

**UNIVERSIDAD MIGUEL HERNÁNDEZ DE
ELCHE**

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA MECÁNICA



**“DISEÑO Y CREACIÓN DE UNA APP DE AYUDA
PARA LA AUTOGESTIÓN DE LA MEDICACIÓN:
PASTILLERO DIGITAL”**

**TRABAJO DE FIN DE GRADO
MARZO 2021**

Autor: José Daniel Pérez García
Directora: María Asunción Vicente Ripoll

ÍNDICE

1.- Introducción	5
1.1.- Contexto y justificación del proyecto	5
1.2.- Objetivos	8
2.- Pastilleros	9
2.1.- Pastilleros físicos	9
2.1.1.- Qué es un pastillero	9
2.1.2.- Tipos de pastilleros	10
2.1.3.- Tipos de personas que utilizan pastilleros	14
2.2.- Pastilleros digitales en app	15
2.2.1.- Qué es un app pastillero	15
2.2.2.- Principales apps pastilleros en Android	16
3.- Diseño de la app para Android	21
3.1.- Selección de la información que necesita el pastillero y diseño de las pantallas de la app	21
4.- Programación	37
4.1.- Software Mit App Inventor	37
4.1.1.- Historia	38
4.1.2.- Características y funciones	39
4.1.3.- La herramienta	40
4.2.- Estructura de la app	42
4.2.1.- Screen1	43
4.2.2.- Screen_LoginUsuario	46
4.2.3.- Screen_ErrorLogin	49
4.2.4.- Screen_NuevoUsuario	51
4.4.5.- Screen_Pastillero	60
4.4.6.- Eliminar_medicamento	99
5.- Conclusiones	112
5.1.-Resultados	112
5.2.-Propuestas de mejora de la app	115
6.- Bibliografía	116

ÍNDICE ILUSTRACIONES

1.1.- Comparativa Android 1	6
1.1.- Comparativa Android 2	7
2.1.2.- Pastillero diario 1, 2, 3, 4 y 5	11
2.1.2.- Pastillero semanal 1, 2, 3 y 4	12
2.1.2.- Pastillero mensual 1, 2 y 3	13
2.1.2.- Pastillero electrónico 1, 2, 3 y 4	14
2.2.2.- App pastillero 1	16
2.2.2.- App pastillero 1, imagen 1, 2 y 3	18
2.2.2.- App pastillero 2	19
2.2.2.- App pastillero 2, imagen 1, 2 y 3	20
3.1.- Screen1, imagen 1	21
3.1.- Screen_NuevoUsuario, imagen 1	22
3.1.- Screen_NuevoUsuario, imagen 2 y 3	23
3.1.- Screen_NuevoUsuario, imagen 4	24
3.1.- Screen_NuevoUsuario, imagen 5	25
3.1.- Screen_Pastillero, imagen 1	26
3.1.- Screen_Pastillero, imagen 2 y 3	27
3.1.- Eliminar_medicamento, imagen 1	28
3.1.- Eliminar_medicamento, imagen 2	29
3.1.- Eliminar_medicamento, imagen 3	30
3.1.- Screen_NuevoUsuario, imagen 6	31
3.1.- Screen_Pastillero, imagen 4	32
3.1.- Screen_LoginUsuario, imagen 1	33
3.1.- Screen_ErrorLogin, imagen 1	34
3.1.- Screen_LoginUsuario, imagen 2	35
3.1.- Screen_Pastillero, imagen 5	36
4.1.- Pasos para creación de apps	42
4.1.- Screen1, imagen 2 y 3	44
4.1.- Screen1, diseño de bloques	45
4.1.- Screen_loginUsuario, imagen 3 y 4	46
4.1.- Screen_loginUsuario, diseño de bloques	47
4.1.- Screen_loginUsuario, subbloques 1 y 2	47
4.1.- Screen_loginUsuario, subbloques 3, 4 y 5	48
4.1.- Screen_loginUsuario, subbloques 6	49
4.1.- Screen_ErrorLogin, imagen 2 y 3	49
4.1.- Screen_ErrorLogin, diseño de bloques	50
4.1.- Screen_ErrorLogin, subbloques 1	50
4.1.- Screen_NuevoUsuario, imagen 7	51
4.1.- Screen_NuevoUsuario, imagen 8	52
4.1.- Screen_NuevoUsuario, diseño de bloques, imagen 1	52
4.1.- Screen_NuevoUsuario, diseño de bloques, imagen 2, 3 y 4	53
4.1.- Screen_NuevoUsuario, subbloques 1, 2, 3, 4 y 5	54
4.1.- Screen_NuevoUsuario, subbloques 6	55
4.1.- Screen_NuevoUsuario, subbloques 7 y 8	56
4.1.- Screen_NuevoUsuario, subbloques 9 y 10	57
4.1.- Screen_NuevoUsuario, subbloques 11, 12 y 13	58
4.1.- Screen_NuevoUsuario, subbloques 14 y 15	59
4.1.- Screen_NuevoUsuario, subbloques 16	60
4.1.- Screen_Pastillero, imagen 6 y 7	60
4.1.- Screen_Pastillero, diseño de bloques, imagen 1	62

