

WaterApp

Tu app para medir agua



Documentación

Semestre 2022-2022

Ingeniería en Programación

Proyecto Medidor de Agua

Carlos Alberto Tavera Arteaga



Introducción

1.Introducción

Este proyecto de desarrollo de software es para implementar un medidor de agua que puede ser fácilmente instalado en un tanque de agua para conocer el nivel en todo momento.

1.1 Propósito

Darles a los clientes un mayor acceso a conocer el nivel del agua en los tanques de sus hogares, ya sea un tinaco, aljibe o cualquier otro lugar en donde se pueda almacenar agua.

Minimizar la interacción humana con los aljibes, tinacos u otra forma de almacenamiento de agua.

1.2 Alcance

El público en general.

Un ejemplo de lo que se puede hacer, es instalar medidores en todos los hogares de un fraccionamiento pequeño para tener el control del nivel de agua en los hogares.

1.3 Personal involucrado

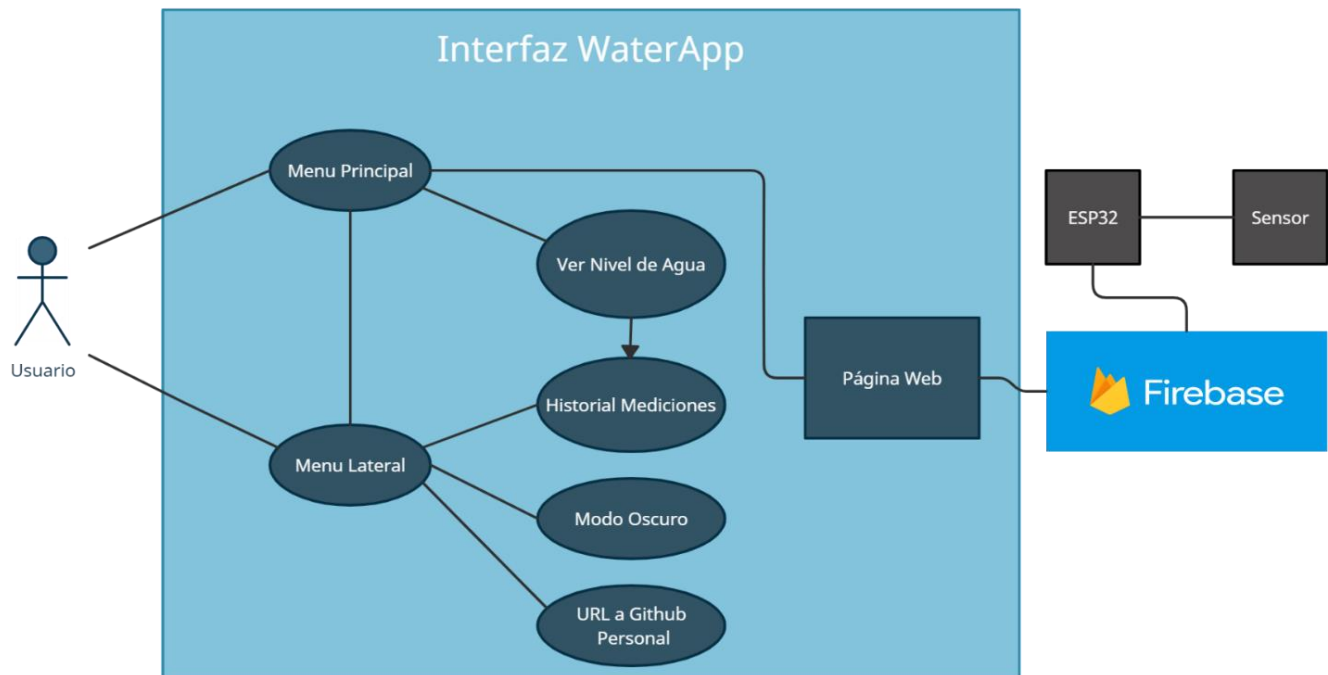
Nombre	Carlos Alberto Tavera Arteaga
Rol	Desarrollador
Categoría profesional	Desarrollador
Responsabilidades	UI UX, Programación del sensor, Full stack.
Información de contacto	https://github.com/LordShadow505
Aprobación	Si

Descripción General

2 Descripción general

2.1 Perspectiva del producto

Para visualizar de una mejor manera el contexto general del dispositivo, podemos analizar el siguiente diagrama de casos de uso:



2.2 Funcionalidad del producto

Mostrar el porcentaje de llenado en un tanque de agua.

Mostrar una gráfica en donde se vea el historial de mediciones.

Ser accesible desde cualquier dispositivo en cualquier lugar con acceso a internet.

Descripción General

2.3 Características de los usuarios

Tipo de usuario	Publico General.
Formación	Cualquiera.
Requisitos	Un dispositivo con acceso a internet (Smartphone, Computadora, etc).
Actividades	Acceder a un navegador.

2.4 Restricciones

Al ser una web app, solo puede ser accesible si el usuario tiene acceso a internet y en el caso de la app para teléfonos móviles, se requiere una versión de Android superior a 5.

