

P.E.P.A (Pastillero Electrónico Programable Automático)

Proyecto de Sistemas Electrónicos para Automatización (Microcontroladores)



P.E.P.A (Pastillero Electrónico Programable Automático)

Índice

1. Introducción:	3
2. Objetivos:.....	3
3. Enfoque Práctico	3
4. Resumen Programas.....	4
4.1 Programa de microcontrolador TIVA	4
4.2 Programa de ESP32	4
5. Realización de los programas:	5
5.1 Programa Micro	5
5.2 Programa ESP32:	11
Repository GitHub:	13
Conclusión:	13

P.E.P.A (Pastillero Electrónico Programable Automático)

1. Introducción:

La idea central del proyecto se cimenta sobre una pregunta clave: ¿cómo se puede hacer la vida de las personas mayores más sencilla? Con el paso del tiempo, es común que las personas mayores necesiten responsables que lleven el registro de su salud. En particular la ingesta de medicamentos puede convertirse en un reto, ya sea por problemas de memoria o estado físico en general.

A partir de esta necesidad nace el proyecto de PEPA (Pastillero electrónico Programable y automático). Un proyecto diseñado para facilitar y automatizar la administración de medicamentos mediante indicadores acústicos y un sistema pastillero con servomotores. En su desarrollo se aplican los conocimientos aprendidos en las sesiones teóricas y prácticas de la asignatura, además de un exhaustivo trabajo de investigación por parte de los integrantes del proyecto.

2. Objetivos:

- Monitorización y registro de las pastillas tomadas en el día.
- Realización de interfaz asequible y práctica para todos los públicos.
- Comunicación externa vía wifi con responsable o con propio usuario.
- Reprogramación de la toma de pastillas.
- Crear un sistema que funcione a lo largo del tiempo, siempre que esté conectado.

3. Enfoque Práctico

Este pastillero tiene las siguientes características:

La toma de pastillas se hará con una diferencia de 6h entre pastilla y pastilla de forma predeterminada. A no ser que el usuario programe lo contrario.

En la primera conexión del micro, se le pedirá al usuario añadir manualmente el día y hora mediante comunicación por puerto serie. Para posteriormente preguntar sobre la ingesta o no de la pastilla asignada a esa franja horaria, con el fin de llevar una monitorización desde el momento que se enciende el micro hasta que se desee apagarlo.

Avisa de cuando debe de tomar la pastilla mediante indicadores acústicos, a la hora deseada por el usuario o por la predeterminada por el programa.

P.E.P.A (Pastillero Electrónico Programable Automático)

Además, si se pospone la alarma y no se toma la pastilla en una franja de tiempo predeterminada, se generará un mensaje WhatsApp que avisará al mismo usuario del pastillero o a un familiar que este a cargo, la no ingesta de la pastilla.

En caso de activar tarde el micro (y por consecuente no haber registrado la toma de una pastilla) o de haber cancelado la alarma; el usuario podrá abrir manualmente la compuerta del pastillero necesario.

Al finalizar la toma de pastillas del día se ruega al usuario recargar manualmente los pastilleros mediante petición de una tecla por puerto serie, en caso contrario se saltará a este estado a una hora en concreto con el fin de poder preservar el funcionamiento del pastillero.

4. Resumen Programas

4.1 Programa de microcontrolador TIVA

Se integra una primera fase de interacción con el usuario por UART, donde se le pide rellenar la fecha y hora exacta del día. Es recomendable que esta parte sea realizada por el adulto a cargo, ya que el control del teclado puede ser enrevesada para el abuelo.

Luego, se realiza una interfaz gráfica con la FT800 donde se integran una serie de pantallas que guían de forma lógica al usuario a lo largo del proceso: pantalla de inicio, pulsadores de alarmas...

Aparte se integra una comunicación externa mediante salidas de periféricos GPIO con una ESP32 para el mensaje de WhatsApp.

4.2 Programa de ESP32

Se ha realizado un programa en paralelo con Arduino IDE, donde se programa la ESP32 a WIFI y el servicio de una API de WhatsApp llamada CallMeBot. Este programa realiza una primera conexión al WIFI suministrado con su SSID y su contraseña, luego una segunda conexión al servidor de la API y por último manda un mensaje WhatsApp dependiendo de las señales de entrada obtenidas por el puerto GPIO.