4.1.- Screen_Pastillero, diseño de bloques, imagen 2	63
4.1.- Screen_Pastillero, diseño de bloques, imagen 3	64
4.1.- Screen_Pastillero, diseño de bloques, imagen 4 y 5	66
4.1.- Screen_Pastillero, diseño de bloques, imagen 6, 7, 8, 9 y 10	67
4.1.- Screen_Pastillero, subbloques 1	68
4.1.- Screen_Pastillero, subbloques 2	69
4.1.- Screen_Pastillero, subbloques 3, 4 y 5	70
4.1.- Screen_Pastillero, subbloques 6 y 7	71
4.1.- Screen_Pastillero, subbloques 8 y 9	72
4.1.- Screen_Pastillero, subbloques 10, 11 y 12	73
4.1.- Screen_Pastillero, subbloques 13 y 14	74
4.1.- Screen_Pastillero, subbloques 15 y 16	75
4.1.- Screen_Pastillero, subbloques 17, 18 y 19	76
4.1.- Screen_Pastillero, subbloques 20,21 y 22	77
4.1.- Screen_Pastillero, subbloques 23 y 24	78
4.1.- Screen_Pastillero, subbloques 25 y 26	79
4.1.- Screen_Pastillero, subbloques 27	80
4.1.- Screen_Pastillero, subbloques 28, 29 y 30	81
4.1.- Screen_Pastillero, subbloques 31 y 32	82
4.1.- Screen_Pastillero, subbloques 33, 34 y 35	83
4.1.- Screen_Pastillero, subbloques 36, 37	84
4.1.- Screen_Pastillero, subbloques 38 y 39	85
4.1.- Screen_Pastillero, subbloques 40 y 41	86
4.1.- Screen_Pastillero, subbloques 42, 43, 44 y 45	87
4.1.- Screen_Pastillero, subbloques 46	88
4.1.- Screen_Pastillero, subbloques 47, 48 y 49	89
4.1.- Screen_Pastillero, subbloques 50 y 51	90
4.1.- Screen_Pastillero, subbloques 52, 53 y 54	91
4.1.- Screen_Pastillero, subbloques 55, 56, 57 y 58	92
4.1.- Screen_Pastillero, subbloques 59 y 60	93
4.1.- Screen_Pastillero, subbloques 61, 62, 63 y 64	94
4.1.- Screen_Pastillero, subbloques 65	95
4.1.- Screen_Pastillero, subbloques 66 y 67	96
4.1.- Screen_Pastillero, subbloques 68, 69 y 70	97
4.1.- Screen_Pastillero, subbloques 71, 72 y 73	98
4.1.- Eliminar_medicamento, imagen 4 y 5	99
4.1.- Eliminar_medicamento, diseño de bloques, imagen 1	100
4.1.- Eliminar_medicamento, diseño de bloques, imagen 2 y 3	101
4.1.- Eliminar_medicamento, diseño de bloques, imagen 4	102
4.1.- Eliminar_medicamento, subbloque 1 y 2	102
4.1.- Eliminar_medicamento, subbloque 3	103
4.1.- Eliminar_medicamento, subbloque 4, 5 y 6	104
4.1.- Eliminar_medicamento, subbloque 7 y 8	105
4.1.- Eliminar_medicamento, subbloque 9, 10 y 11	106
4.1.- Eliminar_medicamento, subbloque 12 y 13	107
4.1.- Eliminar_medicamento, subbloque 14 y 15	108
4.1.- Eliminar_medicamento, subbloque 16 y 17	109
4.1.- Eliminar_medicamento, subbloque 18, 19 y 20	110
4.1.- Eliminar_medicamento, subbloque 21 y 22	111
5.1.- Captura 1 y 2	113
5.1.- Captura 3, 4, 5 y 6	114

AGRADECIMIENTOS

Debo indicar, para comenzar, que me gustaría mostrar mi agradecimiento a mi tutora del Trabajo de Fin de Grado, María Asunción Vicente Ripoll, la cual me ha guiado y ayudado para la realización de este Trabajo tan importante para mí por su finalidad.

Asimismo, la Universidad Miguel Hernández ha sido un campus que considero muy profesional, y no sólo en el sentido de profesorado y administrativo, sino en la llamativa cercanía en el trato y disposición a la ayuda que ha surgido durante mi estancia.

Y algo de gran importancia para mí es agradecer a mi novia Alexandra y mis padres Guillermina y José María esa fe ciega y apoyo incondicional que jamás me ha faltado de ellos. También a mis hermanos Pedro y Lidia por su confianza y cariño.

1.- INTRODUCCIÓN

1.1.- Contexto y justificación del proyecto

El desarrollo de este escrito se ha llevado a cabo con la finalidad de presentar el Trabajo de Fin de Grado y con ello finalizar el Grado en Ingeniería Mecánica en la Universidad Miguel Hernández de Elche.

El desarrollo de la aplicación *PillBox 1.0* se ha realizado bajo el Sistema Operativo Android. En concreto, se ha utilizado el lenguaje de programación App Inventor 2 (AI2), el cual es la versión mejorada de una herramienta de programación creada por el MIT (Instituto Tecnológico de Massachusetts) y que fue adoptada por Google para sus usuarios como solución para crear de una forma sencilla aplicaciones para dispositivos Android.

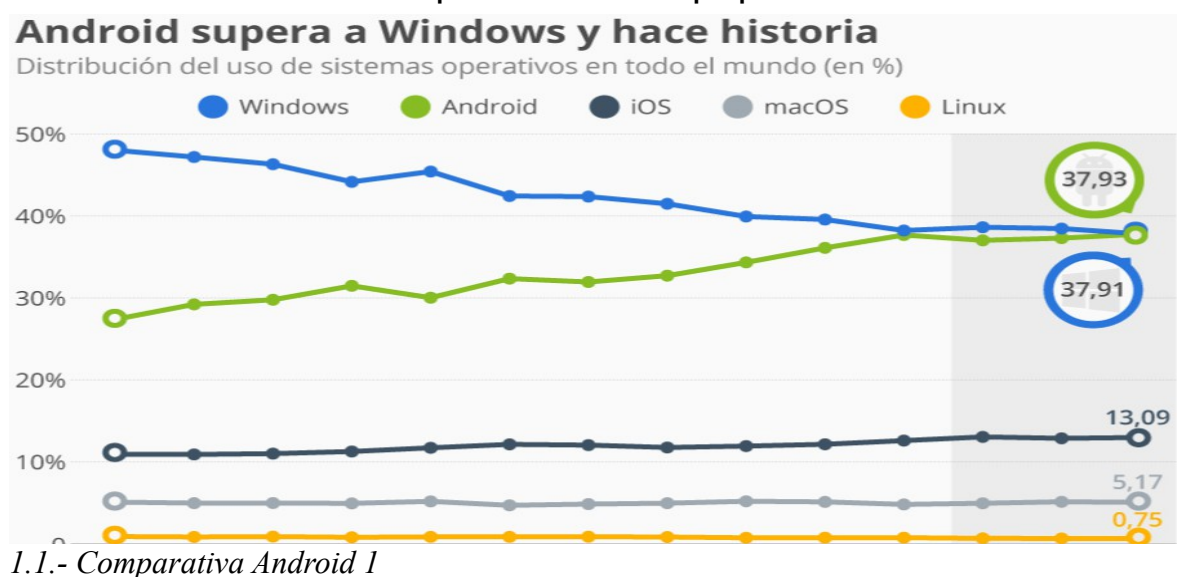
La motivación que se ha tenido para trabajar con este lenguaje de programación es doble: por una parte ha parecido muy interesante ampliar mis conocimientos de programación, lo cuales son humildes, y considero muy importantes en el mundo actual; por otra, me ha atraído mucho la atención la capacidad de App Inventor de crear aplicaciones de una inmensa variedad de una forma relativamente intuitiva y rápida con respecto a otros lenguajes.

PillBox 1.0 pretende servir de ayuda a toda persona que deba o quiera tomarse algún tipo de fármaco, algo que, desgraciadamente, necesita un número muy elevado de personas.

Muchas de ellas pueden tener distracciones o confusiones y por lo tanto errores en sus tomas, por lo que parecía interesante crear algo que pudiera servirles de apoyo. Esta app, quizás la primera de versiones mejoradas, se desarrolló con esa motivación de poder ser útil a dichas personas o a sus cuidadores.

Respecto al Sistema Operativo Android, es el más rentable a nivel mundial, superando a iOS de Apple es el sistema operativo por excelencia en smartphones y tablets. La firma utiliza un sistema basado en Kernel de Linux y otros software de código abierto, para un óptimo funcionamiento y desempeño.

