**Universidad Técnica Estatal de Quevedo**

**Facultad de Ciencias de la Ingeniería**

**Ingeniería en Sistemas**

**Dispensador de medicinas para adultos mayores**

**DOCENTE:**

**Ing. Gleiston Guerrero.**

**UNIDAD DE APRENDIZAJE:**

**Aplicaciones distribuidas**

**Estudiante:**

**María Mercedes Fernández Coello**

**José Luis Zambrano Pishculla**

**Washington Augusto Pareja Galarza**

**Quevedo – Los Ríos – Ecuador**

**2019**

**CONTENIDO**

[1. INTRODUCCIÓN 1](#_Toc33826579)

[2. OBJETIVOS 2](#_Toc33826580)

[**2.1** **OBJETIVO GENERAL** 2](#_Toc33826581)

[**2.2** **OBJETIVOS ESPECIFICOS** 2](#_Toc33826582)

[3. JUSTIFICACIÓN 3](#_Toc33826583)

[4. TRABAJOS RELACIONADOS 4](#_Toc33826584)

[5. METODO DE DESARROLLO 8](#_Toc33826585)

[7.1. Interfaces 27](#_Toc33826586)

[27](#_Toc33826587)

[7.2. Desarrollo 28](#_Toc33826588)

[6. CONCLUSIONES 41](#_Toc33826589)

[7. RECOMENDACIONES 41](#_Toc33826590)

[8. REFERENCIAS 42](#_Toc33826591)

[9. ANEXOS 43](#_Toc33826592)

# INTRODUCCIÓN

El Proyecto presente tiene como objetivo proponer un sistema distribuido basado en IOT que tenga un enfoque innovador permitiendo facilitar la toma de medicamentos diarios a los pacientes que requieren de tratamientos o sus medicinas de forma diaria. Delegar la responsabilidad del control y gestión de estos tratamientos al mismo sistema.

Por este motivo, se trata de busca, crear una forma menos compleja y màs precisa a los pastilleros convencionales usados en la toma de medicamentos. A través del sistema se podrá programar horarios que serán notificados a la persona encargada para que el paciente ingiera la pastilla que expulsara el dispensario creado específicamente para el sistema, el dispensador dispondrá la pastilla a una fecha y hora programada, también será capaz de controlar si la medicación fue retirada del dispensador y quien la obtuvo.

El dispensador, dispondrá de cuatro contenedores independientes que almacenaran las pastillas de acuerdo a la necesidad del paciente. También servirá como centro de monitorización del entorno y de control de las acciones del usuario para la toma de la medicación. Además, se podrán configurar diversos parámetros como el número de medicamentos del pastillero, se avisa al paciente y la persona que le lleve la medicación al paciente será identificada por medio de la cámara como un FaceID de esta forma se confirmará que ha sido extraído la medicina.

El sistema está orientado en primera instancia a pacientes de avanzada edad que usarían el dispensador y familiares o cuidadores que gestionan y monitorizan el dispensador a través de la aplicación. Por ello, el sistema permite no solo programar horarios de medicación y monitorizar, sino que también será capaz de mantener informados a los usuarios de anomalías y situaciones de emergencias.

De esta forma el sistema a desarrollar se destaca, al permitir la interacción del paciente y del cuidador de eventualidades que se presenten, teniendo como fin el ayudar a personas de avanzada edad a la toma de medicamento de forma automática; en el caso de otros sistemas o trabajos relacionados su desarrollo ha sido en la parte de una aplicación que realice recordatorios para la toma de medicamentos. Destacando el sistema desarrollado que está enfocado en la estandarización, robustez, comunicación, flujo de datos y seguridad que demanda las soluciones IoT.

# OBJETIVOS

## **OBJETIVO GENERAL**

* Implementar un dispensador de medicina como sistema distribuido basado IOT para la toma de medicamentos de los adultos mayores.

## **OBJETIVOS ESPECIFICOS**

* Crear un sistema basado en IOT para controlar que el adulto mayor haya tomado su medicina (pastillas unitarias).
* Determinar el sistema que está realizado con otros trabajos mediante una revisión bibliográfica para establecer los beneficios y las mejorías del sistema desarrollado.
* Desarrollar una aplicación móvil para gestionar los recursos del dispensador e interactuar con el adulto mayor.

# JUSTIFICACIÓN

El principal público hacia el que ira dirigido el sistema es al de las personas mayores, debido a que son las susceptibles y las que más dificultades presentan a la hora de tomar la medicación diaria principalmente por el olvido de su toma. Es importante por motivos que el sistema podría ser utilizado por cualquier otra persona que desee automatizar la toma de sus medicamentos o traspasar la responsabilidad del control de la medicación al sistema.

El sistema que se creará tiene como principal objetivo automatizar la toma de medicamentos diarios y ayudar a las personas en este proceso, sobre todo a personas de avanzadas edad.

Para ello, el sistema estará compuesto por un dispensador físico que será el encargado de dispensar las pastillas y de avisar mediante luces y sonidos de la toma de medicación y una aplicación móvil de gestión.

Dentro del merado existen multitud de dispositivos que pretenden dar solución al problema que supone el olvido de la toma de medicación, pero la gran mayoría presentan un gran número de limitaciones y restricciones que los hacen poco útiles. Lo que se pretende con este proyecto es crear un sistema novedoso que complete y extienda las funcionalidades de los actuales dispositivos existentes en el mercado y proporcione u dispositivo realmente útil, fácil de usar y, sobre todo, que contribuya a no olvidad la toma de medicación a sus usuarios.

# TRABAJOS RELACIONADOS

Se han realizado diversas investigaciones relacionadas con el desarrollo de un sistema dispensador de medicinas o muy relacionados con el tema. Dentro de estos trabajos están:

J.Pérez, C.Jesús[1], presentaron una alternativa tecnología que permite conectar a las familias con sus mascotas, utilizando un dispensador de comida a Internet permitiendo dar una ración de comida en un tiempo determinado. Destacando nuestro sistema en permitir dar apertura a un dispensador de medicina caso que permite mayor viabilidad y que sea atractiva para su uso. Es posible la adaptabilidad que se le puede dar para otras interacciones tanto al dispensador de medicamentos como a la aplicación, además el desarrollo de los autores mencionados se ausento de realizar una aplicación móvil.

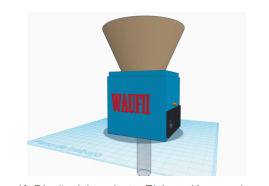


Figura 1 Diseño del producto – dispensador de comida

J. Cabañeros Blanco[2], este trabajo consiste en un sistema automático que ayude en la toma de medicación diario a las personas, pero con la restricción o limitante que dispensara 2 tipos de medicamentos distintos o dos tratamientos diferentes, está diseñado dentro de una ruleta que garantiza la automatización de la salida del medicamento, esto impide mayor manipulación con la medicaciones. Es ahí donde destacamos el desarrollo de nuestro sistema que permite dispensar todo tipo de medicamentos sin restricciones y que además permite la programación de horarios correspondientes al dispensador y su paciente, otra funcionalidad importante de gran valor para nuestro sistema que destaca es la emisión de notificaciones.

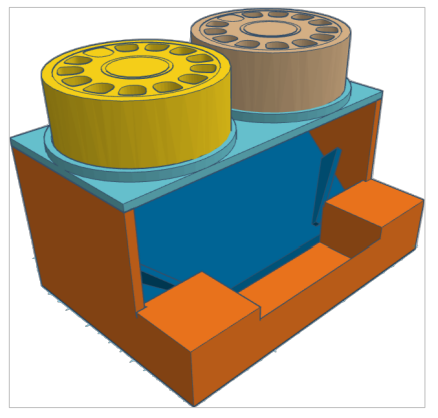


Figura 2 Boceto del dispensador de pastillas conectado

M. García, González [3], analizan la aportación de las tecnologías de la información y las comunicaciones - TIC - en la prevención o reducción de los errores que se producen durante la administración terapéutica de fármacos. Se analiza el papel que desempeña el paciente, tanto en lo relativo al cumplimiento de los tratamientos prescritos como en su rol de usuario de las soluciones técnicas diseñadas para tal fin, y las consecuencias que podría tener en el actual modelo de relación entre los profesionales sanitarios y los pacientes. Este trabajo acoto mucho al determinar dentro de nuestro sistema la debida prevención de dispensar la medicina correspondiente al horario establecido.

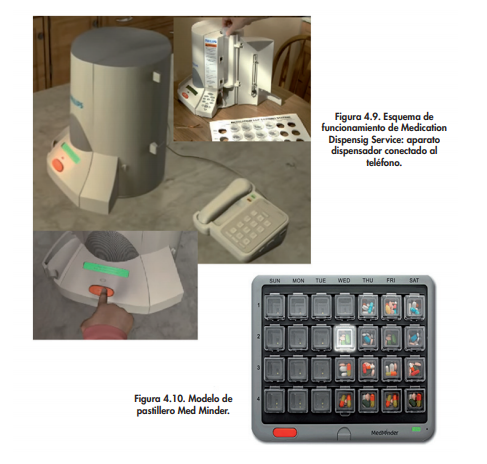


Figura 3 Esquema de funcionamiento de medicación de medicinas – modelo de pastillero

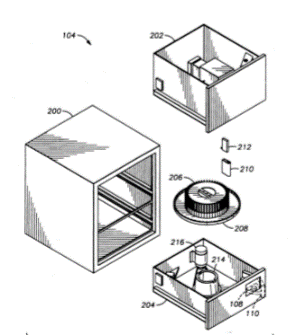
Gennaro [4], MedMinder ofrece dispositivos inalámbricos que proporcionan un todo en uno gestión de medicamentos y un sistema de alerta de emergencia. Este recordatorio y monitoreo de medicamentos las 24 horas El sistema es simple, conveniente y confiable. Los dispensadores de pastillas MedMinder reflejan a los organizadores de pastillas de farmacia, esto Crea una familiaridad inmediata. El módem celular incrustado en la caja permite una programación y monitoreo sin interrupciones para apoyar al paciente y notificar a todas las partes interesadas. Este sistema es muy parecido al sistema que se ha desarrollado, además posee un dispensador mas adaptable, pero a este se le debe extraer la medicina manual a diferencia de nuestro desarrollo que sea expulsado de forma automática.



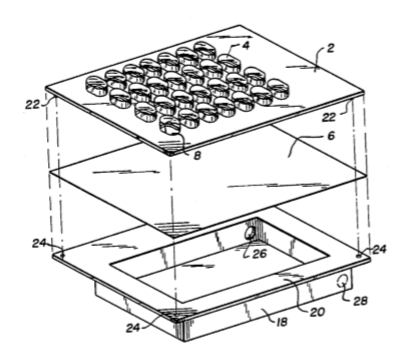
Figura 4 Dispensador de pastillas MedMinder

Berry, John F [5], este trabajo es una patente la cual consiste en un sistema operado al vacío para dispensar de forma individual las medicinas, las medicinas estarán almacenadas en un granel. Incluye esta patente una computadora que proporciona la interfaz del usuario, por otra parte, los gráneles o contenedores pueden ser recargados de forma individual.