2.5 Evolución previsible del sistema

Se prevé hacer una web app con la única opción de mostrar el nivel de agua al usuario por medio de un diseño en la pantalla que indique el porcentaje de agua en el tanque.

Después de distintas implementaciones y cambios se pretende mejorar la confiabilidad y la portabilidad de las aplicaciones a otros sistemas.

Requisitos

3 Requisitos específicos

(Una restricción es un prerrequisito que nos impide o bloquea algún otro requisito, por lo tanto, es una prioridad que debe estar desde la primera implementación de la webapp)

Número de requisito	1
Nombre de requisito	R1-Conexión a WiFi.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	2
Nombre de requisito	R2-Interfaz Gráfica.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	3
Nombre de requisito	R3-Mostar la capacidad del tanque en porcentaje.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	4
Nombre de requisito	R4-Conexión a una base de datos.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	5
Nombre de requisito	R5-Portabilidad a navegadores y dispositivos móviles
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	6
Nombre de requisito	R6-Fácil utilización y bajo mantenimiento.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Requisitos

Número de requisito	7
Nombre de requisito	R7-Impermeabilidad en el hardware.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	8
Nombre de requisito	R8-Interfaz moderna.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Emocionante

Número de requisito	9
Nombre de requisito	R9-La caja debe ser estética.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Emocionante

Número de requisito	10
Nombre de requisito	R10-Mínima interacción del usuario con el hardware.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	11
Nombre de requisito	R11-La app debe ser compatible con Android 5+
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	12
Nombre de requisito	R12-El dispositivo debe ser lo más compacto posible.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	13
Nombre de requisito	R13-La webapp debe ser accesible desde cualquier navegador.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Requisitos

Número de requisito	14
Nombre de requisito	R14-Debe funcionar en móvil y pc.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	15
Nombre de requisito	R15-Interfaz Adaptable al tamaño de pantalla.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	16
Nombre de requisito	R16-Tener un apartado para graficar el historial de mediciones.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	17
Nombre de requisito	R17-Debe ser fácil de instalar.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	18
Nombre de requisito	R18-Dispositivo resistente (Uso rudo)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Emocionante

Número de requisito	19
Nombre de requisito	R19-Modo claro y modo oscuro.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Emocionante

Número de requisito	20
Nombre de requisito	R20-Debe tener animaciones.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Emocionante

Requisitos

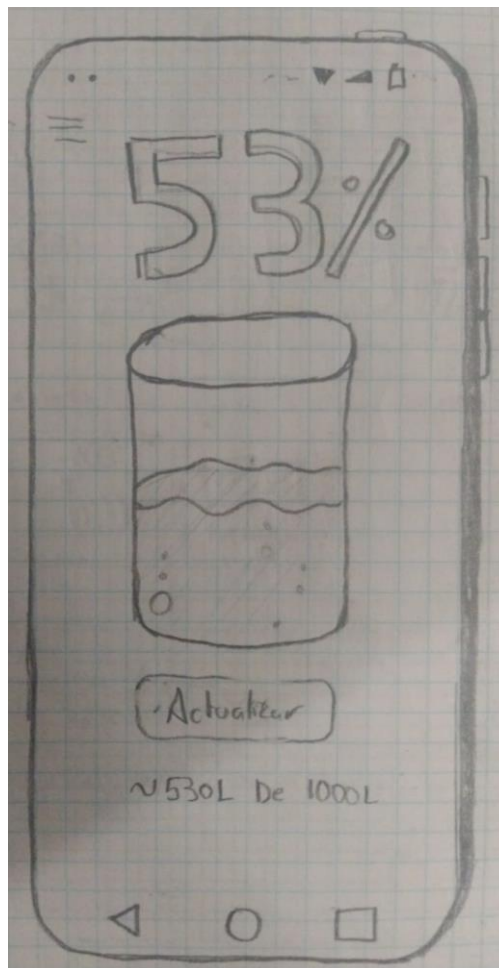
3.1 Requisitos comunes de las interfaces

Entradas: La interfaz no contiene entradas de texto.

Salidas: El porcentaje de agua en el dispositivo, el menú lateral, la gráfica con el historial de mediciones

3.1.1 Interfaces de usuario

Antes de la realización de la interfaz de usuario se hizo un boceto previo a modo de concepto:



Requisitos

3.1.2 Interfaces de hardware

No hay interfaces de hardware.

3.1.3 Interfaces de software

La webapp hará uso de una base de datos en tiempo real llamada Firebase, la cual contiene una interfaz gráfica donde se puede visualizar los datos que recibe del dispositivo medidor de agua.



De igual manera se utilizará una librería llamada Highcharts.js para mostrar la gráfica de mediciones, la cual contiene interfaces propias de la librería.

}



3.1.4 Interfaces de comunicación

La interfaz de comunicación que se va a implementar es una conexión con una página web y un webviewer para Android, cada una de estas usa la misma interfaz gráfica que el apartado 3.1.1.