P.E.P.A (Pastillero Electrónico Programable Automático)

5. Realización de los programas:

5.1 Programa Micro

El programa implementa el control completo de un dispensador automático de pastillas con tres tomas diarias: mañana, tarde y noche. El sistema integra una pantalla táctil FT800, tres servomotores que abren cada compartimento de pastillas, y dos módulos ESP para enviar avisos o recordatorios mediante comunicación UART.

Los periféricos que se han habilitado son:

- Puerto K y H: GPIO para la comunicación con la ESP32.
- 3 módulos PWM para el mover los pastilleros mediante servos.
- UART para comunicación inicial de configuración de la fecha y marcar la recarga de las pastillas.
- Timer0 que habilita las interrupciones cada 20 ms.
- Timer1 que genera un parpadeo para las funciones de alarma.
- Pantalla FT800 inicializada mediante SPI, utilizada para dibujar la interfaz y registrar las pulsaciones táctiles de los botones.

En cuanto al funcionamiento principal del proyecto, se realiza mediante una máquina de estados **switch(*estado_actual*)**:

- **Inicio:**

Este estado inicial pide al usuario por pantalla de introducir la fecha mediante UART. La introducción de la fecha es bastante robusta, pudiendo retroceder si el usuario se ha equivocado de carácter o incluso detectar si los caracteres introducidos no son correctos. Luego, se pasa directamente al siguiente estado de configuración.

- **Dia1, Dia2, Dia3, Verificacion_dia, Hora1, Hora2, Verificacion_hora, Min1, Min2, Verificacion_min:**

Esta serie continua de estados sirve para obtener mediante UART: el día de inicialización del micro, introduciendo las 3 primeras letras del día; la hora y el minuto, teniendo que añadir 2 dígitos (de 00 a 23 para la hora y de 00 a 59 para el minuto), y por último la verificación de que estos estén correctos.

Una vez introducida la hora, el programa redirecciona al usuario según la franja horaria en la que se encuentra: de 6 hasta las 12 será por la mañana, de 12 a las 18 será por la tarde y de las 18 hasta las 00 será por la noche. Esto se hace ya que el programa para inicializarse necesita saber en qué momento del día esta.

P.E.P.A (Pastillero Electrónico Programable Automático)

```
Introduce el dia actual (LUN-DOM) :  
LUN  
    Dia valido: Lunes  
Introduzca la hora del dia (0-23) :  
24  
    Error: hora incorrecta  
Introducir de nuevo  
14  
    Hora correcta: 14  
Introducir minuto(0-59) :  
00  
    Minuto correcto: 0
```

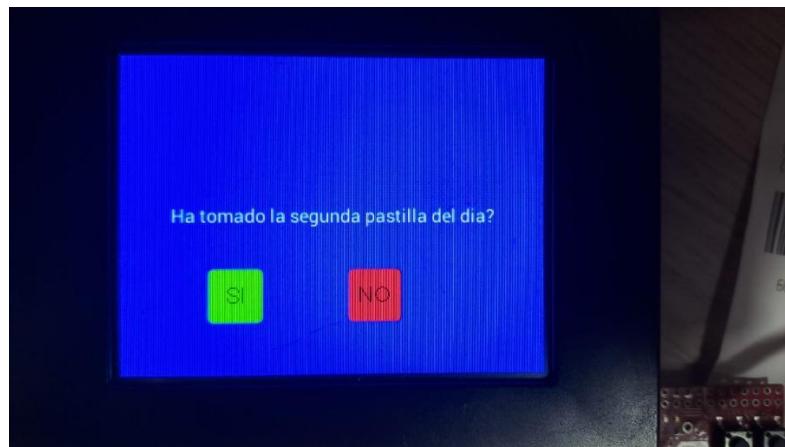
- **Reposo1, Reposo2, Reposo3:**

Como se ha explicado anteriormente, los estados de **Reposo** marcan si el usuario está por la mañana, por la tarde o por la noche.

Aparte, se pregunta por pantalla si se ha tomado la pastilla que corresponde a la franja horaria en la que se encuentra el usuario.

En caso de haber tomado dicha pastilla, el usuario pasará al estado de confirmación de la hora: **MananaOK, TardeOK, NocheOK**.

De lo contrario, el programa redirecciona al usuario a la pantalla principal **Inicio_i**, donde se mostrará visualmente el pulsador de la pastilla que no se ha tomado en rojo.