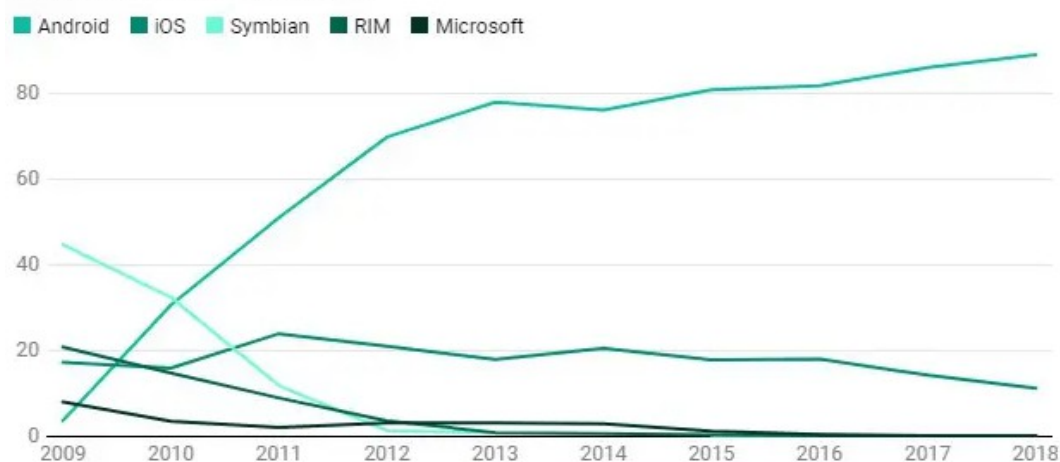
La revolución de los smartphones en los últimos años y la generalización de otros dispositivos móviles como las tablets ha contribuido a que el uso de sistemas operativos para dispositivos portátiles se dispare. Pese a la existencia del sistema iOS de Apple en iPhone, Android es el sistema operativo móvil más utilizado del mundo. Ya en 2017, el portal de estadísticas "Statista" informaba que 2017 sería probablemente el año en que Android releva a Windows como sistema operativo más popular del mundo.



En mayo de 2019, el periódico “La Tercera” informaba que ya se había llegado a tener un alcance de mercado de casi el 90%. Se puede observar en la siguiente imagen el espectacular éxito que ha ido teniendo con los últimos años Android frente a otros sistemas operativo.

Evolución de la cuota de mercado de Android (2009-2018)

Cifras expresadas en porcentajes



1.1.- Comparativa Android 2

Según la última investigación de Strategy Analytics, las ventas globales de smartphones alcanzaron un total de 275 millones de unidades en el primer trimestre de 2020. El segmento Android acaparó una participación dominante del 86% de todos los smartphones vendidos en todo el mundo en el trimestre.

Se ha considerado por tanto, más que interesante, utilizar este sistema operativo debido a su gran popularidad con la consiguiente ventaja de ofrecer productos en un mercado tan amplio y tan en auge como el de Android.

1.2.- Objetivos

PillBox 1.0 se ha desarrollado con el fin de mostrar al usuario la información que necesita para poder administrarse los medicamentos correctos en el momento correcto. El usuario introduciría los datos relevantes de los fármacos prescritos por su facultativo en la app (nombre, año de nacimiento, nombres de los fármacos y esta le mostraría por pantalla una planificación de las tomas en cuestión: filas con los datos: fecha (hora, día, mes, año) y fármaco correspondiente.

2.- Pastilleros

2.1.- Pastilleros físicos

2.1.1.- Qué es un pastillero

Los pastilleros son recipientes que permiten guardar, organizar, y llevar a cualquier lugar, toda la medicación que se debe tomar en un día, una semana, o incluso un mes. Su principal función es ayudar al paciente, o a sus familiares o cuidador a que siga correctamente su tratamiento farmacológico, tomando cada medicamento en la cantidad y momento adecuado, lo que es fundamental para que resulte efectivo. Están diseñados en forma de caja, con diferentes compartimentos para cada día, donde se guardan las diferentes pastillas o cápsulas, y los hay de mayor o menor tamaño, dependiendo de la cantidad de medicamentos que se tomen, y de la necesidad y comodidad del paciente.

Los pastilleros son muy útiles y necesarios para las personas mayores o polimedicadas, así como para quienes padecen demencia o alzhéimer y, por supuesto, para los más despistados de cualquier edad. Su uso evita equivocaciones en la toma, olvidos, e intercambios de medicamentos, errores que no sólo pueden reducir la eficacia del tratamiento, sino que aumentan las posibilidades de sufrir interacciones entre diferentes medicamentos (se estima que tomando cinco medicamentos a la vez el riesgo de interacción es del 50%), lo que puede causar complicaciones y problemas de salud.

Y es que una de las principales complicaciones que se encuentran hoy en día los médicos para ayudar a sus pacientes es la falta de adherencia al tratamiento, es decir, las personas no se toman la medicación como deben, o directamente dejan de tomársela, lo que repercute de forma negativa en su salud, pero también en los recursos sanitarios de la administración. Aunque dispongamos de una amplia variedad de tratamientos, y estos constituyan las principales herramientas para cuidar la salud de la población, solo son eficaces si se usan de forma adecuada; unas pautas mal seguidas pueden dar la vuelta a los efectos beneficiosos de los fármacos, provocando incluso efectos adversos.

Para evitar este problema, en las consultas médicas y en las oficinas de farmacia se recomienda el uso de pastilleros, especialmente a los adultos mayores que viven solos, aunque también son muy útiles para los familiares y cuidadores.

2.1.2.- Tipos de pastilleros

Existen distintos tipos, entre ellos:

- Diarios
- Semanales
- Mensuales
- Electrónicos

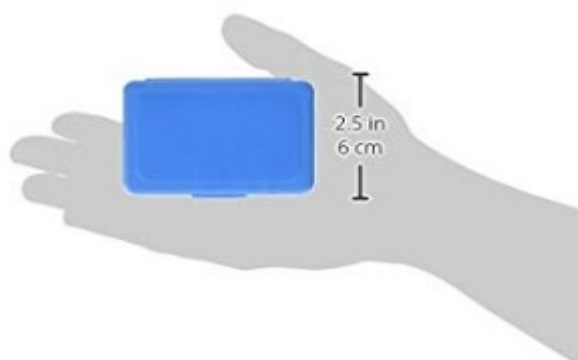
Los pastilleros diarios, como su nombre indica, son los diseñados para su reposición diaria y pueden disponer de varios compartimentos para separar las tomas, las cuales serán a prescripción médica. Es frecuente que los medicamentos se tomen con alimentos, pudiendo coincidir entonces las tomas con el desayuno, almuerzo, comida, merienda o cena. Algunos ejemplos de pastillero diario de uno y de varios compartimentos (siendo estos últimos los más habituales dentro de su tipo):