Requisitos

3.2 Requisitos funcionales

3.2.1 Requisito funcional 1

Hacer 5 mediciones y elegir la que se repite más veces para evitar valores erróneos que genera el sensor.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Se espera que las 5 mediciones se hagan en menos de 2 segundos para garantizar que la base de datos se actualiza rápidamente.

3.3.2 Seguridad

La base de datos solo será accesible a usuarios con acceso.

3.3.3 Fiabilidad

Si la base de datos es desconectada se espera que el sistema pueda recuperar la conexión sin en cualquier parte del ciclo detecta que no se está conectando a la base de datos.

3.3.4 Disponibilidad

El software estará disponible para los principales navegadores (Chrome, Edge y Firefox), así como también en Android 5+.

3.3.5 Mantenibilidad

Los mantenimientos del sistema deben ser realizados por un desarrollador. El mantenimiento deberá realizarse cada una vez cada seis meses o lo que el desarrollador considere adecuado.

3.3.6 Portabilidad

Para facilitar la portabilidad, se usará un diseño adaptable a distintos tamaños de pantalla, permitiendo así, ser visualizado en dispositivos móviles, computadoras, o, incluso televisores con acceso a un navegador.

Contrato

El desarrollo de este proyecto está dirigido por el desarrollador, *Carlos Alberto Tavera Arteaga*, el plazo de entrega son 6 meses, en aproximadamente 4 horas por semana, el desarrollo será informado a los superiores de la empresa, al final del proyecto se deberá entregar una documentación con este contrato adjunto.

Se deben respetar los plazos de entrega así como los requisitos planteados al inicio, el modelo de proceso utilizado será en espiral, por lo tanto, deben mostrarse avances cada semana, así como una fracción de documentación.

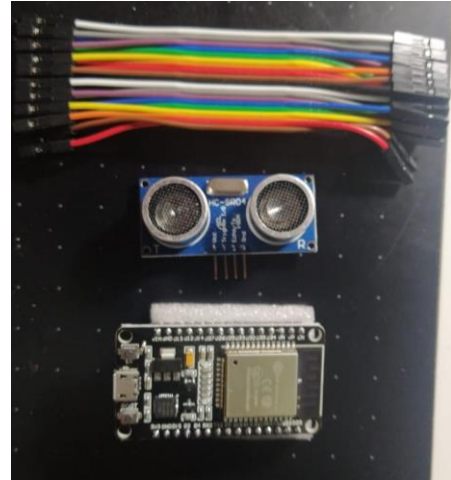
El pago de comisión por el proyecto será de \$250/semana, estaremos en contacto con este medio para tratar cualquier situación relacionada con el proyecto.

Proceso de Desarrollo

Preeliminares:

Para la implementación del proyecto se decidió usar la tarjeta de desarrollo ESP-32 y el sensor ultrasonico HC-SR04, con el cual se realizaran las mediciones de distancia para sacar el porcentaje de llenado en el tanque.

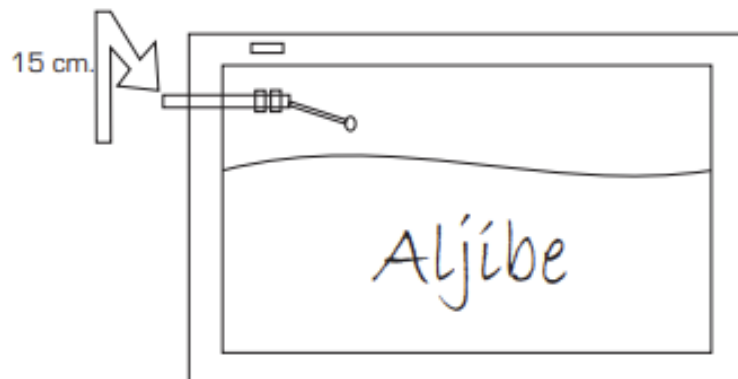
El ESP-32 tiene acceso a WiFi y es bastante económico, lo cual ahorrará presupuesto en el proyecto, además de que es bastante compacto y se calienta poco.



Primera implementación: Lecturas del sensor

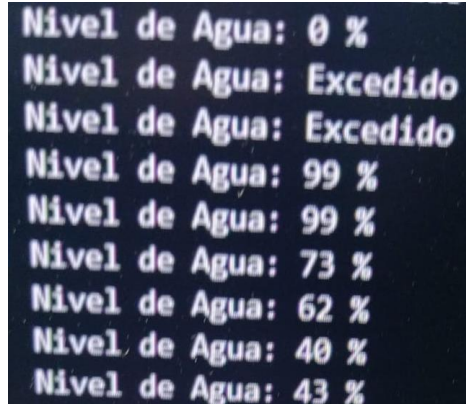
Para la primera implementación se desarrollo un pequeño programa que calcula la distancia a un objeto, esto más tarde serviría para calcular el porcentaje de llenado de un aljibe mexicano promedio (Es decir aproximadamente 2 metros de altura), para calcularlo se tomó en cuenta la documentación de una constructora de aljibes, la cual menciona lo siguiente:

“Es necesario revisar la válvula y el flotador. Es necesario mantener el nivel del agua 15 cm. abajo de la válvula del flotador”.