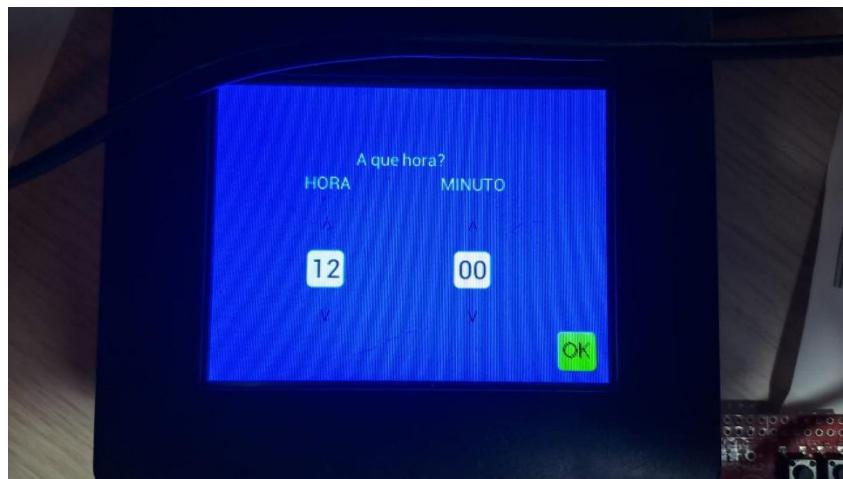


- **MananaOK, TardeOK, NocheOK:**

En este estado, se pregunta al usuario por pantalla la hora en la que se tomó la pastilla correspondiente a su franja horaria, mediante una interfaz interactiva donde el usuario puede aumentar o disminuir las horas y los minutos gracias unas flechas integradas por pantalla.

P.E.P.A (Pastillero Electrónico Programable Automático)

Una vez seleccionada la hora, se recalculará la toma de las siguientes pastillas (diferencia predeterminada de 6h entre pastilla y pastilla). Una vez finalizado se pasará al estado base de la interfaz llamado **Inicio_i**, donde se mostrará visualmente el pulsador de la pastilla que se ha tomado en verde.



- **Inicio_i**

Este estado representa la interfaz principal del programa.

La parte superior marca el día, además de un reloj que marca el paso de las horas, minutos y segundos.

La parte inferior aparecen unos botones **M, T, N** que estarán de un color u otro según en qué estado se encuentre el programa: en rojo si la pastilla de esa franja horaria no se ha tomado, en verde para el caso contrario, o en blanco en caso de aun tener que programar la alarma de la pastilla correspondiente a esa franja horaria.

Si se quiere cambiar la hora programada de las pastillas o abrir el pastillero (en caso de no haber tomado la pastilla), el usuario pulsará el botón correspondiente a la franja horaria que se requiera, pasando a los estados llamados **Manana, Tarde, Noche**.

Si se inicia las alarmas de las pastillas, se saltará a los estados correspondientes de las alarmas: **AlarmaManana, AlarmaTarde, AlarmaNoche**. Las alarmas podrán saltar en cualquier momento de la interacción con la interfaz o de los siguientes estados, guardando en una variable **estado_anterior** el estado anterior previo a la alarma, pudiendo volver a el una vez apagada o cancelada.

A parte este estado nos sirve de comunicación con la esp32 encendiendo los pines K6 y K7 y H1, que mandaran una señal en alto si no se ha tomado la pastilla en un tiempo preinstalado (para simplificar se ha puesto unos 10 segundos, pero podría ser perfectamente configurado).

Por último, se permite al usuario recargar el pastillero manualmente, en el caso de que la última pastilla se haya tomado (**no_noche=0**) y que se apriete la tecla 'r' o 'R' por puerto serie, saltando al estado **Recarga**. En el caso que el usuario no decida recargar, ya sea por olvido o por incapacidad, a partir de una hora predeterminada de la noche se saltará a dicho estado.