Es considerable destacar que, a diferencia de este trabajo, el que se ha desarrollado tiene grandes beneficios entre estos el reconocimiento facial y un dispensador más accesible para el usuario contando con una aplicación móvil que permite la dispensación de la medicina desde esta app.



Pamela J. White [6], la patente del autor antes mencionado se trata de un dispositivo dispensador de píldoras que tiene un varios compartimientos abiertos, están dados en formas de filas, las píldoras estarán en estos compartimientos y se podrán abrir a través de un sello, y adentro estará una bandeja de salida para las píldoras. Lo considerable y mejorado del dispensador elaborado es la expulsión de forma automática de las pastillas a comparación de esta patente que es de forma manual.



# METODO DE DESARROLLO

Test-Driven Development Methodology for IoTS (TDDM4IoTS) es una metodología de desarrollo para la Internet de los objetos (IO) basada en sistemas (IOTS) que recoge ideas de varios de los paradigmas de desarrollo de software más destacados en la actualidad, como la Ingeniería Basada en Modelos (MDE) y el Desarrollo Basado en Pruebas (TDD), además de incorporar los principios que rigen las metodologías ágiles de desarrollo de software, como SCRUM y XP [7].

Esta metodología consiste principalmente en once fases, cuyo orden de aplicación puede ser establecido por el equipo que desarrollará el proyecto en cuestión, los autores sugieren un orden a seguir, así como las herramientas de software existentes que podrían ser utilizadas como soporte para la obtención de los correspondientes entregables en cada fase.

Las fases del ciclo de vida de TDDM4IoTS, que se muestran en la Figura 9, tienen en cuenta el desarrollo de todos los tipos de IoTS, por lo que el orden y la frecuencia de aplicación, así como la asignación de recursos para cada fase dependerán tanto de la naturaleza del proyecto como de los conocimientos, habilidades, experiencia y número de miembros del proyecto. Estas fases se repetirán de forma iterativa para cada entrega [7]. Sin embargo, en el desarrollo de algunos entregables, puede no ser necesario aplicar algunas de estas fases (dibujadas con una línea discontinua en la Figura 9).

Figura 9. Fases del ciclo de vida de TDDM4IoTS

Evaluación de los resultados

Mantenimiento

Despliegue de hardware y software

Refinamiento del software

Refinamiento del modelo

Generación de software

Generación de pruebas

Generación y adaptación de modelos

Análisis detallado de los requisitos

Diseño de la capa tecnológica

Análisis preliminar

FUENTE: Guerrero-Ulloa, Hornos, Rodríguez-Domínguez, 2019

ELABORADO: Villafuerte y Yánez

Las fases se describen a continuación [7]:

***(1) Análisis preliminar.*** El objetivo de esta fase es obtener un estudio de viabilidad del sistema (completo) en sus aspectos tecnológicos, económicos y operativos, así como un análisis del contexto en el que se desplegará el sistema y de la interacción con el usuario final, basado en los requisitos y objetivos expresados por el cliente. Las actividades que se llevarán a cabo podrán incluir:

* *Análisis de requerimientos.* Este análisis determina dos tipos de requisitos: (i) funcionales, que son las especificaciones del cliente (lista de entregables), determinando su prioridad de implementación, y (ii) no funcionales, también llamados atributos de calidad, tales como escalabilidad, intrusismo, estética ambiental (despliegue, apariencia, etc.).
* *Análisis de tecnología*. Esto determina la tecnología que se va a utilizar y que cumple con los requisitos del sistema: (i) Recursos de hardware ya disponibles; (ii) Hardware existente, teniendo en cuenta sus características y costes; (iii) Herramientas (software) para la configuración del hardware; (iv) Herramientas tanto para el desarrollo de software como para la gestión del almacenamiento; (v) Hardware de terceros para tareas específicas, y (vi) Desarrollo de hardware de primera mano, si es necesario y factible.
* *Análisis del medio ambiente.* El cliente puede especificar las características del entorno en el que se desplegará el sistema. Por ejemplo, puntos de suministro de energía disponibles y/o viables, redes de comunicación de datos - acceso a Internet, métodos de interacción preferidos por el cliente, etc.
* *Análisis de viabilidad.* Entre los tipos de factibilidad a analizar se encuentran: i) Viabilidad técnica, para la que se responderían las preguntas: a) ¿Existen las tecnologías necesarias para desarrollar el proyecto? b) ¿Se dispone de personal capacitado para desarrollar el sistema? c) ¿Se puede desarrollar el sistema? (ii) Viabilidad económica, donde estaría la pregunta a responder: ¿Existe un presupuesto adecuado para desarrollar el proyecto? iii) La viabilidad operacional, que es importante para que el sistema siga funcionando después de su aplicación, por lo que es preciso responder a estas preguntas: a) ¿Será posible instalar el sistema una vez terminado? b) ¿Podrá funcionar el sistema con los recursos disponibles? c) ¿Existen las garantías necesarias para que el sistema siga funcionando una vez instalado? d) ¿Tendrán los botes un mantenimiento debidamente programado?

***(2) Diseño de la capa tecnológica.*** El objetivo de esta fase es obtener el primer diseño del sistema global que sirva de guía para los equipos de desarrollo. Esto es muy importante, dado que la IO implica tecnologías emergentes y heterogéneas, y hasta ahora es difícil encontrar un profesional que pueda dominar todas las tecnologías implicadas en el desarrollo de la IOTS. Para el diseño del sistema, se pueden utilizar herramientas de diseño de circuitos que puedan representar de la forma más clara posible los elementos que se han determinado para el proyecto. Por ejemplo, si se utilizan placas Arduino, se pueden utilizar herramientas en línea como Circuito.io o Fritzing. El equipo de desarrollo puede complementar los diseños obtenidos. Si es necesario, el diseño resultante puede ser actualizado al final de cada entrega. Este será uno de los documentos de mayor interés para todos los equipos de desarrollo del proyecto. Por lo tanto, siempre debe ser accesible para todos ellos.

***(3) Análisis detallado de los requisitos.*** El objetivo de esta fase, que se ejecutará para cada entregable del sistema, es obtener los requisitos detallados del entregable a desarrollar. Además, se pedirá al cliente que describa las pruebas junto con los desarrolladores, para reducir las ambigüedades. Para la especificación de los requisitos, se recomienda utilizar herramientas comprensibles para todos los grupos de interés, considerando al cliente como uno de ellos. Una de las notaciones a utilizar podría ser UML, con sus diferentes herramientas, como casos de uso y casos de uso semiestructurados, utilizando plantillas preestablecidas, buscando eliminar ambigüedades, además de otras herramientas, como diagramas de estado y de despliegue, que ayuden a entender los requerimientos proporcionados por el cliente.

***(4) Generación y adaptación de modelos.*** El propósito de aplicar MDE en TDDM4IoTS es reutilizar modelos y así mejorar la productividad del equipo de desarrollo. Por lo tanto, en esta fase se generarán nuevos modelos o se adaptarán los modelos existentes. El uso de modelos abstractos o al menos minimiza los aspectos de heterogeneidad de las tecnologías, permitiendo una buena comunicación entre los equipos de desarrollo y los clientes. Además, dependiendo de las herramientas de software utilizadas para el modelado, el software puede ser generado automáticamente a través de modelos, desde modelos abstractos hasta modelos específicos. Uno de los modelos es el diagrama de clases, que se utiliza para generar la base de datos.

***(5) Generación de la prueba.*** TDDM4IoTS sigue el paradigma TDD, por lo que debe generar las pruebas que el software debe superar para asegurar la calidad del sistema. Las pruebas se pueden agrupar en dos grupos: (1) Las pruebas escritas por los desarrolladores, dentro de las cuales hay: a) pruebas unitarias, que son las más exhaustivas, para examinar el funcionamiento completo de una función, es decir, se comprueba si la función produce los resultados que debería producir e incluso si admite las excepciones que puedan surgir, y b) pruebas de integración, que también incluyen pruebas de sistemas. Y (2) pruebas documentadas por el cliente, que son básicamente pruebas de aceptación, incluyendo pruebas funcionales.

***(6) Generación de software.*** Esta fase se basa en modelos y pruebas. El desarrollador debe escribir/generar el código de las pruebas a superar. Los nuevos modelos generados y/o los modelos existentes adaptados forman parte de la documentación del sistema. TDDM4IoTS propone hacerlo antes de la generación de código, ya que existen herramientas que, basadas en los modelos, ayudan en esta tarea. Una vez generado el software, se comprueba que pasa las pruebas correspondientes, y luego finaliza esta fase. El resultado de esta fase es el software probado y funcionando, aunque casi siempre tendrá que ser refinado más tarde.

***(7) Refinamiento del modelo.*** Los modelos UML facilitan el refinamiento del modelo. Tenga en cuenta que la solución encontrada es una solución funcional, pero no necesariamente una solución óptima. Por lo tanto, es importante hacer los ajustes y refinamientos necesarios, tales como mejorar la robustez, escalabilidad o reutilización de los entregables que conforman los IoTS a desarrollar. El resultado de esta fase será el modelo definitivo, en base al cual se generará el código del sistema. Las herramientas recomendadas para realizar las tareas de esta fase son las mismas que en la fase de generación y adaptación de modelos.

***(8) Refinamiento del software.*** El trabajo de los desarrolladores en esta fase será garantizar la calidad del software, eliminando redundancias y haciendo que el software sea fácil de mantener. Las herramientas que apoyarán esta actividad serán las proporcionadas por el IDE elegido. Debe asegurarse de que el software final cumple las especificaciones de un código limpio.

***(9) Despliegue de hardware y software.*** Una vez que el software ha sido probado (simulado), se desarrolla e implementa en los dispositivos y recursos a utilizar en el sistema, confirmando el cumplimiento de los requisitos finales negociados entre el cliente y el equipo de desarrollo. En este punto, y para el primer entregable, tanto el sistema de almacenamiento de información como las aplicaciones que servirán para la interacción usuario-sistema ya estarán configuradas. Para las entregas posteriores, se realizarán los cambios necesarios en esta infraestructura ensamblada para añadir las nuevas entregas. Dado que los resultados posteriores dependen de la tecnología ya instalada, debe garantizarse el funcionamiento del (nuevo) sistema integrado antes de continuar el proceso de desarrollo. Las herramientas que se utilizarán en esta fase dependerán de la tecnología utilizada (ordenadores de placa única, sensores empotrados...).

***(10) Evaluación de los resultados.*** Una vez finalizado el desarrollo del entregable (tuvo que pasar las pruebas necesarias para garantizar su funcionamiento), las pruebas de integración, las pruebas del sistema y, por supuesto, las pruebas funcionales deben volver a realizarse en esta fase.

***(11) Mantenimiento.*** IoTS combina la complejidad del mantenimiento de software con la menor complejidad del mantenimiento de hardware. Si un componente físico de IoTS (por ejemplo, sensor, actuador, computadora de a bordo, entre otros) falla, es reemplazado por otro dispositivo similar. Sin embargo, si los requisitos del software cambian, el código tiene que ser modificado, dado que no tiene piezas de repuesto. Además, los IoTS necesitan un mantenimiento operativo constante (baterías o líneas eléctricas, conectividad, entre otros).