Proceso de Desarrollo

Tomando eso en cuenta, se decidió que la lectura del sensor empezara a 20 cm del sensor, esto para garantizar que la lectura sea lo más exacta posible (Ya que de otra manera la medición marcaría menos porcentaje del que debería ser), para mostrar los datos se empleó la terminal de Visual Studio, y una vez se comprobó múltiples veces que los datos fueran correctos se pasó a la siguiente fase.

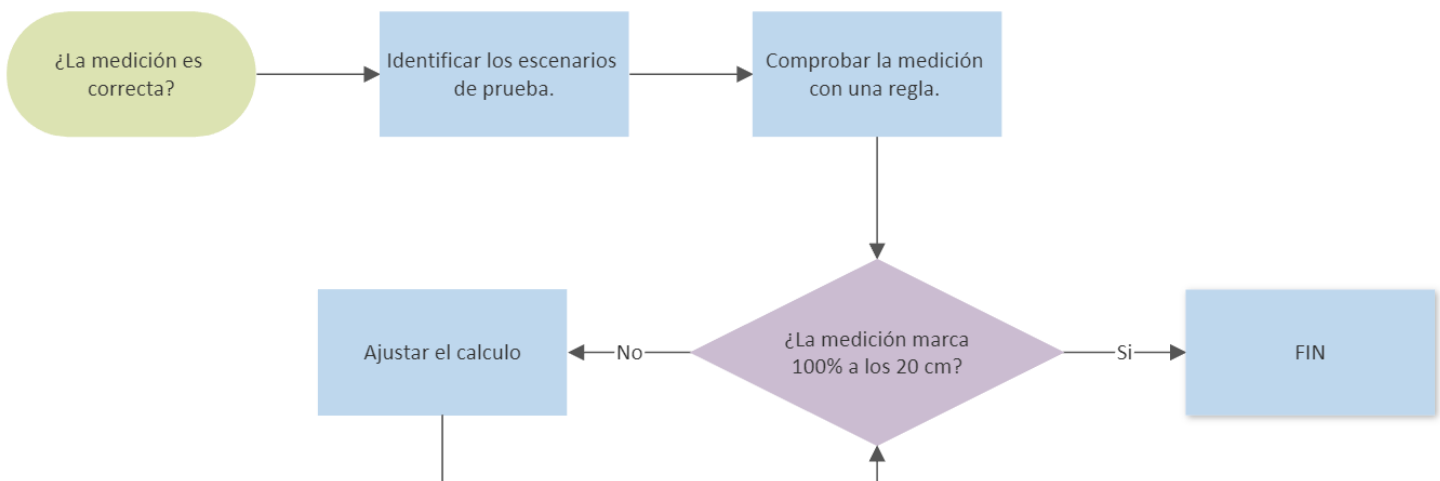


Nivel de Agua: 0 %
Nivel de Agua: Excedido
Nivel de Agua: Excedido
Nivel de Agua: 99 %
Nivel de Agua: 99 %
Nivel de Agua: 73 %
Nivel de Agua: 62 %
Nivel de Agua: 40 %
Nivel de Agua: 43 %



Sin el ajuste de 20 cm marcaba ~90% y realizando el ajuste correspondiente el sensor marca la lectura correcta 100%

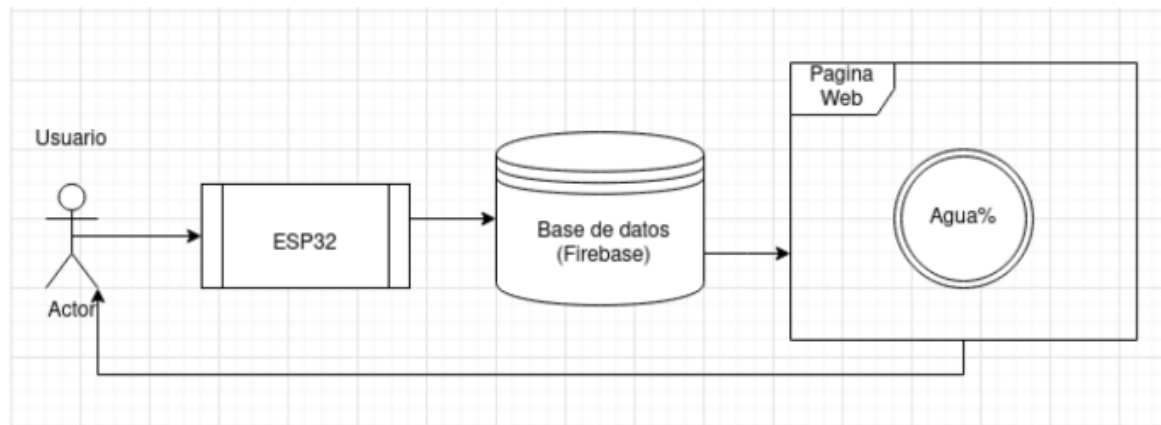
Pruebas de unidad primera implementación:



Proceso de Desarrollo

Segunda implementación: Firebase

Para la segunda implementación, se decidió usar una base de datos para subir la medición del sensor, ya que por limitaciones del ESP-32 es más difícil implementar el HTML, CSS y JavaScript por separado, además de que esto no permite crear una página web en línea (Ya que solo se puede hacer de manera local con el ESP-32), por lo tanto luego de una investigación se llegó a que Firebase de Google sería una buena opción para implementar tanto la base de datos, como el host de la página, ya que la plataforma tiene ambas opciones y se pueden enlazar más fácilmente.