P.E.P.A (Pastillero Electrónico Programable Automático)



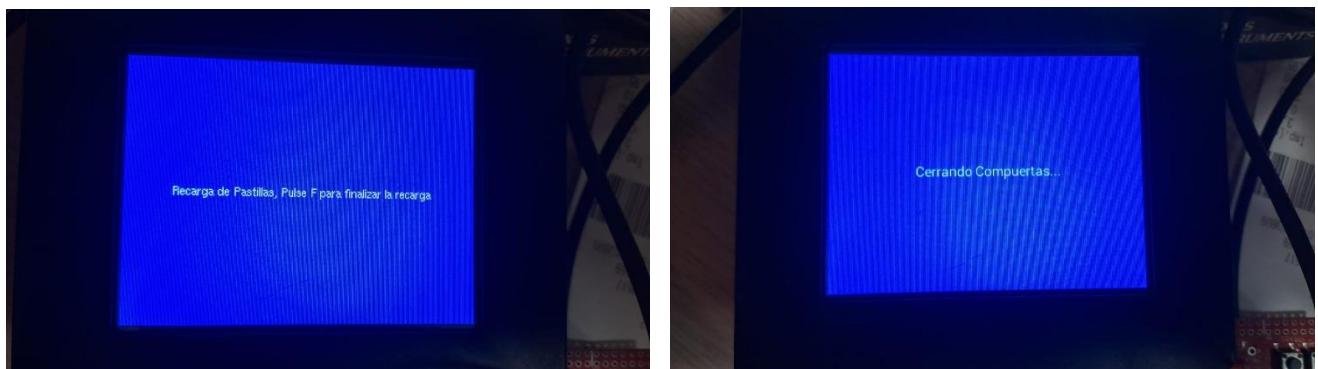
- **Recarga, Fin_Recarga:**

En este estado se permite al usuario recargar manualmente.

Se muestra tanto por pantalla como por puerto serie un aviso de recarga para el usuario, avisando de las teclas a pulsar para la debida recarga. En este caso, si se manda por UART los caracteres **m**, **t** y **n**, se abrirán los respectivos pastilleros, permitiendo su recarga.

Cuando el usuario plazca, puede enviar de nuevo por puerto serie la tecla **f** para finalizar el proceso, donde se cerrarán los pastilleros de uno en uno. Posteriormente se vuelve al estado **Inicio_i**, continuando con el correcto funcionamiento del programa.

Este estado de interacción con el usuario es recomendable que sea realizada por el adulto a cargo, ya que puede suponer demasiado complicado para el abuelo.



P.E.P.A (Pastillero Electrónico Programable Automático)

- **Manana,Tarde,Noche:**

En estos estados se despliega una pantalla donde se muestra un botón para abrir el pastillero manualmente, un botón para editar la alarma correspondiente y un botón de cancelación en caso de querer volver a la pantalla inicial *Inicio_i*.

El botón para abrir manualmente el pastillero estará en gris o en verde según se deje o no abrir. Es decir, que se haya cumplido el tiempo para tomar la pastilla o que no se haya tomado en un primer momento. En caso de estar disponible se saltaría a los siguientes estados según la franja horaria en la que se encuentre el usuario: **AbrirManana, AbrirTarde, AbrirNoche**.

En cuanto al botón para configurar la hora de la alarma, en este se representará la hora predeterminada o programada por el usuario. En caso de querer modificar la alarma y pulsar el botón, se pasará a los siguientes estados de configuración: **configHora1,2 o 3**.



- **AbrirManana, AbrirTarde, AbrirNoche:**

Para acceder a estos estados el usuario debe:

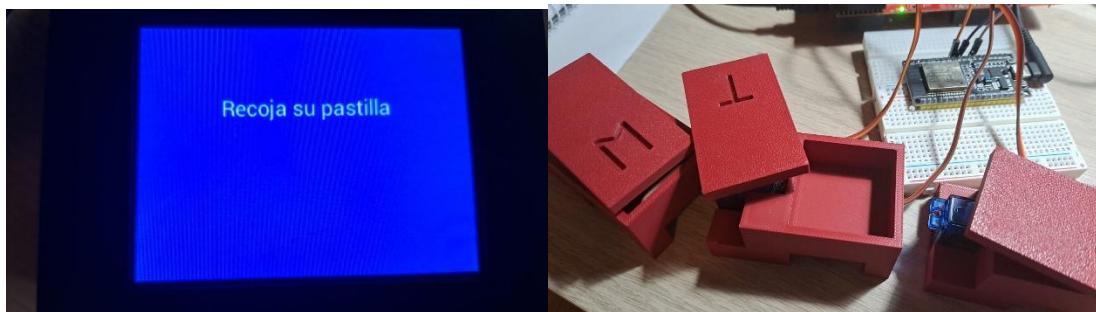
- Haber cancelado la alarma. En ese caso, se le permite abrir manualmente la compuerta del pastillero.