**Resultados de la metodología**

Los resultados que se han obtenido en cada fase de la metodología TDDM4IoTS para llevar a cabo esta investigación se detallan a continuación:

1. **Análisis preliminar.**
   1. ***Análisis de requerimiento***

Mediante la búsqueda de información y conocer como es el funcionamiento de los dispensadores, se obtuvieron varios problemas para que un adulto mayor acceda a su medicina o incluso algún medio que permita dar aviso de injerir algún medicamento en particular, por lo tanto, se han determinado como análisis de requerimiento las especificaciones del cliente, las cuales son las siguientes:

1. El sistema dispensador de medicina basado en aplicaciones distribuidas con IoT que permite dispensar las medicinas de forma automática mediante una aplicación móvil y con el reconocimiento facial, este mismo sistema que realiza las siguientes acciones:

1. Personalización de dosificación dependiendo del horario que corresponda, expulsará el medicamento
2. Los horarios son creados por el enfermero para cada paciente o adulto mayor que tenga a su cuidado.
3. Se notifica a qué hora se dispensará el medicamento para cada paciente.
4. Expulsar las medicinas por medio de la aplicación móvil o por el reconocimiento facial de las personas autorizadas.
5. La aplicación permitirá el registro de cuidadores y a los cuidadores registrar a sus pacientes.
   1. ***Análisis de tecnología***

Una vez determinado los requerimientos de sistema, es necesario determinar la tecnología que se emplea para llevar a cabo el desarrollo de este prototipo, los mismo que se detallan a continuación:

1. Recursos de hardware

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Cantidad | Hardware | Característica/Función | Costo |
| 1 | 1 | Arduino Uno | Memoria de instrucciones, código o instrucciones para el manejo del dispensador | $15.50 |
| 2 | 4 | Servomotores | Mecanismo para la expulsar la pastilla hacia afuera | $40.00 |
| 3 | 1 | Raspberry Pi3 Modelo B+ | Servidor local para el reconocimiento facial | $52.00 |
| 4 | 1 | PiCamera | Detectar rostros | $28.20 |
| 5 | 1 | Protoboad | Puente para voltaje y neutro | $3.75 |
| 6 | 1 | Módulo Bluetooth HC-06 | Comunicador entre Arduino y android studio | $7.00 |
|  |  |  | **Total** | **$147.45** |

1. Recursos de software

|  |  |  |
| --- | --- | --- |
| Software | Función | Licencias |
| Arduino 1.8.10 | Codificación para la operación de los servomotores y Modulo Bluetooth | Gratuito |
| Android Studio | Programación de la aplicación del dispensador de medicinas | Gratuito |
| Raspbian | Programación para el reconocimiento facial | Gratuito |
| Postgres | Base de datos | Gratuito |
| Microsoft Word | Desarrollo de la documentación | Pago |

1. Recursos de terceros

|  |  |
| --- | --- |
| Hardware |  |
| Madera |  |
| Clavos |  |

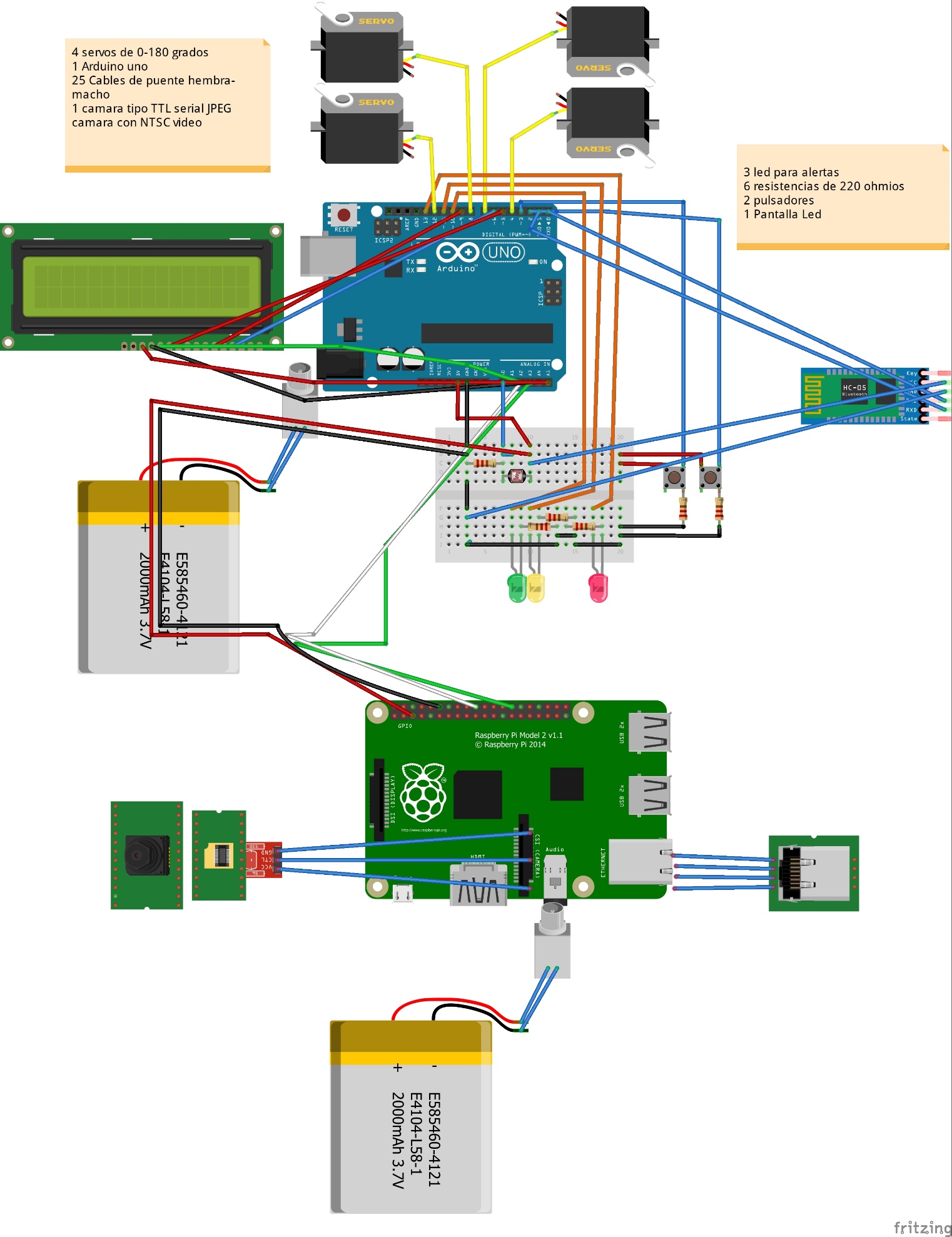
* 1. ***Análisis de medio ambiente***

El entorno donde se implementa el dispensador de medicamento es cerca del área donde este el paciente ya sea en una habitación o en un área donde sea accesible para el uso del adulto mayor, la altura está a disposición del usuario. De la misma forma el usuario debe estar a 10 metros máximo para expulsar las pastillas desde la aplicación móvil.

* 1. ***Análisis de viabilidad***

Para la realización de este proyecto se contaron con las tecnologías necesarias como lo principal que fue el dispensador médico y los materiales como el Arduino, servomotores, el bluetooth y el desarrollo de la aplicación móvil, de la misma forma se contaron con los integrantes necesarios para su total elaboración, los materiales que se utilizan son totalmente programables permitiendo su respectivo mantenimiento o actualización.

El desarrollo del sistema es plenamente desarrollable es decir que no existen alguna tecnología u otro medio que impida su viabilidad, los costos no son altos para el nivel de gestión que posee el dispensador de medicina; el mismo que es manejable y puede ser instalado en cualquier área.

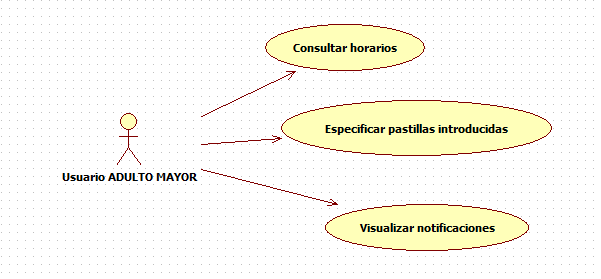
1. **Diseño de la capa de tecnología**
2. **Análisis detallado de los requisitos**

Los actores que interactúan con el sistema pueden ser actores primarios encargados de interactuar con el sistema o secundarios, sobre los que le sistema actúa, los distintos actores que intervienen dentro del sistema son:

* ***Usuario cliente Adulto Mayor:*** es el usuario identificado en el sistema como cliente o paciente y tendrá acceso a dispensar el medicamento que le corresponda en el horario establecido.
* ***Usuario administrador:*** es el usuario identificado en el sistema como administrador y que puede añadir, consultar, modificar y eliminar usuarios registrados y horarios, además puede programar, monitorizar, configurar y modificar todo con respecto al dispensador.
* ***Dispensador:*** es el componente hardware encargado de dispensar las medicinas, además de procesar, ejecutar, enviar y recibir parámetros del sistema

Los casos de usos están directamente relacionados con los requisitos del sistema y se detallarán por medio de diagramas y descripciones.

***Caso de uso del actor Usuario Adulto Mayor***

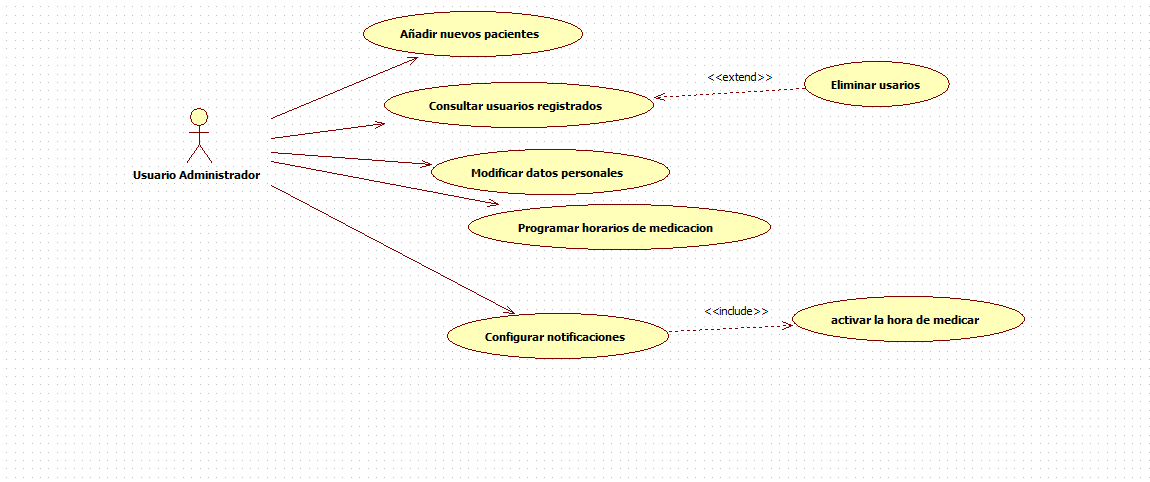


|  |
| --- |
| *Nombre del caso de uso* |
| Consultar horarios |
| *Descripción* |
| El usuario Adulto mayor iniciado en sesión consulta los horarios programados de toma de medicamentos. Estos horarios muestran la fecha y hora a la que se dispensarán los medicamentos. |

|  |
| --- |
| *Nombre del caso de uso* |
| Especificar pastillas introducidas |
| *Descripción* |
| El usuario Adulto mayor iniciado en sesión dispensará las pastillas del tratamiento que estén programadas para ser expulsadas. |

|  |
| --- |
| *Nombre del caso de uso* |
| Visualizar notificaciones |
| *Descripción* |
| El usuario Adulto mayor iniciado en sesión le llegaran las notificaciones de alarmas en la aplicación móvil. |