La medición es actualizada en tiempo real, por lo tanto, está disponible siempre y cuando el ESP32 esté conectado a internet y a su vez a Firebase, para este momento aun no se tenía interfaz, por lo tanto, el dato se mostraba como texto en la pantalla.

Como indicador extra, el dispositivo enciende un led azul si el dispositivo está conectado correctamente a Firebase.

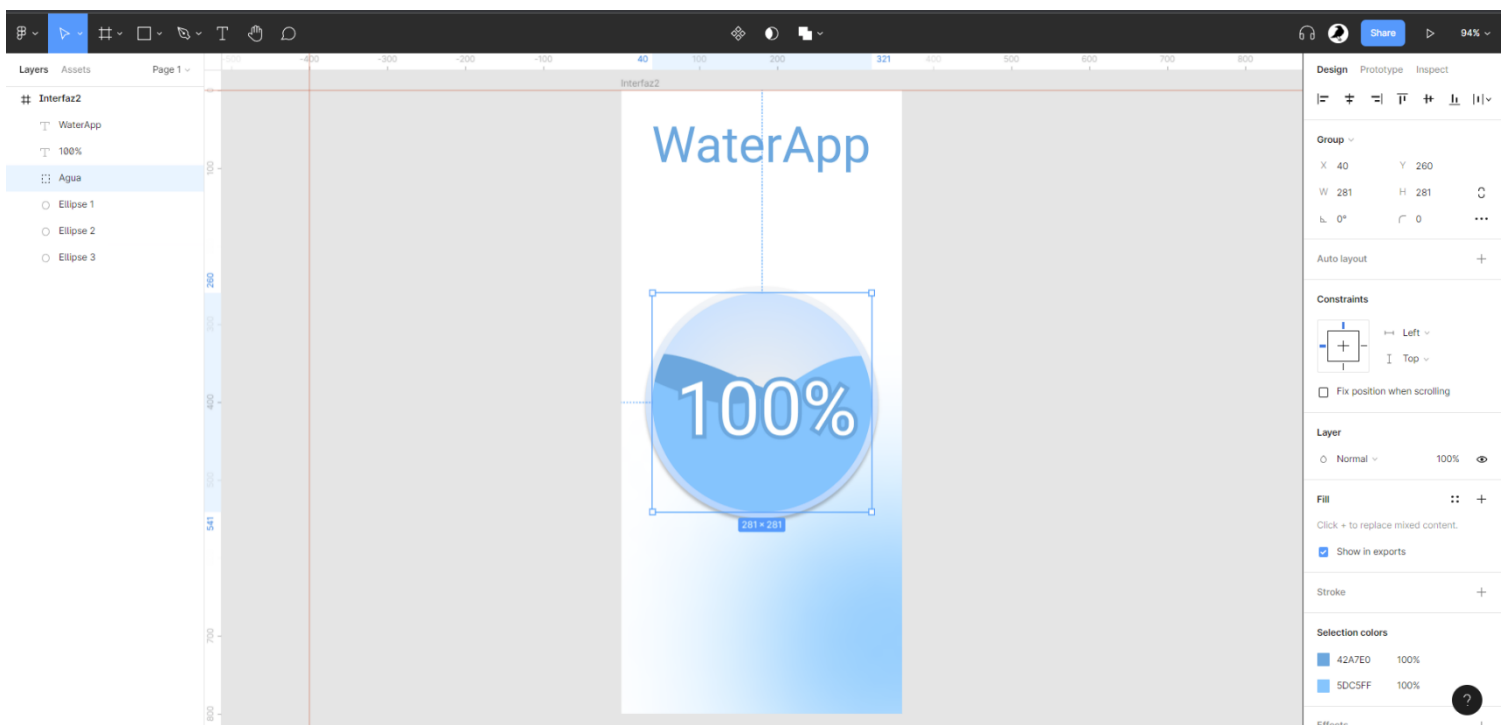
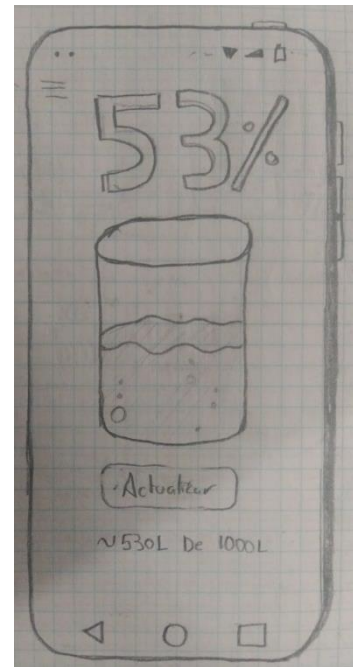
Proceso de Desarrollo

Tercera implementación: Primera Interfaz

Para implementar la interfaz, se partió del boceto de concepto que se planteó inicialmente al cliente.

