Tabla de mediciones (fecha, hora, indice\_uv\_int, indice\_uv\_float, Vout\_330, temperatura, humedad).
* Tabla de descargas(id, id\_usu, fecha\_descarga)

**## Limitaciones**

* La API tendrá una latencia de X milisegundos debido a las consultas a la base de datos.
* Límite de X solicitudes por usuario al día para evitar sobrecarga.
* La precisión de los datos dependerá de la calidad de los sensores usados para medir el índice UV, temperatura y humedad.
* La disponibilidad del servicio puede verse afectada por fallas en el servidor o en la red.

**## Costo**

Descripción/Análisis de costos

Ejemplo:

"Considerando N usuarios diarios, M llamadas a X servicio/baseDatos/etc"

\* 1000 llamadas diarias a serverless functions. $XX.XX

\* 1000 read/write units diarias a X Database on-demand. $XX.XX

Total: $xx.xx (al mes/dia/año)