***Caso de uso del actor Usuario Administrador***



|  |
| --- |
| *Nombre del caso de uso* |
| Añadir nuevos pacientes |
| *Descripción* |
| El usuario administrador iniciado en sesión añade a los nuevos pacientes que estén a su cuidado |

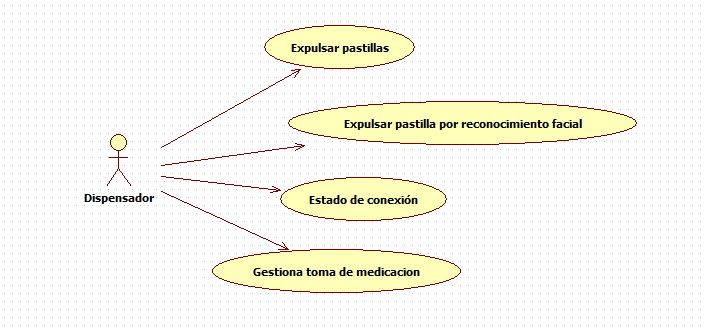
|  |
| --- |
| *Nombre del caso de uso* |
| Consultar usuarios registrados |
| *Descripción* |
| El usuario administrador iniciado en sesión consulta los usuarios registrados; correos, nombres, enfermedad entre otros. |

|  |
| --- |
| *Nombre del caso de uso* |
| Modificar datos personales |
| *Descripción* |
| El usuario administrador iniciado en sesión modifica sus datos personales y los del paciente que tenga a cargo entre estos datos se incluyen el nombre, apellido, email y cambio de contraseña. |

|  |
| --- |
| *Nombre del caso de uso* |
| Programar horarios de medicación |
| *Descripción* |
| El usuario administrador iniciado en sesión selecciona el pastillero del dispensador que está programado, la fecha y la hora a la que se dispensará las pastillas. Opcionalmente podrá programar más de un horario a la vez añadiendo la hora que le corresponda por días. |

|  |
| --- |
| *Nombre del caso de uso* |
| Configurar notificaciones |
| *Descripción* |
| El usuario administrador iniciado en sesión le llegaran las notificaciones de alarmas del dispensador. Las notificaciones que se realizan al usuario son:   * Especifica la hora que debe seguir el tratamiento. En caso de la activación el usuario puede ver el detalle de la pastilla. |

***Caso de uso del actor Dispensador***

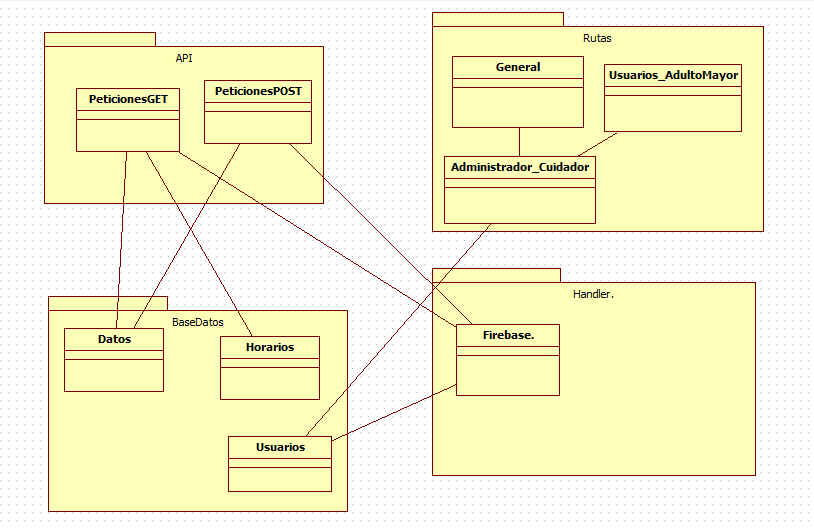


|  |
| --- |
| *Nombre del caso de uso* |
| Expulsar pastillas |
| *Descripción* |
| El dispensador expulsa las pastillas a la fecha y hora correspondientes a un horario programado por el usuario administrador. |

|  |
| --- |
| *Nombre del caso de uso* |
| Estado de conexión |
| *Descripción* |
| El dispensador realiza cada minuto peticiones al sistema y este controla el estado de conexión del dispensador. |

|  |
| --- |
| *Nombre del caso de uso* |
| Gestionar toma de medicación |
| *Descripción* |
| El dispensador gestiona los tiempos de toma de medicaciones programadas para expulsar el medicamento. |

|  |
| --- |
| *Nombre del caso de uso* |
| Expulsar pastillas por reconocimiento facial |
| *Descripción* |
| El dispensador expulsa las pastillas al identificar al usuario que corresponde ese dispensador. |

1. **Generación y adaptación de modelos**

1. **Generación de pruebas**

Mediante las pruebas realizadas con el dispensador medico se construyó los resultados esperados y resultados obtenidos*.*

|  |  |
| --- | --- |
| *Requisito funcional – Expulsar pastillas* | |
| *Prueba* | ***Resultado Esperado*** |
| *Dispensar pastillas* | *El dispensador expulsa pastillas y el sistema almacena el nuevo estado de los pastilleros tras expulsarlas* |
| ***Resultado obtenido*** |
| *El dispensador expulsa las pastillas de un horario efectivamente.* |
| *Requisito funcional – Registrar estado de dispensador* | |
| *Prueba* | ***Resultado Esperado*** |
| *Registrar nuevo estado de los pastilleros del dispensador* | *El sistema almacena el nuevo estado de los pastilleros del dispensador una vez a expulsado las pastillas de un horario de programación.* |
| ***Resultado obtenido*** |
| *El sistema almacena el nuevo estado de los pastilleros del dispensador correctamente* |
| *Requisito funcional – Expulsar pastillas por reconocimiento facial* | |
| *Prueba* | ***Resultado Esperado*** |
| *Dispensar pastillas al identificar que es el usuario que corresponde* | *El dispensador expulsa pastillas una vez que reconoce que es el paciente o el cuidador y el sistema almacena la identificación de la persona.* |
| ***Resultado obtenido*** |
| *El dispensador expulsa las pastillas tras reconocer al usuario que le corresponde ese dispensador y valida que sea el horario al que corresponde.* |

1. **Generación de software**

|  |  |
| --- | --- |
| *Requisito funcional – Registro de usuario en el sistema* | |
| *Prueba* | ***Resultado Esperado*** |
| *Registrar usuario* | *El sistema posee un usuario, registrar a un administrador y este administrador registra a sus pacientes* |
| ***Resultado obtenido*** |
| *El sistema efectivamente posee los usuarios que han sido registrados* |
| *Requisito funcional – Modificar datos personales* | |
| *Prueba* | ***Resultado Esperado*** |
| *Modificar daros personales del usuario* | *El sistema actualiza los datos personales del usuario* |
| ***Resultado obtenido*** |
| *El sistema modifica efectivamente los datos personales del usuario* |
| *Requisito funcional – Programar días y horas a las que debe expulsar las pastillas* | |
| *Prueba* | ***Resultado Esperado*** |
| *Programar día y hora de toma de medicación* | *El sistema registro un nuevo horario de programación y se muestra el usuario* |
| ***Resultado obtenido*** |
| *El sistema registra efectivamente el nuevo horario y aparece el horario programado.* |
| *Requisito funcional – Consultar horarios programados* | |
| *Prueba* | ***Resultado Esperado*** |
| *Consultar próximos horarios programados* | *El usuario consulta a través de la aplicación la fecha, hora y el tipo de medicina con el que debe medicarse, esto en el horario* |
| ***Resultado obtenido*** |
| *El usuario visualiza efectivamente los detalles de los próximos horarios que están programados.* |
| *Requisito funcional – Eliminar horarios programados* | |
| *Prueba* | ***Resultado Esperado*** |
| *Eliminar horarios programados* | *El usuario administrador a través del sistema elimina el horario programado y actualiza la pantalla con nuevos registros* |
| ***Resultado obtenido*** |
| *El usuario administrador efectivamente elimina y desaparece de los próximos horarios.* |
| *Requisito funcional – configurar notificaciones de alarmas* | |
| *Prueba* | ***Resultado Esperado*** |
| *Configurar las notificaciones para emitir las alertas* | *El sistema configura las notificaciones de la toma de medicinas para enviarlas a la hora que corresponde.* |
| ***Resultado obtenido*** |
| *El sistema envía las notificaciones en la hora exacta que le toca medicarse.* |
| *Requisito funcional – Gestionar horarios de toma de medicación* | |
| *Prueba* | ***Resultado Esperado*** |
| *Gestionar horarios de toma de medicación* | *El sistema controla el tiempo establecido para tomar la medicación dispensada* |
| ***Resultado obtenido*** |
| *El sistemas efectivamente controla el tiempo de toma para el medicamento dispensado* |