2.1.2.- Pastillero diario 1



2.1.2.- Pastillero diario 2



2.1.2.- Pastillero diario 3



2.1.2.- Pastillero diario 4



2.1.2.- Pastillero diario 5

Los pastilleros semanales están diseñados para dejar preparados todos los medicamentos a administrar durante toda la semana, siendo menos engorroso que los diarios, además de que elimina en cierto modo la duda de si 'se tomó las pastillas el día en cuestión'. Los hay también de una toma o varias, viniendo con iconos de los momentos del día o comidas o bien indicando lo mismo de forma escrita. Algunos ejemplos de la inmensa oferta que hay en el mercado:

De una toma diaria



2.1.2.- Pastillero semanal 1



2.1.2.- Pastillero semanal 2

O de varias, encontrando modelos muy elegantes y funcionales



2.1.2.- Pastillero semanal 3



2.1.2.- Pastillero semanal 4

Los pastilleros mensuales se encargarían de organizar todo el tratamiento del mes, y de la misma manera se pueden encontrar en el mercado tanto de una toma como de varias. Algunos ejemplos:



2.1.2.- Pastillero mensual 1



2.1.2.- Pastillero mensual 2



2.1.2.- Pastillero mensual 3

Finalmente, los pastilleros electrónicos se diferencian de los anteriores en que llevan alarmas para avisar de cuándo debe administrarse el fármaco, pudiendo ser una opción muy

interesante, existiendo asimismo modelos para una toma diaria así como para varias. Algunos ejemplos los tenemos en las siguientes imágenes.



2.1.2.-Pastillero electrónico 1



2.1.2.- Pastillero electrónico 2



2.1.2.- Pastillero electrónico 3



2.1.2.- Pastillero electrónico 4

2.1.3.- Tipo de personas que utilizarán un pastillero

Si bien cualquiera puede ser el usuario de un pastillero para poder llevar un correcto seguimiento de las tomas de los medicamentos recetados, hay que tener en cuenta los siguientes aspectos:

Si el usuario en cuestión padece algún tipo de enfermedad cognitiva, como demencia senil por ejemplo, ya no es éste el más indicado para llevar el seguimiento que, aunque facilitado, es sin duda insuficiente para este tipo de usuarios tan dependientes. En este caso debería ser, o bien un familiar o bien algún cuidador quien se encargara de la utilización del pastillero para ayudar a dicho usuario a llevar una correcta medicación.

Incluso si los usuarios tuvieran problemas auditivos sin revolver también sería conveniente una ayuda externa ya que no sería fiable que utilizaran alarmas para realizar correctamente sus tomas, dejando todo en manos de su atención, la cual sin duda acaba siendo no demasiado fiable.

Si añadimos a todo esto la cierta resistencia que ofrece un importante porcentaje de la población de alta edad a las nuevas tecnologías, la aplicación desarrollada en el presente proyecto como pastillero, no sería seguramente del agrado de dichos usuarios, relevando de nuevo a los familiares o cuidadores la tarea del correcto seguimiento de los fármacos recetados a las horas adecuadas.

2.2.- PASTILLEROS DIGITALES EN APP

2.2.1.- Qué es una app pastillero

Una app pastillero es una aplicación cuyo diseño y funcionalidad están destinadas a la información al usuario y

posibilidad de seguimiento por parte de éste de cuándo y cuáles de los medicamentos que le han sido prescritos por un facultativo debe administrarse. Si bien, esta información podría consultarla directamente el usuario, existen muchos tipos de apps pastillero, algunos de los cuales disponen también de alarmas y notificaciones para recordar al usuario qué medicamento debe tomar y cuándo, resultando así en una buena "herramienta aliado" para facilitar las tomas a los usuarios.

2.2.2.- Principales apps pastilleros en Android

Se van a mostrar un par de ejemplos de app pastilleros para Android, cuya información se ha extraído de google play.

Alarma de Medicina - Recordatorio de Medicamentos

Como primer ejemplo se va a mostrar la siguiente app, llamada *MyTherapy*, la cual tiene un puntual muy alta (4,8) otorgada por más de 96000 usuarios (a fecha de marzo de 2021).



Alarma de Medicina - Recordatorio de Medicamentos

MyTherapy   Medicina

★★★★★ 96.619 

 PEGI 3

 Esta aplicación está disponible para todos tus dispositivos

 Añadir a la lista de deseos

Instalar

2.2.2.- App pastillero 1

MyTherapy es un recordatorio gratuito de píldoras y medicación: Combinando un registro de pastillas, control de estados de ánimo y diario de salud, esta app de recordatorio de medicamentos para lograr un tratamiento exitoso.

Características principales

- Recordatorio de píldoras, pastillas, tabletas, para cualquier medicamento.
- Control de píldoras con diario de registro para tomas omitidas y confirmadas.
- Alarma de medicamentos con diferentes tipos de tomas o dosis.
- Seguimiento de tabletas, dosis, medidas, actividades.
- Medidas como peso, presión arterial, nivel de azúcar en sangre, etc, para todas las condiciones (diabetes, artritis reumatoide, ansiedad, depresión, hipertensión, esclerosis múltiple).
- Compartido en familia para su motivación y salud.
- Diario de salud imprimible para compartir con su médico.

Control de humor y estado de ánimo, peso, presión arterial

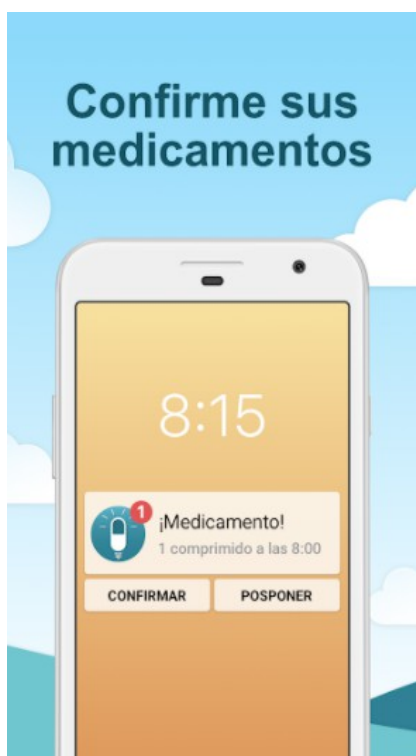
La app permite registrar los medicamentos del usuario, controlar su estado de ánimo y bienestar general en el diario de la aplicación o registrar mediciones como presión arterial y peso. Si el usuario vive con Diabetes y utiliza la insulina, MyTherapy puede ser su diario de diabetes y podrá controlar la glucosa en la sangre. MyTherapy soporta más de 50 mediciones. El registro de síntomas de la aplicación es importante para usuarios que sufren de

Esclerosis Múltiple, Artritis Reumatoide o enfermedades cardiovasculares como la Fibrilación Atrial.

