

Olvidín

Céspedes, Cristian; Guerrero, Mara; Menchaca, Brian; Vázquez Petracca, Pablo
DNI 41.704.776; DNI 40.538.513; DNI 40.476.567; DNI 33.895.297
Lunes, Grupo L1

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina

Resumen.

El proyecto describe el desarrollo de la aplicación Olvidín, pensada como complemento de un pastillero inteligente. La aplicación permite al usuario cargar horarios de medicación, eliminar horarios previamente configurados, visualizar un historial de tomas anteriores y consultar el volumen actual que se ajusta manualmente en el dispositivo físico. La aplicación actúa como una interfaz útil para organizar y seguir la rutina de medicación del usuario.

Palabras claves: pastillero inteligente, horarios de medicación, sistema embebido, aplicación

1 Introducción

La aplicación Olvidín fue diseñada como una herramienta de apoyo para personas que necesitan cumplir tratamientos médicos y requieren ayuda para recordar sus tomas de medicación. Está pensada para trabajar en conjunto con un sistema físico que gira un carrusel de pastillas y activa una alarma sonora al momento indicado.

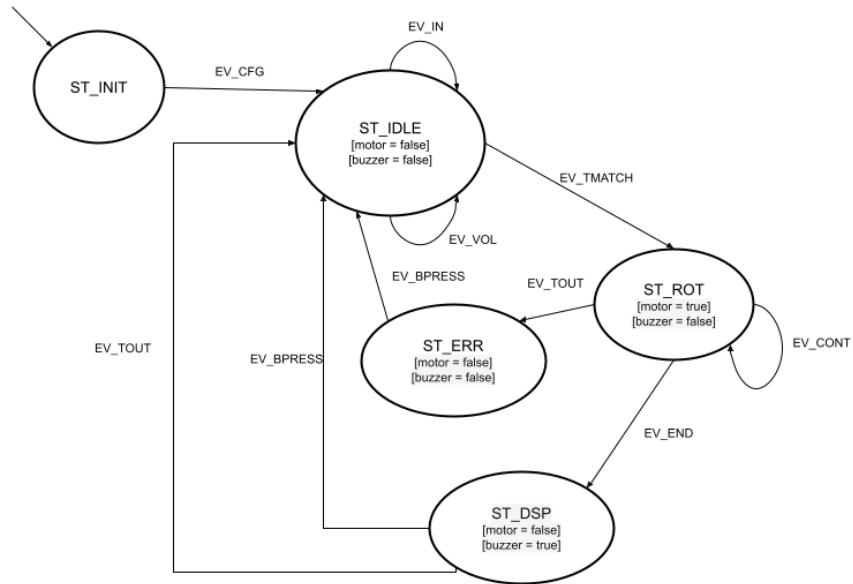
Desde la aplicación, el usuario puede programar nuevos horarios de toma, eliminar aquellos que ya no necesita, ver un historial de tomas anteriores y consultar el volumen actual, que es configurado en el hardware mediante un potenciómetro. La aplicación cumple el rol de facilitar la gestión de información desde una interfaz simple.

2 Desarrollo

Dirección Web del repositorio GitHub:
<https://github.com/UNLAM-SOA/2025-SOA-Q1-L1>

Dirección Web de Wokwi: [Simulador](#)

Diagrama de Máquina de estados:



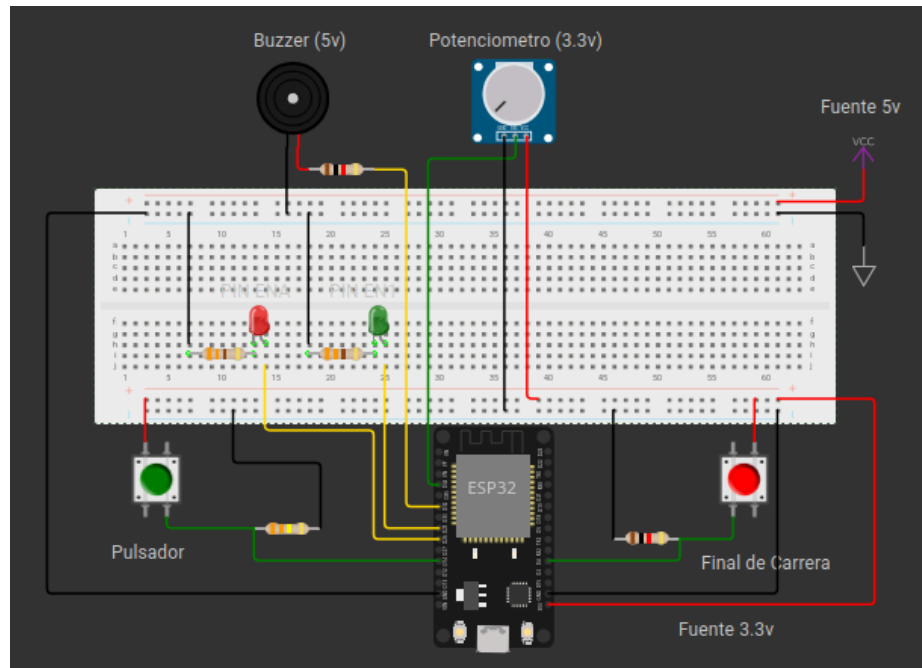
Estados

- ST_INIT: Inicio
- ST_IDLE: Inactivo
- ST_ROT: Rotando
- ST_DSP: Despachando
- ST_ERR: Error