1. **Despliegue de hardware y software**

# Interfaces

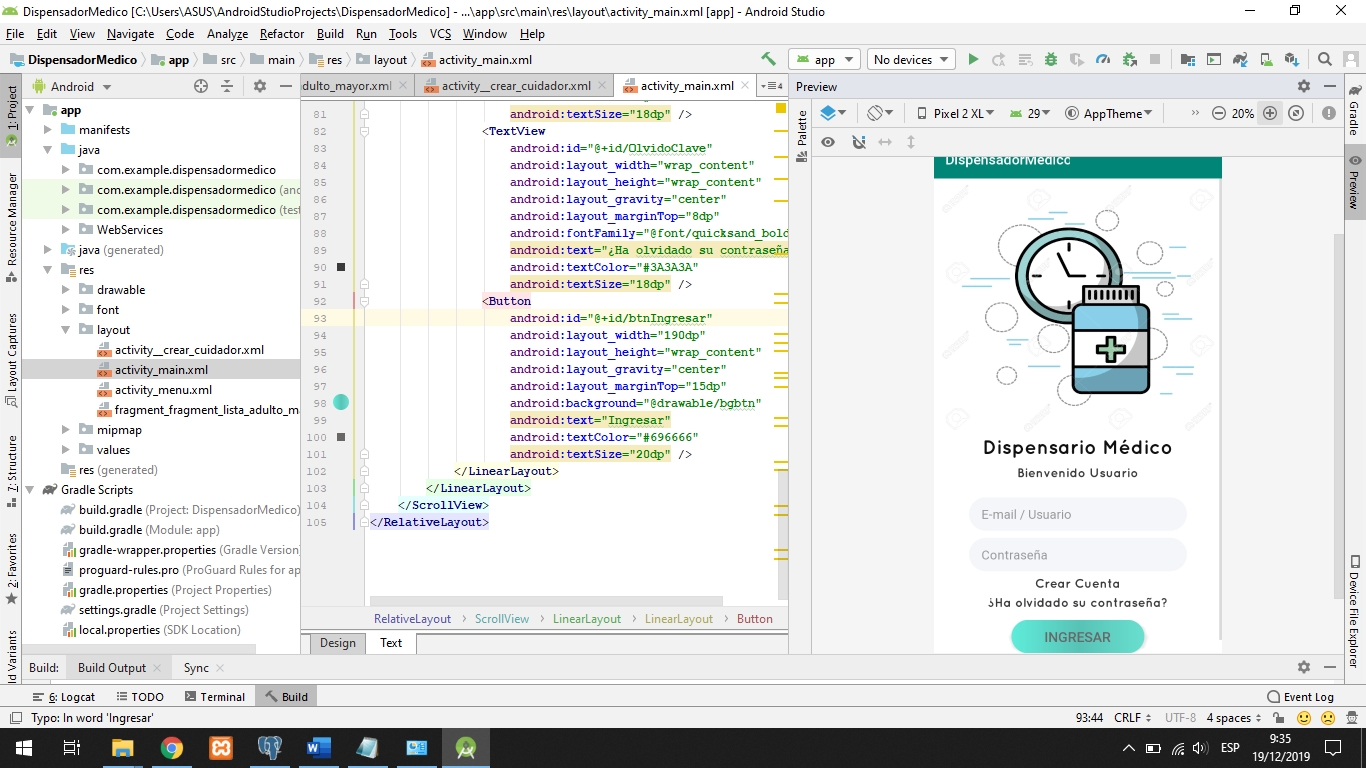
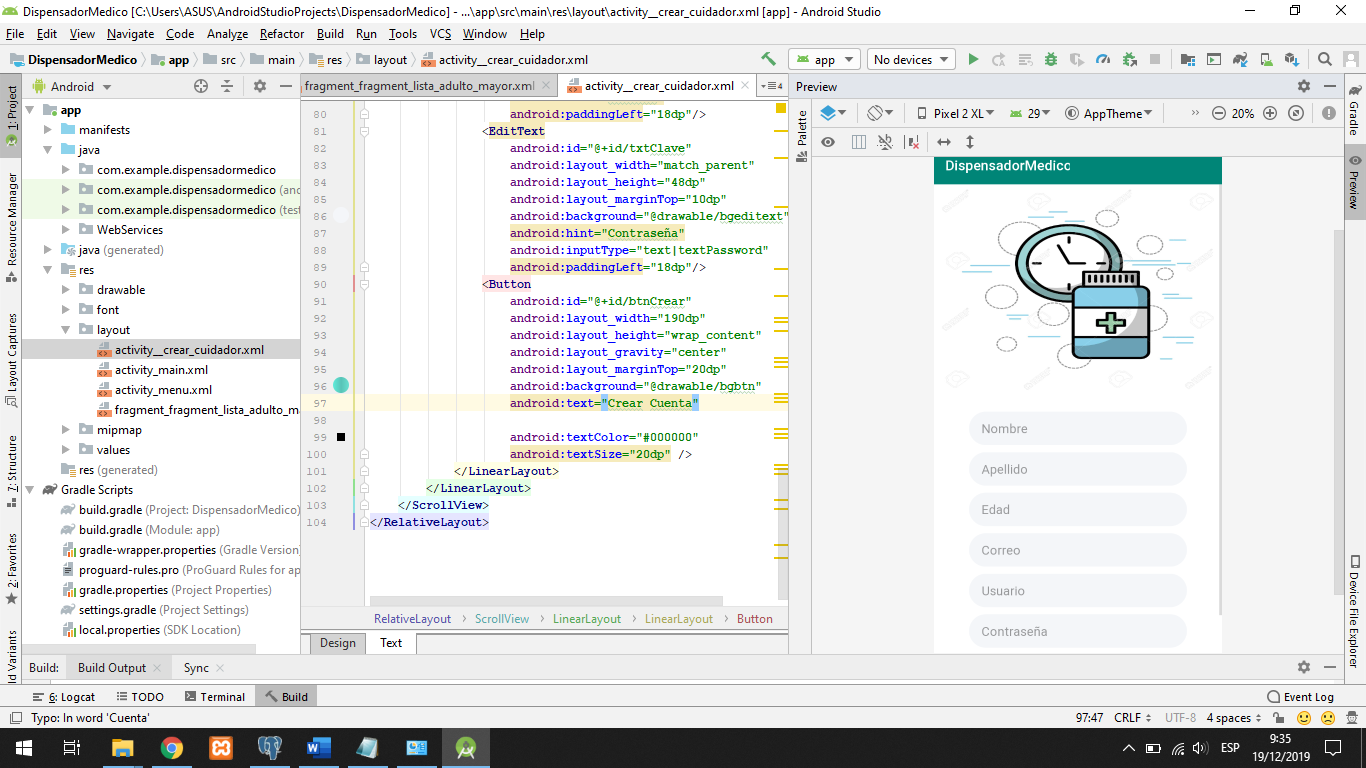


Figura 8 Inicio de sesión y registro de cuenta.

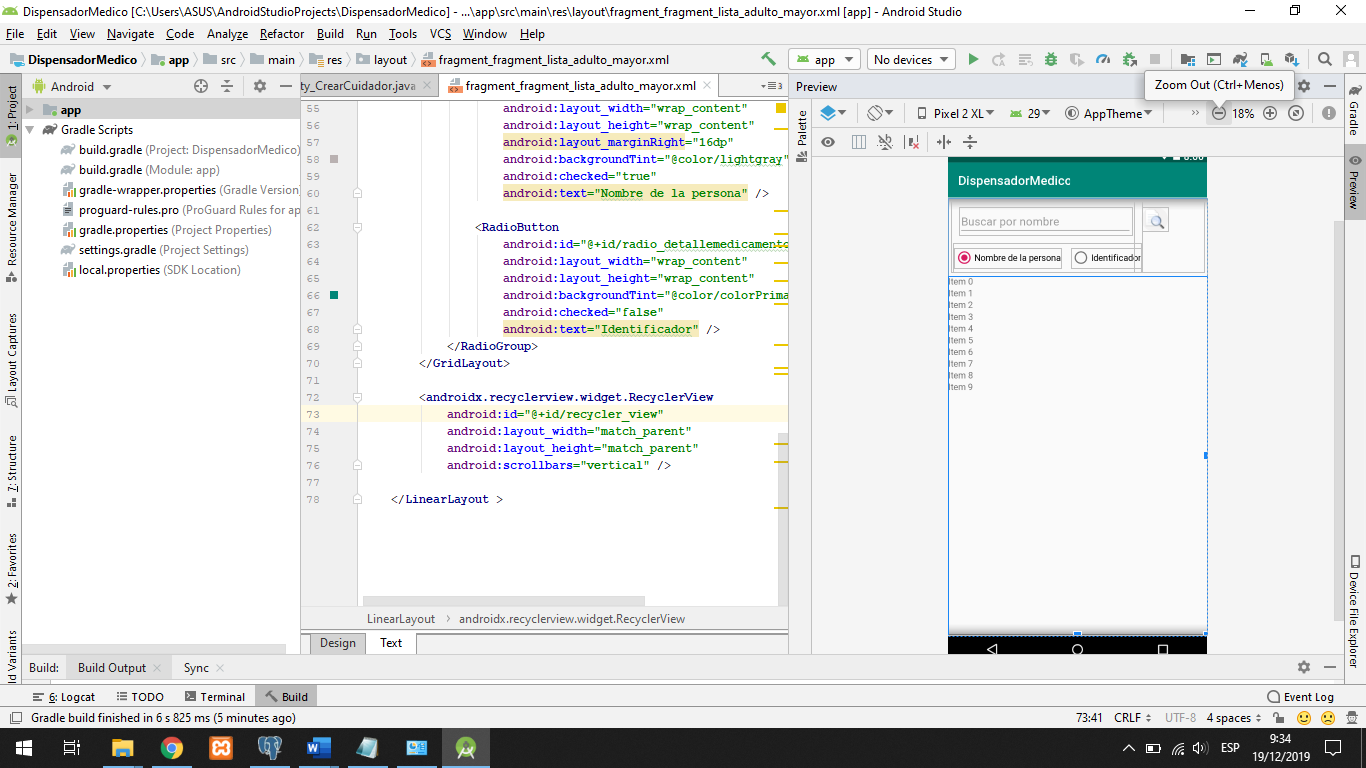
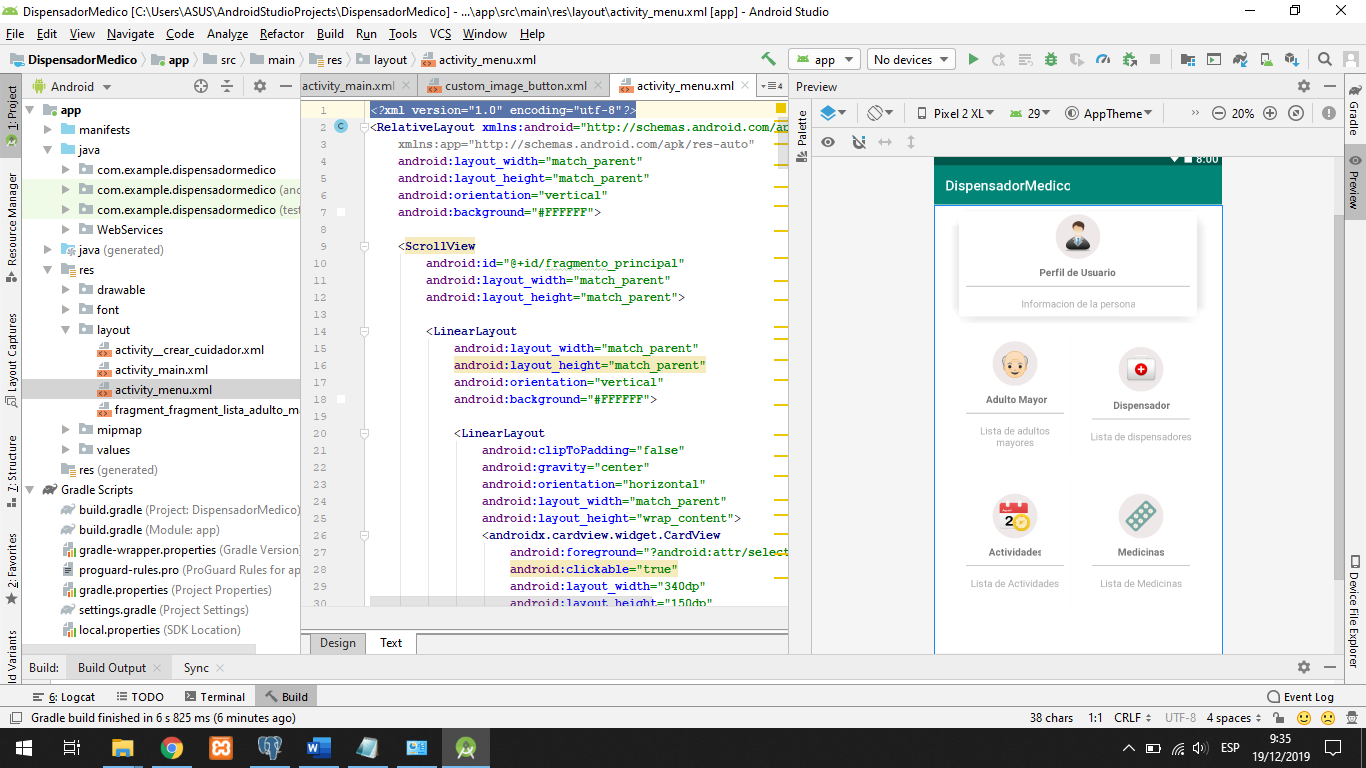


Figura 9 Visualizar la lista de adultos mayores, dispensadores, actividades y medicinas

Al tener una cuenta registrada en la aplicación se muestra el menú donde puede visualizar la lista de adultos mayores, dispensadores, actividades y medicinas que están bajo su tutela.