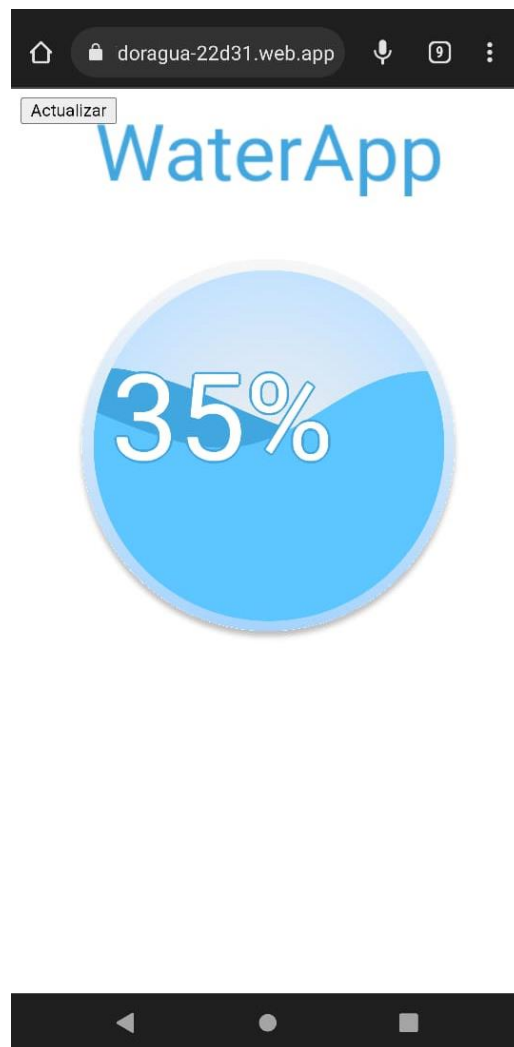
Al principio se pensó en un recipiente/vaso que contuviera el agua en su interior, pero luego se planteó que fuera una esfera de cristal, ya que esto le daría un diseño más moderno a la interfaz, para hacer esta interfaz, se uso una herramienta de diseño de interfaces llamada Figma, la cual permite crear conceptos de interfaces que posteriormente pueden ser utilizadas para desarrollar la interfaz en código.

La herramienta fue muy útil a la hora de determinar los colores y el espaciado de la interfaz, así como también, generar los SVG que servirían como concepto en la primera implementación de la interfaz de la página web.



Proceso de Desarrollo

Aquí hay una captura de pantalla (Tomada desde un dispositivo móvil) donde se la primera vez que la interfaz funcionó, como se puede ver los datos no se mostraban centrados en el círculo, y aparecía un botón que dice actualizar, el cual tomaba manualmente el dato de Firebase, esto se corrigió más tarde para que fuera automático.