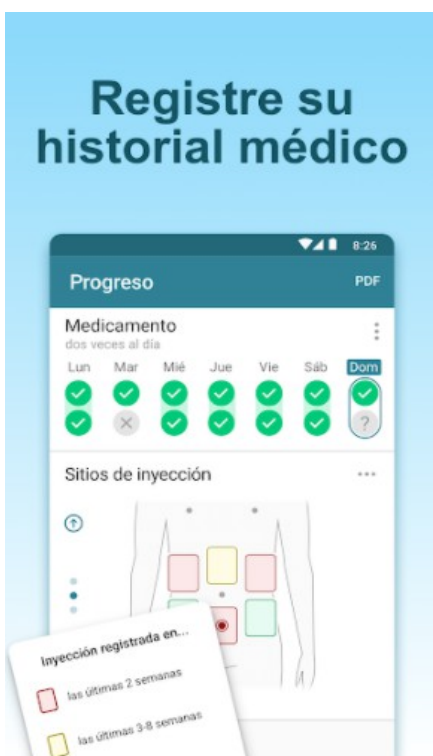
Motivación para el usuario de tomar su medicamento

El usuario recibe una foto al día como motivación para llevar su tratamiento. Involucre a su familia y amigos con Función Equipo y maneje su tratamiento acompañado. La familia del usuario puede ver si éste ha olvidado una dosis a pesar de la alarma de la píldora y ayudarlo a seguir el tratamiento.

Algunas capturas de pantalla de esta app son las siguientes:



2.2.2.-App pastillero, imagen 1



2.2.2.- App pastillero 1, imagen2



2.2.2.- App pastillero 1, imagen 3

Pastillas recordatorio

El segundo ejemplo de app, con una puntuación también

considerable (4,5) de unos 10000 usuarios (también a fecha de marzo de 2021), se mostrará información y capturas de pantalla de *Pastillas recordatorio*.



2.2.2.- App pastillero 2

Pastillas recordatorio recuerda al usuario cuándo es el momento de tomar sus medicamentos y lo ayuda a realizar un seguimiento de todas sus píldoras en el calendario. Recordatorio de medicamentos le dice que tome su píldora antes o después de la comida y le recuerda la dosis correcta de medicamento.

Recordatorio para tomar pastillas es una aplicación muy simple y fácil de usar que le permite agregar y administrar diferentes horarios de medicamentos. El usuario puede editar su programa de medicamentos, agregar nuevas píldoras u otros medicamentos, eliminar la receta existente o reemplazarla con más medicamento real en cualquier momento que desee. De esta forma, puede usar esta aplicación como horario de medicamentos, organizador de pastillas, alarma de medicación o como control de medicación.

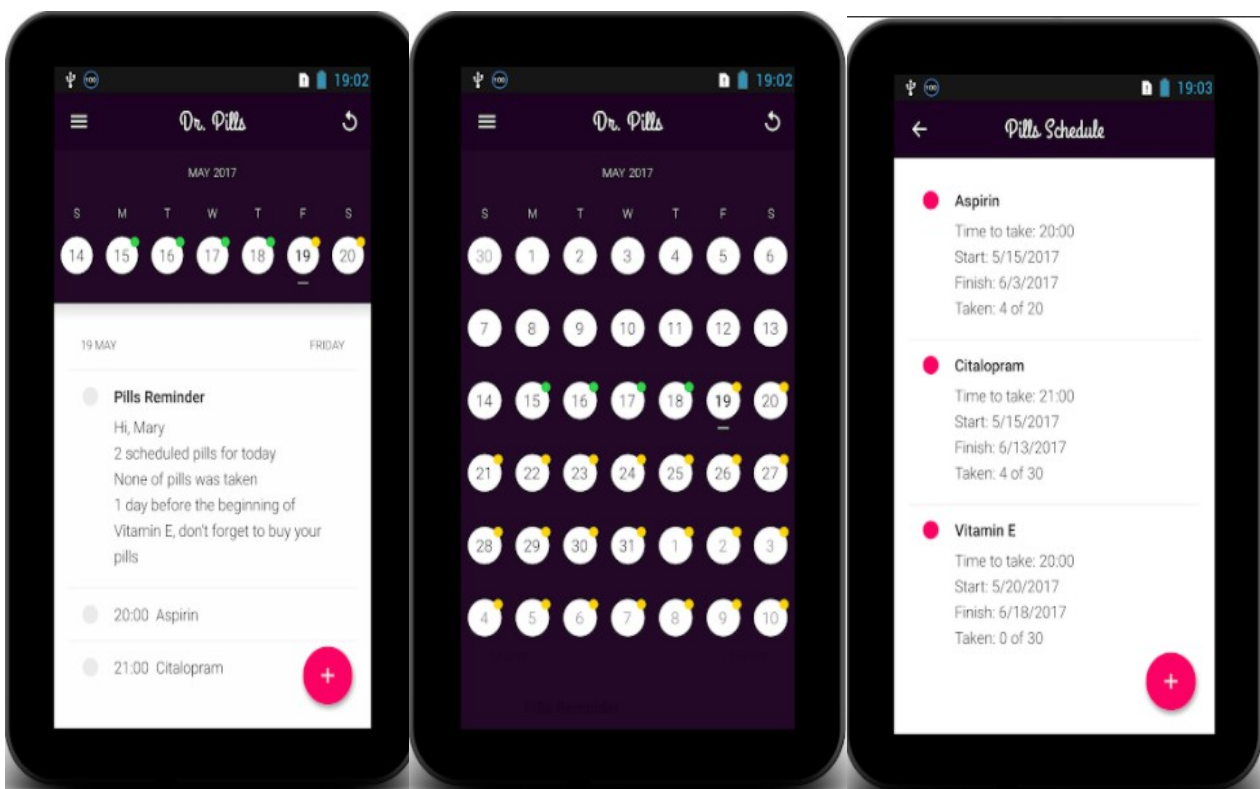
La aplicación ayuda al usuario a tener siempre el medicamento correcto en la dosis correcta en el momento adecuado. Todos los días, el usuario recibe la notificación a la hora programada para

tomar sus medicamentos. Además, el usuario tiene la posibilidad de que la aplicación le recuerde sobre la píldora hasta que la haya tomado. Para ello solamente debe activar esta y otras notificaciones importantes en su aplicación. Puede el usuario mantener un registro de todos sus medicamentos en una sola aplicación y administre su salud con la ayuda de las pastillas recordatorio.

Funciones disponibles en la versión gratuita:

- Horario de medicamentos.
- Organizador de pastillas.
- Alarma de medicación.
- Control de medicación.