Añadir las demás pantallas y descripciones

# Desarrollo

**Web Services**

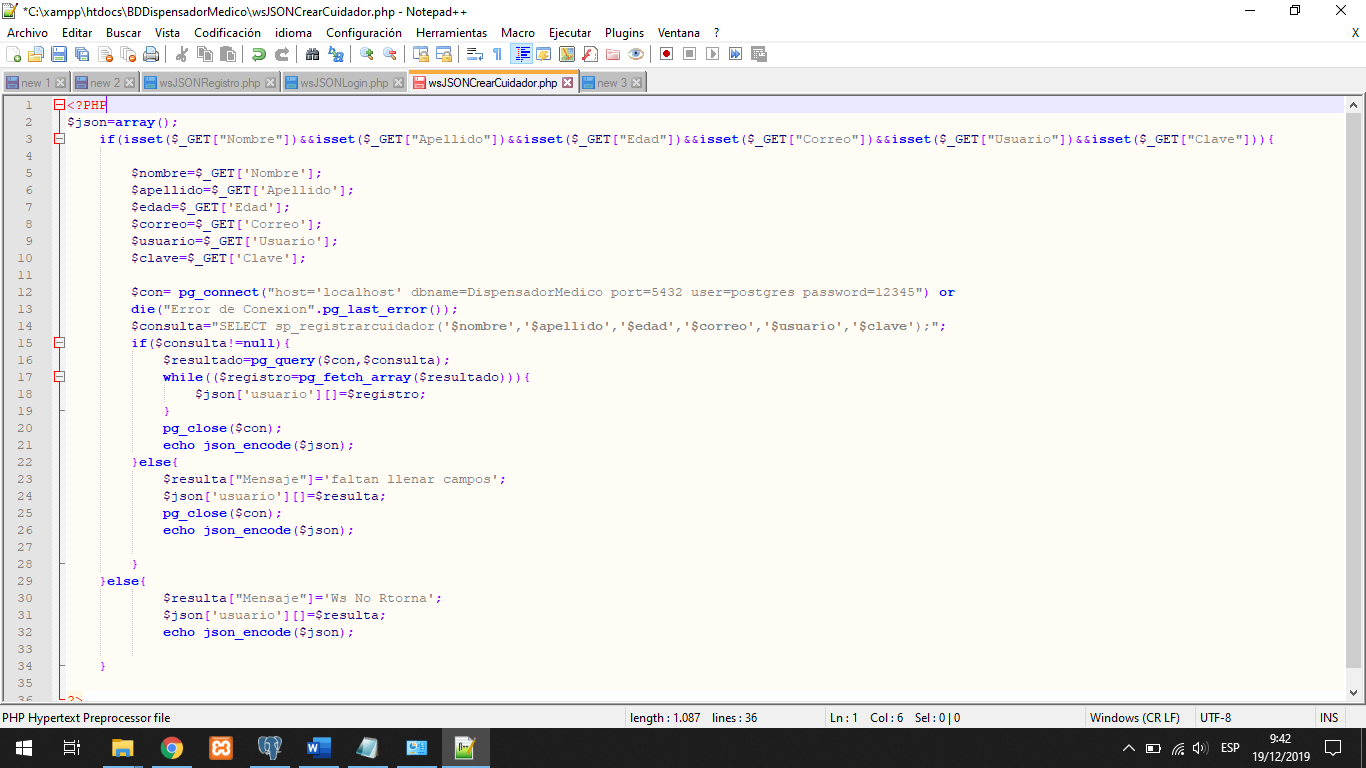


Figura 10 Codigo PHP para la creacion de los WebService

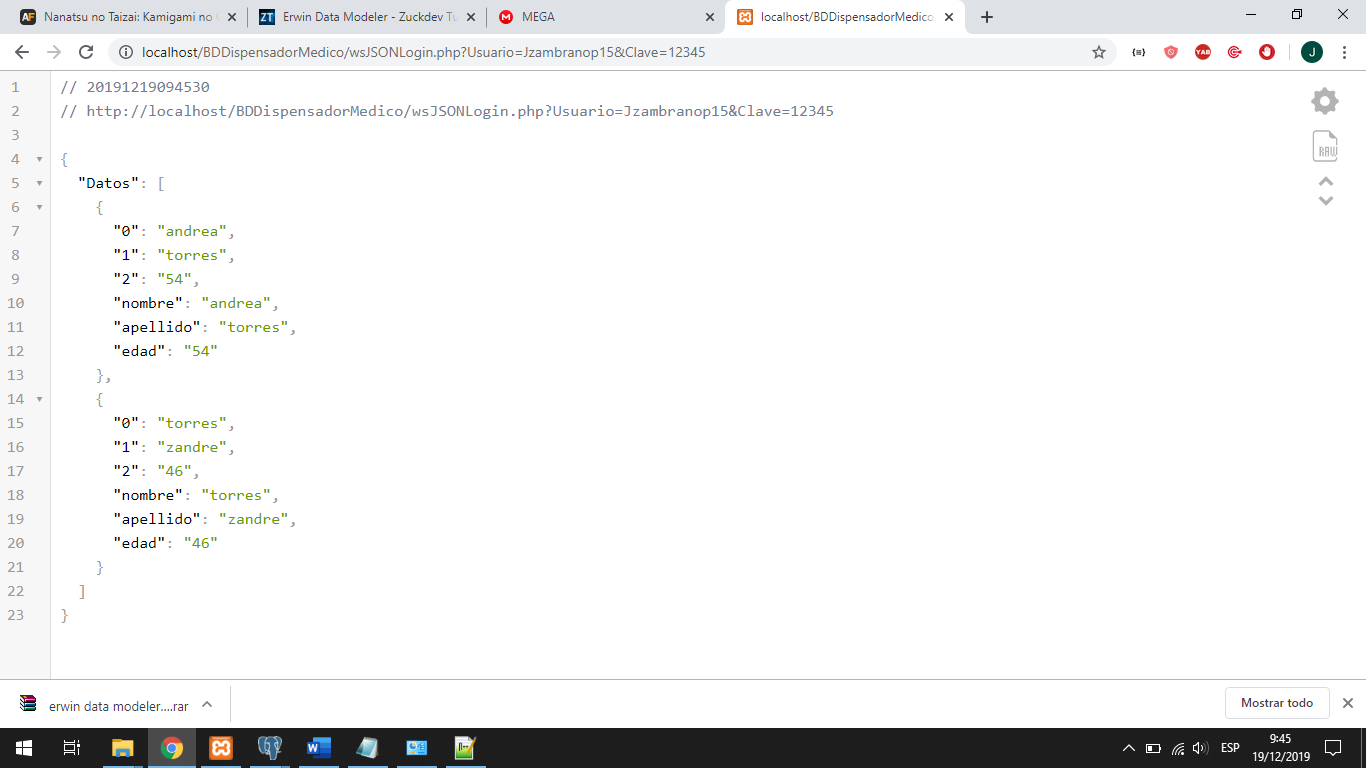
Para realizar el envio de datos a una dase de datos y su visualizacion en la aplicación movil se utiliza webServices y JSONs elaborados en php, es un medio que permite llevar a cabo la comunicación entre la app movil y gestor de base de datos postgresql.

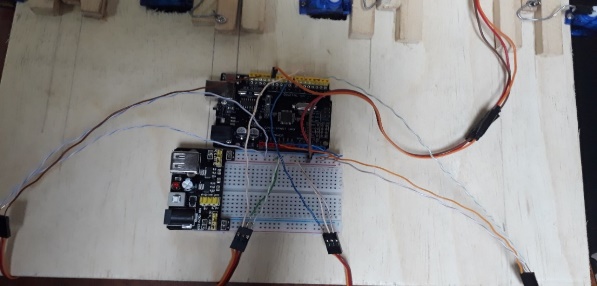
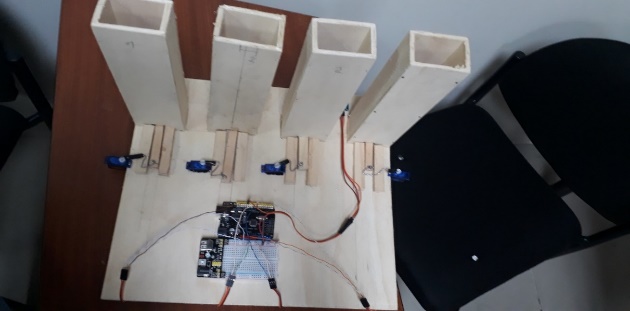
Figura 11 Json para el parseo de los datos

**Dispensador medico**

Dentro de este código se detalla en primera parte incluir la librería para trabajar con los servos motores, luego inicializamos una variable de la misma librería “Servo servo”. Además, establecemos el puerto por donde responderá nuestro infrarrojo.

En el setup se establece con cuantos baudios trabajaremos en este caso con 9600 y se le da un inicio a los servomotores para que inicien desde un punto.

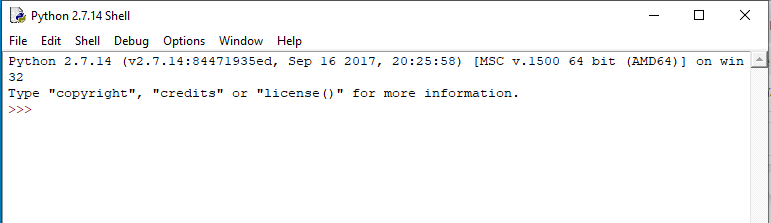
En el loop se indica todo el desempeño de los servomotores y los ángulos que tendrán para impulsar la pastilla de un recipiente. Se ha condicionado por medio de 1,2,3,4 se manipulará los dispensadores con el fin de que cada pastilla sea expulsada dependiendo de la acción que se tome, este caso input==’1’ o input==’2’, en cada condición se da el puerto de lectura del servomotor y los ángulos de entrada y salida. De esta forma se trabajará cada dispositivo, además se trabajó con un módulo bluetooth que permita la conexión entre Arduino y Android Studio.

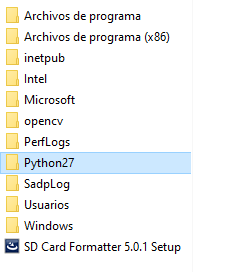


Cambiar

Se construyó 4 hileras en como recipientes donde contendrán cajitas con pastillas, la parte central de cada hila esta tiene un servomotor que actuará mediante una tablita adherida al mismo para expulsar el contenido. En el centro del tablero se encuentra la conexión del Arduino con los servos y la protoboad como parte de un punto para la distribución de corriente a los demás dispositivos.