Proceso de Desarrollo

Cuarta implementación: Carcasa

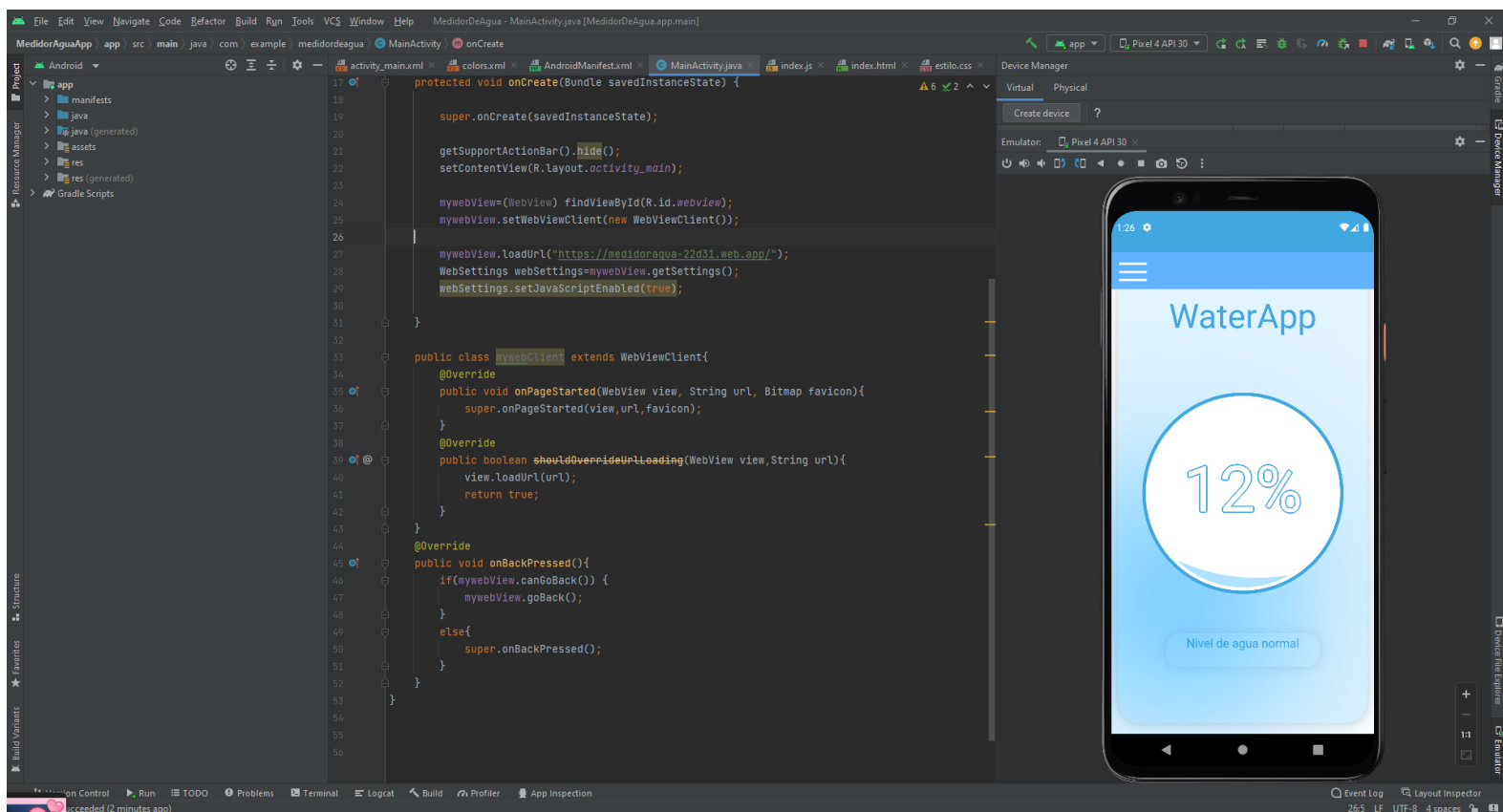
Como el dispositivo va a ser instalado en un tanque de agua, se prevé que pueda resistir a la corrosión y las salpicaduras, por lo tanto el material utilizado fue acrílico, el cual se selló con resina epoxica para asegurar que no se fugara agua al interior, como tambien se quiere acceder al dispositivo en cualquier momento, se necesita que pueda ser accesible de alguna manera, por lo tanto se soldaron bases para tornillos a unas escuadras de metal en el interior y para sellarlo a prueba de agua se utilizó un empaque improvisado.



Proceso de Desarrollo

Quinta implementación: App de Android

Para la implementación en Android utilicé la IDE de Android Studio, para esto, la manera mas sencilla de realizarlo fue hacer un webviewer, lo cual quiere decir que se implementó un navegador interno dentro de la app, el cual permite ver la página web e instalar la app en cualquier dispositivo Android con una versión superior a 5. Android Studio tiene un emulador de Android incorporado y en la siguiente imagen puede verse la implementación en un dispositivo virtual.



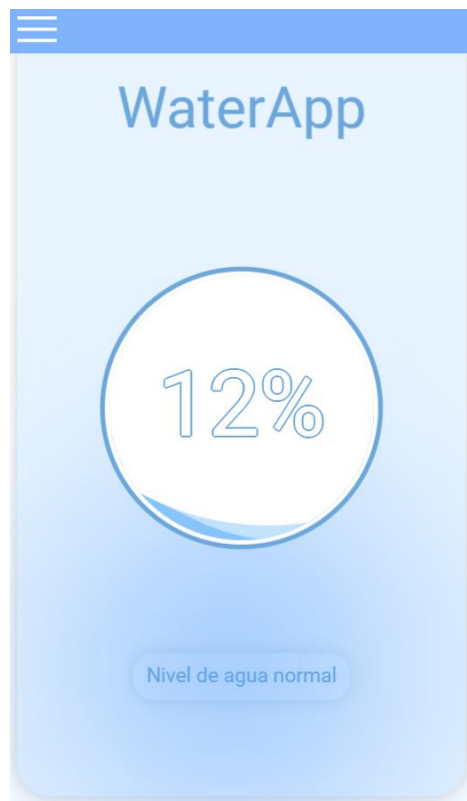
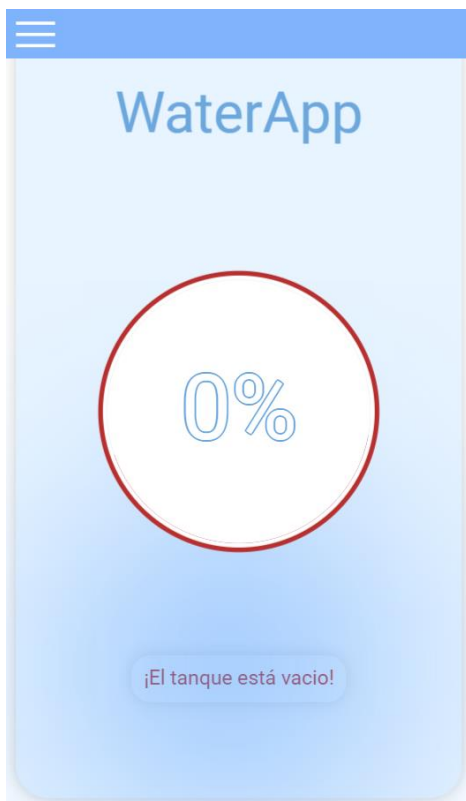
(El archivo apk esta disponible para su descarga en el repositorio de GitHub)