Algunas capturas de pantalla de esta app son las siguientes:



2.2.2.- App pastillero 2, imagen 1

2.2.2.- App pastillero 2, imagen 2

2.2.2.- App pastillero 2, imagen 3

3.- DISEÑO DE LA APP PARA ANDROID

3.1.- Selección de la información que necesita el pastillero y diseño de las pantallas

Al iniciar la aplicación, ésta ofrece al dos opciones:

- 1.- "Iniciar sesión"
- 2.- "Crear usuario"



3.1.- Screen1, imagen 1

1.- Crear usuario

Al seleccionar esta opción, la app abrirá una ventana donde se solicitará la siguiente información:

- a).- Nombre.
 - b).- Año de nacimiento.
 - c).- Enfermedad del usuario.
 - d).- Fármaco prescrito para el tratamiento de la enfermedad.
 - e).- Frecuencia de la administración del fármaco en horas.
 - f).- Hora en la que empieza el tratamiento (en formato 24h).
- Ejemplo: 21 (si se desea indicar que la hora de inicio son las 9 P.M.
- g).- Duración del tratamiento (expresado en días)

3.1.- Screen_NuevoUsuario, imagen 1

Tras rellenar los siete datos anteriores, se deberá seleccionar el botón "Guardar", de lo contrario, cuando se intentara acceder a la planificación, saldría un mensaje informando que no se han

guardado los datos. Se puede seguir añadiendo tratamientos rellenando las siguientes casillas, y añadir nuevos seleccionando "Añadir tratamiento", tras haber guardado la información previamente seleccionando "Guardar tratamiento". La aplicación permite guardar hasta cinco tratamientos por usuario.

Se va a proceder a crear un usuario con datos ficticios de nombre Daniel López, para mostrar de manera más clara ciertas pantallas y funciones de la app, en especial la obtención de la planificación a seguir de las tomas de fármacos. Se utilizará para la demostración tres fármacos. Introduciendo dichos datos:

21:37 Screen_NuevoUsuario

NUEVO USUARIO

Nombre	Daniel López
Año de nacimiento	1960
Enfermedad	Cefalea
Fármaco	Aspirina 500 mg
Frecuencia (h)	12
Inicio (hora)	09
Duración tratamiento (d)	7

Guardar

Enfermedad	Inflamación del músculo
Fármaco	Ibuprofeno 600 mg
Frecuencia (h)	8
Inicio (h)	08
Duración tratamiento (d)	6

Guardar Añadir tratamiento

Ir a Planificación

3.1.- Screen_NuevoUsuario, imagen 2

21:40 Screen_NuevoUsuario

NUEVO USUARIO

Nombre	Daniel López
Año de nacimiento	1960
Enfermedad	Cefalea
Fármaco	Aspirina 500 mg
Frecuencia (h)	12
Inicio (hora)	09
Duración tratamiento (d)	7

Guardar

Enfermedad	Mareos
Fármaco	Biodramina 50 mg
Frecuencia (h)	24
Inicio (h)	12
Duración tratamiento (d)	5

Guardar Añadir tratamiento

Ir a Planificación

3.1.- Screen_NuevoUsuario, imagen 3

Para introducir el tercer tratamiento, solamente se ha tenido que pulsar el botón inferior "Guardar" y después "Añadir

tratamiento”, después de lo cual se limpia la ventana inferior correspondiente a todos los datos del segundo fármaco y se procede a introducir los nuevos.

En el caso de intentar guardar, tanto el primer fármaco como el segundo sin haber introducido todos los datos en las casillas correspondientes, aparecerá el siguiente mensaje de error:

23:54

Screen_NuevoUsuario

NUEVO USUARIO

Nombre Daniel López

Año de nacimiento 1960

Enfermedad Cefalea

Fármaco Aspirina 500 mg

Frecuencia (h) 12

Inicio (hora) 09

Duración tratamiento (d)

Por favor, introduzca todos los datos en las casillas correspondientes

Guardar

Enfermedad

Fármaco

Frecuencia (h)

Inicio (h)

Duración tratamiento (d)

Guardar Añadir tratamiento

Ir a Planificación

3.1.- Screen_NuevoUsuario, imagen 4

De la misma manera, si se pulsara el botón “Ir a Planificación” sin haber guardado antes los datos, al menos, relativos a la identificación del usuario y primer fármaco, aparecería otro

mensaje de error, esta vez en azul, indicando acerca de la necesidad de guardar los datos antes de poder cambiar a la ventana *Screen_Pastillero*.

23:55

Screen_NuevoUsuario

NUEVO USUARIO

Nombre: Daniel López

Año de nacimiento: 1960

Enfermedad: Cefalea

Fármaco: Aspirina 500 mg

Frecuencia (h): 12

Inicio (hora): 09

Duración tratamiento (d): 7

Por fafor, guarde antes los datos del nuevo usuario

Guardar

Enfermedad: Inflamación del músculo

Fármaco: Ibuprofeno 600 mg

Frecuencia (h): 8

Inicio (h): 08

Duración tratamiento (d): 6

Guardar

Añadir tratamiento

Ir a Planificación

3.1.- Screen_NuevoUsuario, imagen 5

Posteriormente, pulsando el botón “Ir a Planificación”, se abrirá una nueva ventana donde se leen los datos: Nombre del usuario, fecha de nacimiento, los fármacos que debe tomar el usuario y cinco botones, cuyas funciones se irán comentando posteriormente.



3.1.- Screen_Pastillero, imagen 1

Para obtener el “calendario” de las tomas de fármacos, se pulsará el botón “Planificación”, tras lo cual aparecerá, en la misma pantalla, una lista que detalla cuándo y qué fármaco debe ser tomar el usuario. El resultado, para el ejemplo que se está utilizando, es el siguiente:



3.1.- Screen_Pastillero, imagen 2



3.1.- Screen_Pastillero, imagen 3

Donde se ha optado por representar la pantalla completa antes y después de deslizarla hasta el final.

Otras de las opciones que ofrece esta pantalla son las de: "Eliminar medicamento" (por si fuera necesario por cualquier razón) y la de "Añadir" medicamento, representadas por sus respectivos botones.

Si, por la razón que sea, se deseara eliminar uno de los fármacos que tiene el usuario programados, se debería seleccionar el botón "Eliminar medicamento". Entonces la app abrirá una nueva ventana donde solicita el nombre del medicamento a eliminar.

También se podrá leer cierta información en dicha ventana, como son el nombre y año de nacimiento del usuario, así como una lista de los medicamentos que tiene programados el usuario.