- No haber tomado la pastilla correspondiente, una vez inicializado el micro.

Dentro de los estados, se abren las cajetillas mandando señales de PWM a los servos durante 5 segundos. Aparte aparece un mensaje por pantalla indicando al usuario que puede recoger su pastilla.

Una vez finalizada la recogida, se vuelve a la interfaz principal *Inicio_i*.

P.E.P.A (Pastillero Electrónico Programable Automático)

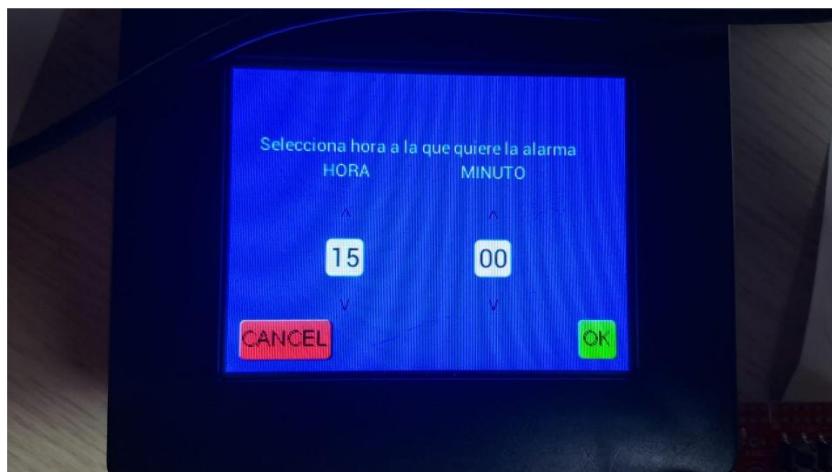


- **ConfigHora1, ConfigHora2, ConfigHora3:**

En estos estados, se despliega una pantalla en donde el usuario puede aumentar o disminuir las horas y los minutos gracias unas flechas integradas por pantalla.

No obstante, hay un rango de operación de 6 horas, es decir, el usuario no podrá programar la alarma fuera de ese rango: de 6 a 12 para la mañana, de 12 a 18 para la tarde y de 18 a 00 para la noche.

Luego de configurar o no la hora, el usuario podrá o bien cancelar (sin guardar la hora programada) o bien aceptar (guardando la hora programada), gracias a los botones de las esquinas inferiores de la pantalla. Una vez apretado alguno de los 2 botones, se volvería al estado correspondiente **Manana, Tarde o Noche**.



- **AlarmaManana, AlarmaTarde, AlarmaNoche.**

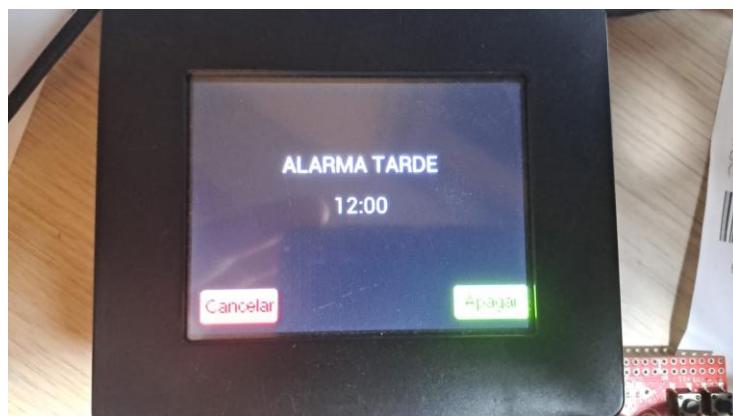
P.E.P.A (Pastillero Electrónico Programable Automático)

En estos estados se generará una alarma sonora y un parpadeo avisando de que es la hora de tomar la pastilla, además de dos botones en las esquinas inferiores de la pantalla.

La alarma sigue indefinidamente a menos que:

- El usuario apague la alarma. En cuyo caso, saltara a los estados **AbrirManana**, **AbrirTarde**, **AbrirNoche**, abriendo la compuerta del pastillero.
- El usuario cancela la alarma. En cuyo caso, se vuelve al estado anterior, se pone en rojo el botón correspondiente a la pastilla por tomar y se deja al usuario abrir manualmente el pastillero.