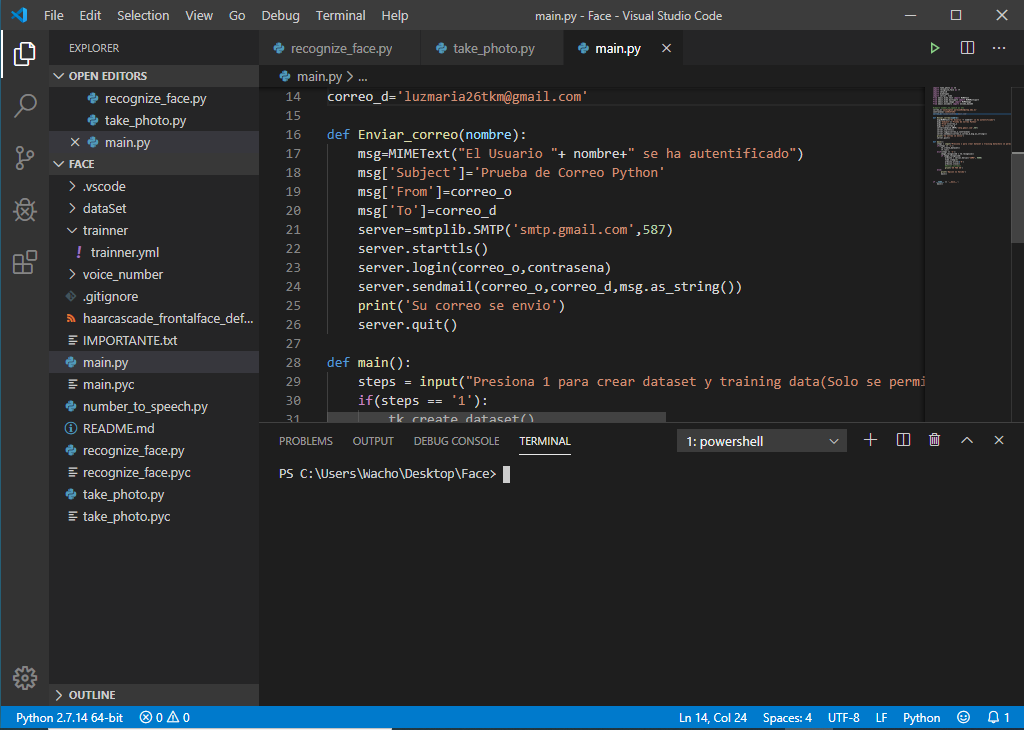
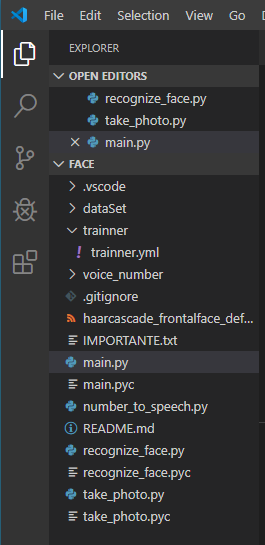
**Reconocimiento Facial**

* Primero se descarga el programa de Python de la página oficial <https://www.python.org/download/releases/2.7/>
* Luego descargamos Opencv para poder realizar el reconocimiento fácil de la página oficial <https://opencv.org/releases/>
* Comenzamos con la instalación de Python y Opencv, se puede seguir los pasos de este video. <https://www.youtube.com/watch?v=WSudP0RBu8Q>
* Una vez concluida la instalación podremos acceder al IDE de Python que se muestra en la siguiente imagen
* También tendremos una carpeta donde se encuentran los archivos Opencv y Python obtenidos durante la instalación y que los usaremos mas adelante.



* Luego descargamos el Proyecto Opencv que nos va a permitir realizar el reconocimiento entrenamiento y el envió del rostro. <https://github.com/Manokero/face-recognition-and-tts-numbers> (Este proyecto solo es una guía. Lo que nos sirve son la parte de crear el dataset con los cuadros del rostro y la parte del reconocimiento de caras)
* Luego con Powershell accedemos a la carpeta raíz donde se encuentra ubicado el proyecto y ejecutamos code.



* Nos aparece la ventana de Visual Studio Code con los archivos del proyecto
* Tendremos 3 archivos que vamos a utilizar RECOGNIZE\_FACE.PY, TAKE\_PHOTO.PY, MAIN.PY también tendremos un archivo trainner.yml que es donde el programa va a realizar el entrenamiento del rostro de la persona.
* Dentro del archivo MAIN.PY es donde se va a realizar todo el código principal del reconocimiento importamos las librerías necesarias

import take\_photo as tk

import recognize\_face as rf

import smtplib

import mimetypes

import serial, time

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

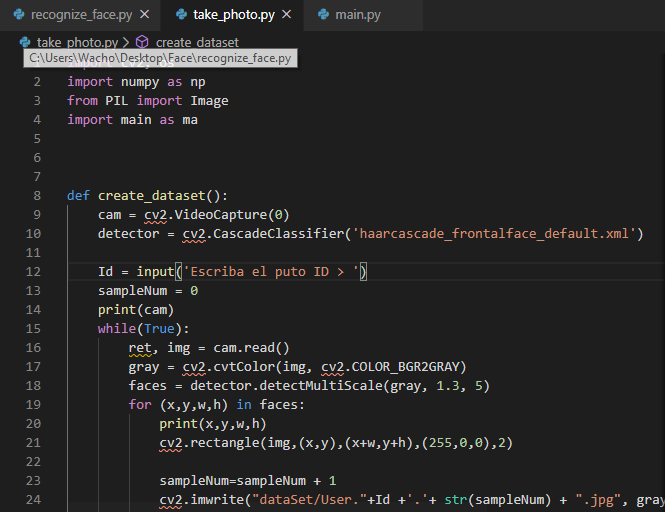
from email.mime.image import MIMEImage

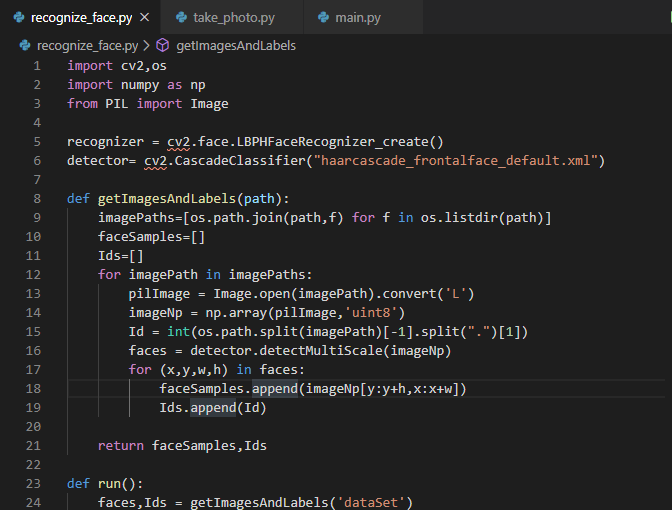
from email.encoders import encode\_base64

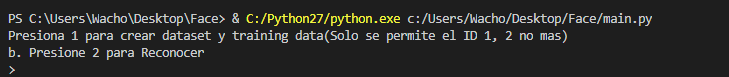
* Definimos 2 funciones que vamos a usar

def Enviar\_correo(nombre):

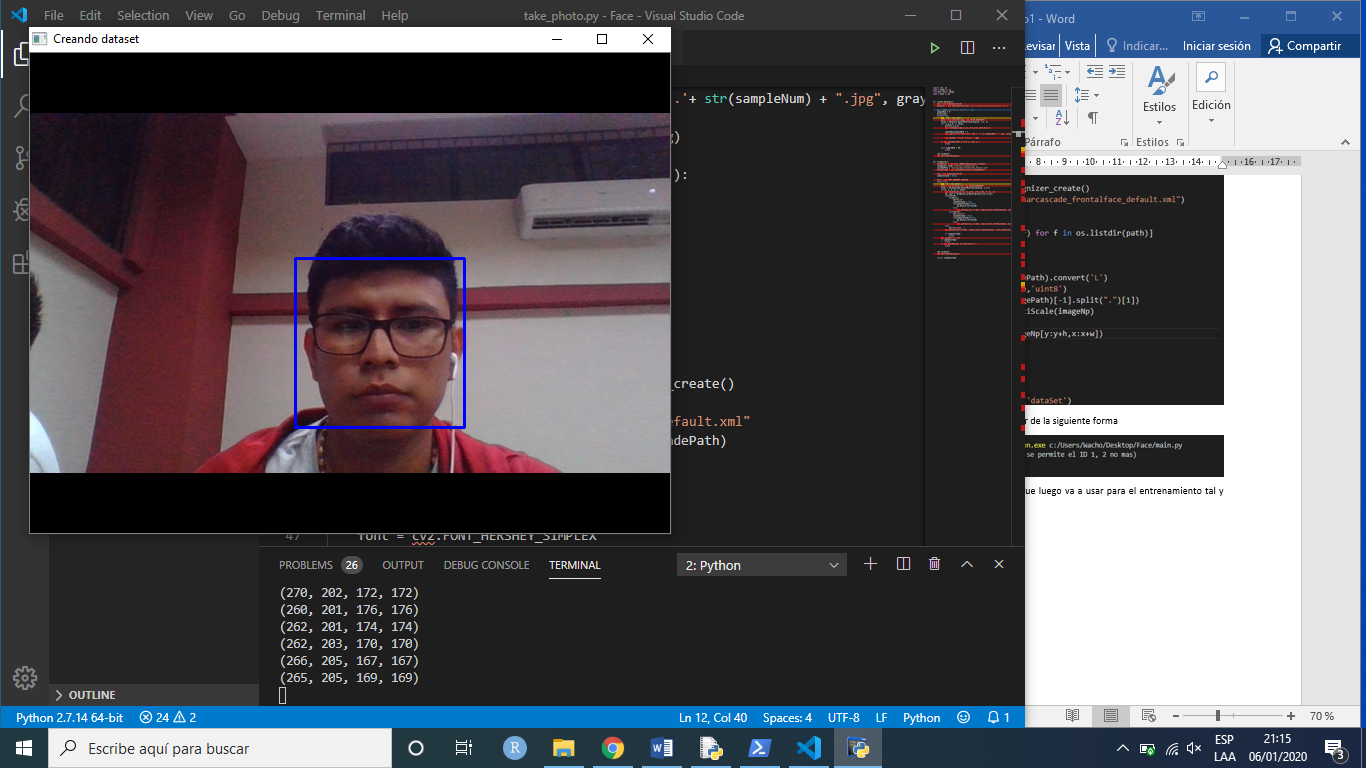
def main():