Estimulos

- EV_CFG: Hora configurada
- EV_IN: Fecha/Hora Ingresada
- EV_TMATCH: Hora programada alcanzada
- EV_BPRESS: Pulsador presionado
- EV_END: Final de carrera detectado
- EV_VOL: Volumen modificado.
- EV_TOUT: Timeout
- EV_CONT: Continuación del proceso de rotación mientras el carrusel no llegó al final.

Circuito generado en Wokwi:



Navegación de las Activities:

MainActivity: Pantalla principal que muestra los horarios cargados. Desde acá se accede a otras funciones y también se pueden eliminar horarios.

AddScheduleActivity: Permite crear un nuevo horario de toma.

HistoryActivity: Muestra un historial de tomas realizadas.

VolumeActivity: Muestra el nivel de volumen que fue configurado manualmente en el dispositivo físico.



Manual de Usuario de la Aplicación:

La aplicación Olvidín fue diseñada como una interfaz simple para que el usuario pueda gestionar la toma de su medicación.

Al abrir la aplicación, se accede a una pantalla principal donde se visualizan los horarios activos ya cargados. Desde esta misma pantalla, el usuario tiene la posibilidad de eliminar cualquiera de los horarios previamente configurados; al hacerlo, la aplicación cancela la tarea en segundo plano asociada y también actualiza al sistema embebido para que ya no tenga en cuenta ese horario.

Para agregar un nuevo horario, se accede a una pantalla específica donde el usuario selecciona la hora deseada. Al confirmar, ese horario se guarda localmente en el dispositivo y se transmite al sistema físico a través del protocolo MQTT. De esta manera, el ESP32 del pastillero puede tomar decisiones en el momento justo, como girar el carrusel y activar el buzzer que avisa al usuario.

Además, la aplicación cuenta con una sección de historial, donde se puede consultar las tomas realizadas. Esta información permite al usuario hacer un seguimiento simple de su tratamiento. También existe una pantalla donde se muestra el valor de volumen configurado manualmente desde el dispositivo físico, a través de un potenciómetro. Ese valor es leído por el ESP32 y enviado a la aplicación mediante MQTT, para que el usuario lo pueda visualizar desde su celular.

Una vez que llega la hora programada, el sistema embebido activa el buzzer y gira el carrusel para dejar disponible la pastilla. Para finalizar la alerta sonora, el usuario debe presionar un botón físico ubicado en el dispositivo. Este pulsador cumple la función de indicar que la pastilla fue tomada, y al ser presionado, el sistema vuelve automáticamente al estado de espera hasta la próxima toma.

Toda la comunicación entre la aplicación y el sistema embebido ocurre mediante MQTT, usando un broker público. La aplicación publica los horarios de toma, que son recibidos por el ESP32, y a su vez escucha los mensajes que el sistema embebido envía con información como el nivel de volumen. Toda la lógica que implica activar el buzzer, detectar el momento exacto de una toma y rotar el carrusel se ejecuta en el sistema embebido.

3 Conclusiones

Durante el desarrollo de la aplicación Olvidín, se abordaron distintos desafíos vinculados tanto al diseño de la interfaz como a la comunicación con el sistema embebido. Uno de los principales recaudos fue garantizar que la aplicación mantuviera la coherencia entre los horarios cargados por el usuario y los procesos en segundo plano encargados de gestionarlos. Para lograrlo, se implementó un mecanismo que permite programar las tomas de medicación de forma automática y cancelar aquellas que ya no corresponden, cuando el usuario elimina un horario. Esto permitió evitar ejecuciones innecesarias.

También se presentaron complicaciones durante la integración con el sistema embebido, especialmente en lo relacionado con la recepción de datos como la fecha y hora. El uso de MQTT fue clave para resolver este punto, permitiendo una comunicación efectiva entre la aplicación y el ESP32. Además, se prestó atención a que la lógica de cada componente estuviera claramente definida, es decir, que la aplicación se encargue de la gestión de la información, mientras que el hardware se encarga de la rotación del carrusel y la activación de la alarma.

Como aprendizaje principal, el trabajo permitió comprender cómo se estructuran sistemas distribuidos entre software y hardware, y cómo diseñar aplicaciones móviles que no solo almacenen datos, sino que participen activamente en el control de dispositivos externos.

4 Referencias

Sistemas embebidos e Internet de las Cosas. Sistemas Operativos Avanzados, Universidad Nacional de La Matanza (2025). [Sistemas embebidos e Internet de las Cosas - SOA - Wiki - Unlam](#)

Tiempo Real. Sistemas Operativos Avanzados, Universidad Nacional de La Matanza (2025).
[Tiempo Real - SOA - Wiki - Unlam](#)