Continuando con el ejemplo que nos ocupa, obtendríamos la siguiente imagen:



3.1.- *Eliminar_medimento, imagen 1*

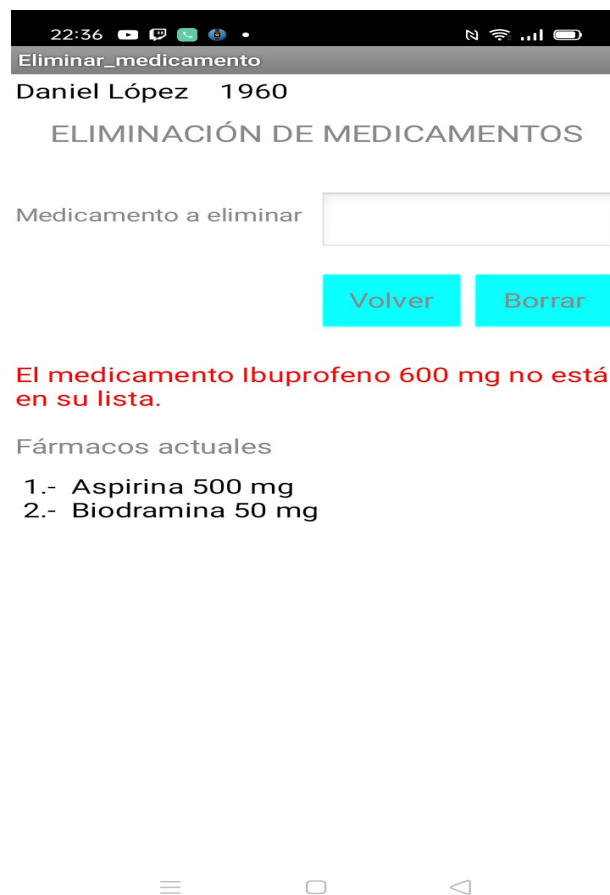
En el caso que se deseara eliminar, por ejemplo, el segundo fármaco (Ibuprofeno), tras escribirlo en la casilla correspondiente (es necesario escribir el nombre completo del medicamento, tal como aparece en la lista de medicamentos actuales, en este caso *Ibuprofeno 600 mg*) y pulsar el botón "Borrar", se borrará dicho fármaco, aparecerá un mensaje en la pantalla indicando que se ha eliminado correctamente y se actualizará la lista de los fármacos.

El resultado de lo anterior lleva a la siguiente imagen:

The screenshot shows a mobile application interface. At the top, the status bar displays the time 22:36 and various icons. Below the status bar, the title bar reads 'Eliminar_medicamento'. The user's name 'Daniel López' and birth year '1960' are displayed. The main heading is 'ELIMINACIÓN DE MEDICAMENTOS'. There is a text input field labeled 'Medicamento a eliminar' with an orange border. Below the input field are two cyan buttons: 'Volver' and 'Borrar'. A pink message states: 'El medicamento Ibuprofeno 600 mg ha sido borrado de su lista.' Below this, the section 'Fármacos actuales' lists two items: '1.- Aspirina 500 mg' and '2.- Biodramina 50 mg'.

3.1.- *Eliminar_medicamento, imagen 2*

Por otra parte, si se intentara borrar un fármaco que no aparece en la lista de "Fármacos actuales", aparecería un mensaje de error acompañado del nombre del fármaco que se ha intentado borrar. En este caso se ha introducido de nuevo el nombre del medicamento eliminado anteriormente (Ibuprofeno 600 mg), ofreciendo el siguiente resultado:



3.1.- *Eliminar_medamento, imagen 3*

Una vez eliminados los medicamentos que se desearan pulsaríamos el botón “Volver” para regresar a la pantalla *Screen_Pastillero*, donde se podría consultar la actualizada planificación programada de toma de medicamentos.

El siguiente de los botones a mencionar de esta ventana es el de “Añadir” medicamento. Al pulsar sobre éste se abrirá de nuevo la ventana *Screen_NuevoUsuario*, donde se deberá introducir los datos requeridos para guardar los tratamientos deseados (tener en cuenta que se deberá empezar rellenando las casillas por la subventana superior). Recuérdese que la aplicación permite trabajar con un máximo de cinco fármacos distintos.

Utilizando el botón “Añadir” para incluir otro medicamento en la planificación de las tomas de los mismos y rellenando las casillas para un fármaco, se obtendría (para el ejemplo que se está considerando) la siguiente imagen en la pantalla del dispositivo:

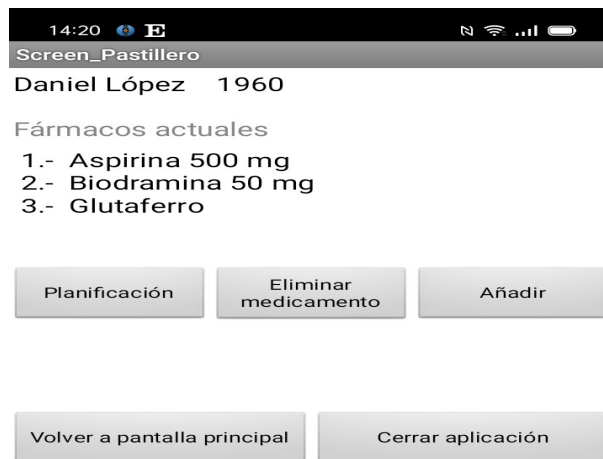
The screenshot shows a mobile application interface titled "Screen_NuevoUsuario". At the top, there is a status bar with the time "14:08" and various icons. Below the title bar, the text "NUEVO USUARIO" is centered. The form contains the following fields and values:

Field	Value
Nombre	Daniel López
Año de nacimiento	1960
Enfermedad	Anemia
Fármaco	Glutaferro
Frecuencia (h)	12
Inicio (hora)	10
Duración tratamiento (d)	14

Below the first set of fields, there is a green button labeled "Guardar". Underneath this, there is a second set of empty input fields for "Enfermedad", "Fármaco", "Frecuencia (h)", "Inicio (h)", and "Duración tratamiento (d)". At the bottom of the form, there are three green buttons: "Añadir tratamiento", "Ir a Planificación", and "Guardar". The bottom of the screen shows standard Android navigation icons.

3.1.- Screen_NuevoUsuario, imagen 6

Y, tras guardar los datos y pulsar el botón “Ir a Planificación”, se podrá observar que la aplicación guarda e incluye este nuevo fármaco en su base de datos, de manera que se podrá acceder a ellos en otro momento tras haber cerrado la aplicación. El resultado de este ejemplo sería la siguiente imagen:



3.1.- Screen_Pastillero, imagen 4

De nuevo, desde esta ventana se podrá obtener la planificación de las tomas de los fármacos, eliminarlos o añadir nuevos.

El botón “Volver a pantalla de usuarios” permite regresar a la pantalla principal, de manera que se pueda iniciar sesión con otro usuario si se deseara, bien por utilizar la app una unidad familiar o bien por utilizarla un cuidador que se encargara de administrar las tomas a los usuarios.

El último botón de la pantalla es “Cerrar aplicación”, cuyo nombre indica su función.

2.-Iniciar sesión

Cuando existe un usuario con una cuenta se deberá elegir esta opción. Una vez elegida, la aplicación abrirá otra pantalla, y en ésta se solicitará la siguiente información del usuario para poder acceder a los datos almacenados de éste:

a).- Nombre.

b).- Fecha de nacimiento.