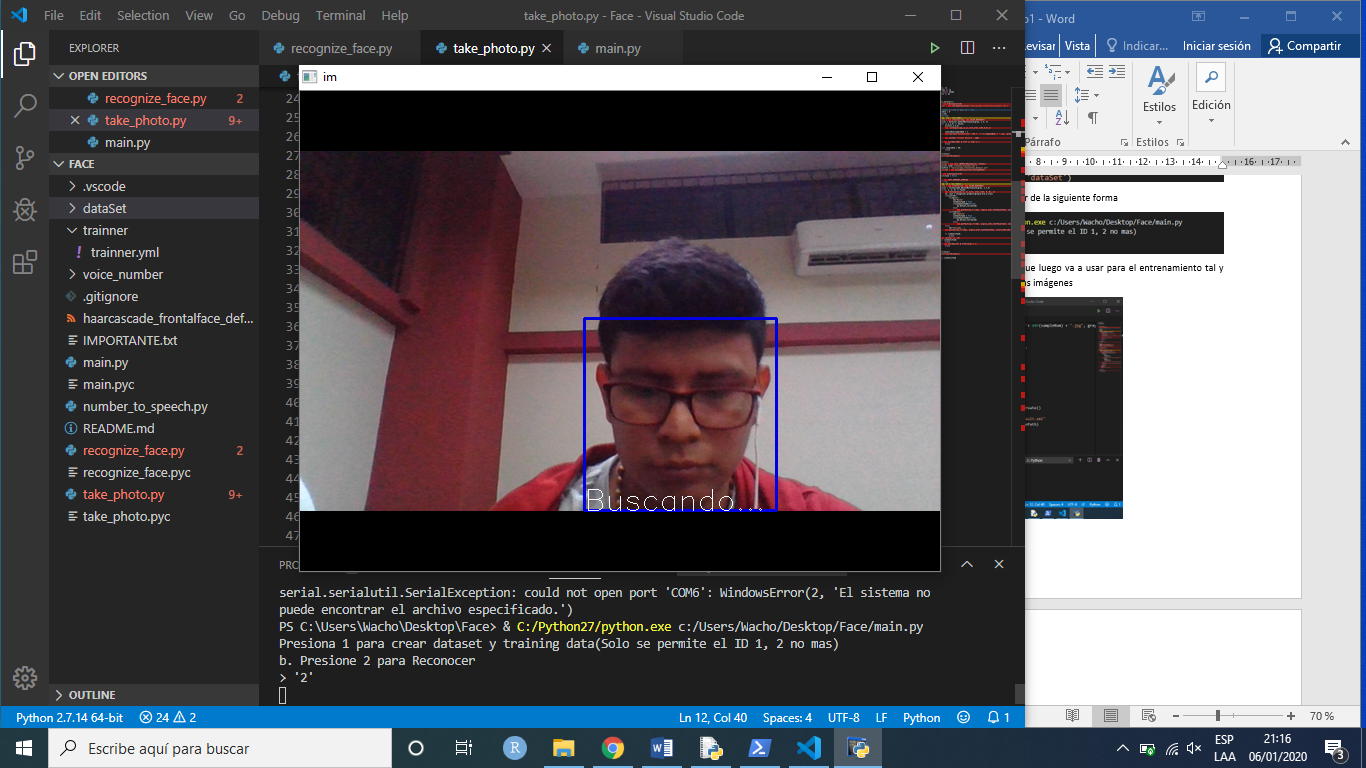
* enviar correo nos va a enviar un correo electrónico con el nombre de la persona que paso por el reconocimiento facial en ese momento
* main es el cuerpo de nuestro programa, las instrucciones se encuentran dentro de el
* luego tenemos el archivo TAKE\_PHOTO.PY que este se encarga de capturar las fotos para luego realizar el entrenamiento, guardando todos estos resultados en la carpeta dataset junto a la carpeta con el nombre de la respectiva persona.
* tenemos el archivo RECOGNIZE\_FACE.PY este archivo es el que se encarga de hacer las comparaciones entre la imagen que captura la cámara y las que previamente realizo el entrenamiento.

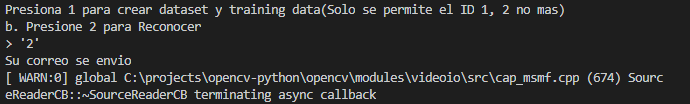


* cuando el programa se nos va a mostrar de la siguiente forma

La opción 1 nos va tomar una serie de fotos que luego va a usar para el entrenamiento tal y como aparece en la siguiente imagen, todas esas imágenes



* en la opción de nuestro main el código va a activar la cámara para iniciar el reconocimiento facial y comparar entre esa imagen y las que ya entreno previamente en el algoritmo, si el reconocimiento es TRUE y si se logro va a enviar un correo electrónico con la confirmación que el reconocimiento ha sido exitoso junto al nombre del usuario que paso por el reconocimiento seguido de esto enviara una señal al arduino para que pueda ejecutar sus tareas.
* opción 2 y el mensaje de que se envió correctamente el correo



1. **Evaluación de los resultados**

Finalmente, con la aplicación móvil y el dispensador de medicinas terminados y con todos los cambios destinados a mejorar la usabilidad de ambos componentes, se realizó una última batería de pruebas a cada uno de los perfiles de usuario analizados de la que se obtuvieron los siguientes resultados.

En la tabla presentada, se debe tener en consideración que los pasos o preguntas fueron evaluados por los propios autores de las pruebas, mientras qye las preguntas fueron respondidas por los usuarios de la aplicación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Pruebas de usabilidad sobre la aplicación móvil* | | | | | |
| *Paso o pregunta* | ***Perfil de usuario*** | ***Respuestas*** | | | |
| ***Siempre*** | ***Frecuente*** | ***Ocasional*** | ***Nunca*** |
| 1. *Registrarse en el sistema* | *Con habilidad en apps* | *\** |  |  |  |
| *Sin conocimientos informáticos* | *\** |  |  |  |
| *Con conocimientos informáticos* | *\** |  |  |  |
| 1. *Inicio sesión con usuario registrado* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿comprende lo que le muestra la pantalla de inicio?* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* |  | ***\**** |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *Programar un único horario de toma de medicamento* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿comprende lo que le muestra la pantalla de inicio?* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿Le ha resultado sencillo programar varios horarios para un paciente?* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿Navega en la pantalla para dispensar la medicina?* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿comprende cuál es el objetivo de modificar el tratamiento de un pastillero ?* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* |  | ***\**** |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *Configurar las notificaciones para que se envíen cuando tengan que tomar un medicamento* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* |  | ***\**** |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |
| 1. *Programar un horario de medicación* | *Con habilidad en apps* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| *Con conocimientos informáticos* | ***\**** |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Pruebas de usabilidad sobre el dispensador de medicinas* | | | | | |
| *Paso o pregunta* | ***Perfil de usuario*** | ***Respuestas*** | | | |
| ***Siempre*** | ***Frecuente*** | ***Ocasional*** | ***Nunca*** |
| 1. *Recoger pastillas dispensadas* | *Edad avanzada* | *\** |  |  |  |
| *Sin conocimientos informáticos* | *\** |  |  |  |
| 1. *¿El dispensador le ha dispensado la medicina en los horarios correctos?* | *Edad avanzada* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿El dispensador dispensa la medicina a través de la app móvil?* | *Edad avanzada* | ***\**** |  |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |
| 1. *¿Recuerda cómo dispensar las medicinas?* | *Edad avanzada* |  | ***\**** |  |  |
| *Sin conocimientos informáticos* | ***\**** |  |  |  |

De acuerdos a los resultados obtenidos han sido el dispensador bien estructurado con la dispensación de medicina de forma correcta y en los tiempos esperados, además las notificaciones dentro de la aplicación ayudan a recordar al cuidador como al adulto mayor. El diseño del dispensador es eficaz y manejable para su utilización.

Con respecto a los inconvenientes que se han presentado fueron por el reconocimiento facial al no contar con una cámara de mejor calidad y precisión, que permita tener un mejor enfoque y calidad para la foto.

1. ***Mantenimiento***

El mantenimiento de los equipos físicos de IoT como los servomotores, el modulo bluetooth deben ser supervisados por el tiempo de vida que tienen además de que son de poca potencia, susceptibles a daños, a sobrecargas de energía, por ello es necesario mantener el mantenimiento operativo de los servomotores para que no ocurra algún tipo de error.

La conectividad del dispensador con la aplicación móvil debe ser supervisada por el cuidador o el técnico correspondiente, la conectividad es dado por medio de bluetooth, este tipo de conexión es de poco alcance por ello es preciso mantener la distancia correcta, que está entre 10 metros.

# CONCLUSIONES

1. Se construyó el dispensador de medicina que permite dispensar las pastillas de forma unitaria o dosis dependiendo del horario que este establecido.
2. El diseño del dispensador fue realizado simultáneamente a la construcción del sistema electrónico, de todos los componentes, sensores y microcontrolador, programación del microcontrolador (Arduino) y el desarrollo de la aplicación móvil.
3. Se ha logrado conseguir no solo desarrollar todas las funcionalidades que habían sido fijadas, sino que durante el desarrollo del mismo se añadieron trabajos similares, además destacando el dispensador realizado con las nuevas funciones.
4. Dentro de la aplicación móvil se gestiona y programa todos los aspectos del paciente y del dispensador

# RECOMENDACIONES

1. Lo que se propone es realizar adaptaciones necesarias en el diseño del dispensador con el objeto de soportar varios números de pastillero y, por ende, el número de medicamentos diferentes que pueda dispensar.
2. Se recomienda realizar un análisis de cado uno de los módulos que componen el dispensador de medicinas antes de ampliar o modificar alguna funcionalidad por la gran importancia que cumplen cada uno dentro del desarrollo.
3. Para ampliar o modificar tanto la implementación del dispensador como el desarrollo del sistema distribuido, se recomienda consultar referencias bibliográficas que ayudarán al programador a entender cómo se han implementado ciertas funcionalidades del sistema
4. Mantener supervisado los horarios y el tratamiento de cada paciente a través de la aplicación móvil y el dispensador de medicina.

# REFERENCIAS

[1] J. Perez and C. Jesus, “INVESTIGACION DE PRE FACTIBILIDAD PARA LA FABRICACION DE DISPENSADOR DE COMIDA PARA MASCOTAS AUTOMATIZADO,” 2018.

[2] J. A. Cabañeros Blanco, “Dispensador Inteligente de Medicamentos Conectado y Sistema de Gestión,” p. 219, 2018.

[3] M. D. M. García and V. R. González, “La administración segura de medicamentos en los nuevos escenarios electromagnéticos de Internet de las Cosas ( IoT ),” 2018.

[4] MedMinder, “El dispensador de pastillas inteligente con conectividad inalámbrica - ITSitio.” [Online]. Available: https://www.itsitio.com/us/el-dispensador-de-pastillas-inteligente-con-conectividad-inalambrica/. [Accessed: 19-Dec-2019].

[5] J. F. Berry, S. A. Kelley, R. Mayer, S. Land, and M. G. Henckel, “Vacuum Operated Medicine Dispenser,” no. 19, 1995.

[6] B. Columbia and P. E. Price, “PILL DISPENSING SYSTEM.,” no. 19, 1987.

[7] G. Guerrero-Ulloa, M. Hornos, and C. Rodríguez-Dominguez, “TDDM4IoTS : A Test-Driven Development Methodology for Internet of Things ( IoT ) - based Systems,” in *1st International Conference on Applied Technologies ICAT 2019*, Comunications in Computer and Information Science, 2019.

# ANEXOS