En caso de cancelar la pastilla, se inicia un temporizador con la variable **tiempo_inicio**, para poder contar el tiempo que pasa entre cancelar la alarma y que el usuario abra manualmente el pastillero. Como se ha comentado anteriormente, si ese tiempo fuera superior a un tiempo predeterminado, se enviaría un mensaje por WhatsApp al adulto a cargo.



5.2 Programa ESP32:

El código implementa un sistema de notificación de toma de medicación basado en un ESP32 conectado a WiFi, que detecta la activación de tres entradas digitales correspondientes a las pastillas de la mañana, tarde y noche, enviando alertas vía WhatsApp usando el servicio CallMeBot.

Arquitectura y módulos del microcontrolador

- **WiFi:** Se configura con SSID y contraseña para establecer conexión a la red local, lo que permite la comunicación con el servicio CallMeBot mediante HTTP.
- **GPIO:** Tres pines del ESP32 (**pinManana**, **pinTarde**, **pinNoche**) se configuran como entradas con resistencia pull-down interna para detectar el estado de cada pastilla.
- **Serial:** Se utiliza para depuración y seguimiento del estado de la conexión Wifi y del envío de mensajes.

P.E.P.A (Pastillero Electrónico Programable Automático)

Funcionamiento principal

Inicialización (setup):

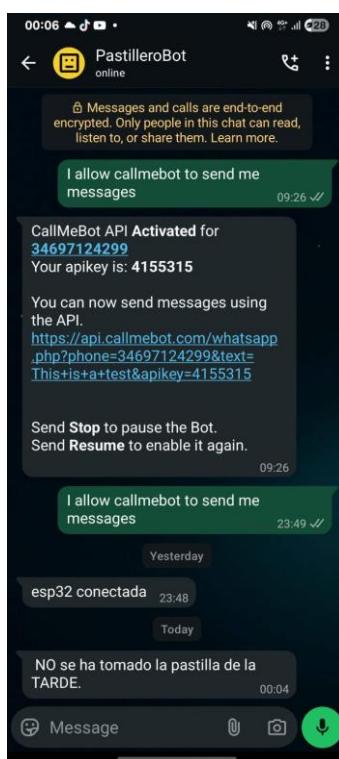
Se inicia la comunicación serie. Se configuran los pines de entrada con INPUT_PULLDOWN. Se conecta a la red Wifi, mostrando progreso por Serial. Al establecer conexión, envía un mensaje de confirmación de conexión vía WhatsApp.

Bucle principal (loop):

Se leen continuamente los estados de los pines correspondientes a cada pastilla. Se detectan cambios de estado comparando el valor actual con el valor anterior (**lastManana**, **lastTarde**, **lastNoche**). Si se detecta que un pin pasa de LOW a HIGH (lo que indica que la pastilla no se ha tomado), se envía un mensaje de alerta vía CallMeBot especificando la franja horaria correspondiente: mañana, tarde o noche. Se actualizan las variables de estado previo para poder detectar futuros cambios.

Comunicación con CallMeBot:

La función **sendMessage()** construye una URL HTTP con el número de teléfono, API key y texto codificado. Se realiza una petición POST y se comprueba el código de respuesta para confirmar el envío. Los mensajes incluyen el detalle de la pastilla que no se ha tomado, lo que permite a un cuidador recibir alertas en tiempo real.



P.E.P.A (Pastillero Electrónico Programable Automático)

Repositorio GitHub:

Se adjunta un enlace GitHub con el repositorio generado por los integrantes del grupo, donde se ven reflejadas dudas externas a la memoria como puede ser la guía de instalación para la esp32, los planos de impresión del hardware utilizado etc.

https://github.com/Curro110/Mirco_SEPA.git

Conclusión:

El desarrollo de este proyecto ha sido fundamental para observar un problema real de la sociedad y cómo podríamos solucionarlo. Sabemos que el prototipo no es un producto que podríamos lanzar al mercado, pero nos podría sentar las bases para poder ampliar y perfeccionar funcionamientos que aseguren que un paciente se toma la pastilla y su control sea óptimo.

Queremos destacar que hemos trabajado conjuntamente para poder desarrollar este proyecto y hemos utilizado herramientas profesionales como GIT para poder mantener un seguimiento de trabajo conjunto.

Creemos firmemente que nuestro proyecto expresa los conocimientos técnicos que hemos adquirido a lo largo de la asignatura, como la intención de expandir nuestros conocimientos e integrar lo aprendido.