Proceso de Desarrollo

Sexta implementación: Interfaz final y Gráfica

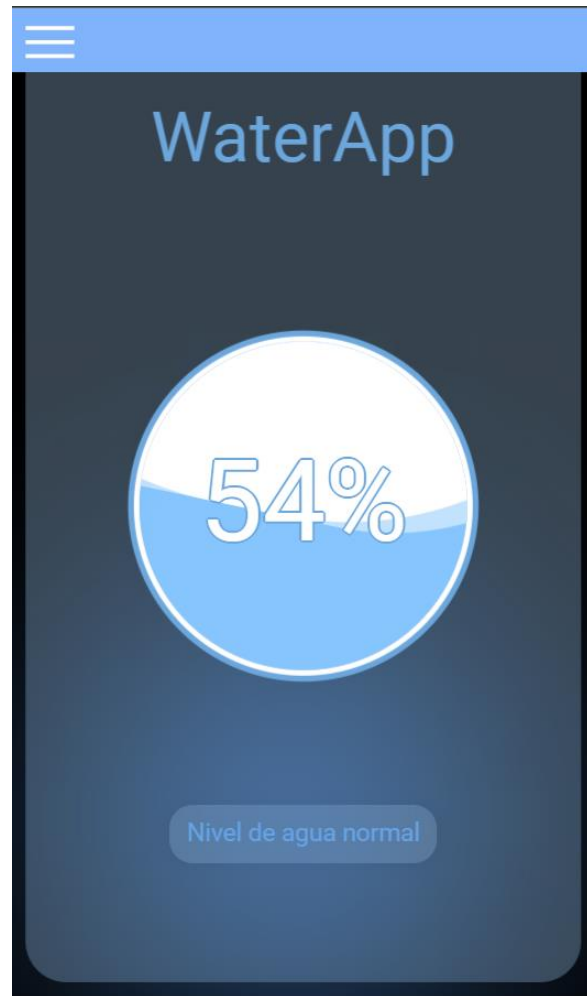
En la quinta implementación se realizó la última iteración de la interfaz, en donde:

1. El nivel del agua cambia de posición de acuerdo al porcentaje del tanque.
2. Si el tanque está lleno el color de la interfaz cambia, al igual cuando el tanque está vacío.
3. Se implementó una gráfica para mantener un historial del nivel del agua en las últimas 24 horas.
4. Modo oscuro que puede ser alternable con un botón.
5. Mostrar un mensaje indicando una advertencia.
6. Acceso directo al Github del desarrollador.



Proceso de Desarrollo

Gráfica y modo oscuro:



Verificación y Validación

El proyecto tuvo una fase de verificación cuando el proyecto estaba al 70% y otro al ser terminado, estas pruebas de verificaciones fueron realizadas por distintos usuarios sin conocimiento técnico del programa, esto para garantizar que la app fuera fácil de utilizar, incluso por niños o personas con dificultades para utilizar dispositivos electrónicos.

La webapp fue probada en estos dispositivos y se espera que la WebApp pueda funcionar incluso con televisores, smartwach o cualquier dispositivo que contenga un navegador web, la aplicación móvil es compatible con todos los dispositivos Android con una versión 5 o superior.

WebApp	Aplicación movil
Navegador Chrome	Android 10
Navegador Edge	Android 8.0 Oreo
Navegador Firefox	Android 6.0 Marshmallow
Navegador Brave	Android 11
Navegador de Xbox	Android 5 Lollipop

El proyecto cumple con todos los requerimientos que pidió el cliente al inicio, por lo tanto, la validación fue realizada satisfactoriamente.

Conclusión

El proyecto fue entregado en el plazo acordado con el cliente y la validación se realizó correctamente, por lo tanto, se puede concluir que el proyecto está finalizado y se espera hacer mantenimiento en las próximas semanas para corregir los errores de lanzamiento que no fueron planeados originalmente.

La página web del proyecto puede ser visitada desde cualquier navegador en cualquier parte del mundo, accediendo mediante código QR o la url: <https://medidoragua-22d31.web.app/>

Este código QR puede ser usado por el cliente para ingresar a la página, y se piensa que deberá ser incluido en el manual de usuario del cliente.