3.1.- Screen_LoginUsuario, imagen 1

Una vez introducidos los datos, tanto si éstos han sido escritos incorrectamente como si el usuario no tuviera una cuenta, la aplicación abrirá otra pantalla de "error" donde informará que, o

bien los datos introducidos no coinciden con los del usuario, o dicho usuario no tiene creada una cuenta.

La pantalla de error en cuestión es la que aparece en la siguiente imagen, la cual permanecerá visible durante un tiempo de 1.5 segundos.



**ERROR: Los datos
de este usuario no
coinciden. Por favor,
revisa los datos
introducidos o crea un
nuevo usuario**

≡ □ ◀

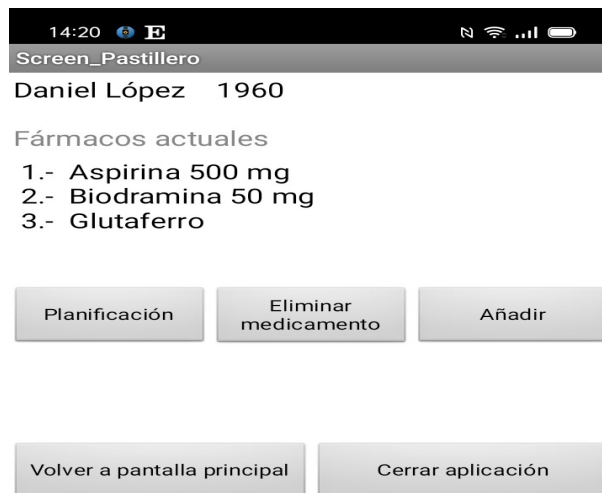
3.1.- Screen_ErrorLogin

Tras mostrarse la información de la imagen anterior durante 1.5 segundos, se abrirá de nuevo la primera ventana de la aplicación, donde se podrá intentar de nuevo acceder a los datos de un usuario ya creado anteriormente o crear una cuenta de uno nuevo. En este caso se va a utilizar, para continuar con el ejemplo con el que se ha venido trabajando, los datos del usuario creado en dicho apartado anterior para iniciar sesión con un usuario existente.



3.1.- Screen_LoginUsuario, imagen 2

Al reconocer la app la información del usuario solicitada para “Iniciar sesión”, ésta abrirá una nueva ventana, la cual corresponde a la siguiente imagen y donde el usuario podrá obtener la planificación ordenada por fecha y hora para sus tomas de medicamentos (entre otras opciones y como se ha comentado en el apartado anterior). Esta información se podrá obtener seleccionando el botón “planificación”. Asimismo aparece como información en la pantalla los datos del usuario y los fármacos que debe seguir tomando.



3.1.- Screen_Pastillero, imagen 5

Entrando de nuevo en la ventana que se mostró en el apartado “Crear usuario”, una vez creado este, y con la particularidad con respecto al caso anterior que la lista de fármacos es distinta ya que quedó guardada al haber procedido a eliminar uno de los fármacos (*Ibuprofeno 600 mg*) y añadir otro (*Glutaferro*) como se observó en dicho punto.

Las funcionalidades de la app a partir de este punto ya se han comentado en el apartado anterior **“Crear Usuario”**.

4.- PROGRAMACIÓN

4.1.- Software Mit App Inventor

MIT App Inventor es un entorno de programación visual intuitivo que permite a todos, incluso a los niños, crear aplicaciones totalmente funcionales para teléfonos inteligentes y tabletas. Aquellos que son nuevos en MIT App Inventor pueden tener una primera aplicación simple en funcionamiento en menos de 30 minutos. Y lo que es más, la herramienta basada en bloques facilita la creación de aplicaciones complejas y de alto impacto en mucho menos tiempo que los entornos de programación tradicionales. El proyecto MIT App Inventor busca democratizar el desarrollo de software al permitir que todas las personas, especialmente los jóvenes, pasen del consumo de tecnología a la creación de tecnología.

Un pequeño equipo de personal y estudiantes de CSAIL, dirigido por el profesor Hal Abelson, forma el núcleo de un movimiento internacional de inventores. Además de liderar la divulgación educativa en torno a MIT App Inventor y realizar investigaciones sobre sus impactos, este equipo central mantiene el entorno de desarrollo de aplicaciones en línea gratuito que sirve a más de 6 millones de usuarios registrados.

Los programas de codificación basados en bloques inspiran el empoderamiento intelectual y creativo. MIT App Inventor va más

allá de esto para proporcionar un poder real para que los niños hagan una diferencia, una forma de lograr un impacto social de valor incalculable para sus comunidades. De hecho, los App Inventors en la escuela y fuera de los entornos educativos tradicionales se han unido y han hecho precisamente eso.

App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

4.1.1.- Historia

La plataforma se puso a disposición del público el 25 de diciembre de 2008 y está dirigida a personas que no están familiarizadas con la programación y con la ayuda que nos brinda la

informática. En la creación de App Inventor, Google se basó en investigaciones previas significativas en informática educativa y sirve para crear páginas.

4.1.2.-Características y funciones

El editor de bloques de la plataforma App Inventor, utilizaba anteriormente la librería Open Blocks.

1.- Basado en [httpy Blockly] de JavaScript para crear un lenguaje visual. Estas librerías están distribuidas por Massachusetts Institute of Technology bajo su licencia libre.

El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza *Kawa* como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation.

2.- Permite crear una aplicación en menos tiempo que otros, y se pueden programar aplicaciones más complejas en mucho menos tiempo que con los lenguajes más tradicionales, basados en texto.

Inicialmente desarrollado por el profesor Hal Abelson y un equipo de Google Educación, mientras que Hal pasaba un año sabático en Google, App Inventor se ejecuta como un servicio web administrado por personal del Centro del MIT para el aprendizaje

móvil –una colaboración de MIT de Ciencia Computacional e Inteligencia Artificial de laboratorio (CSAIL) y el Laboratorio de Medios del MIT–. El App Inventor contaba en 2015 con una comunidad mundial de casi dos millones de usuarios que representaban a 195 países en todo el mundo. Más de 85 mil usuarios semanales activos de la herramienta han construido más de 4,7 millones de aplicaciones de Android. Una herramienta de código abierto que pretende realizar la programación y la creación de aplicaciones accesibles a una amplia gama de audiencias.

3.- La interfaz gráfica: permite al usuario crear aplicaciones con muchas funcionalidades. Al alcance de unos cuantos clics, por lo tanto se abre una gran puerta para muchas personas indiscriminadas que deseen crear aplicaciones sin necesidad de ser programador.

Asimismo, indicar que existe un sin fin de tutoriales en los cuales se puede aprender acerca del funcionamiento de las posibilidades de Mit App Inventor para adquirir destreza en el uso de esta poderosa herramienta.

4.1.3.- La herramienta

Como se ha comentado anteriormente, MIT App Inventor es una herramienta en línea, y su dirección de acceso es: <http://ai2.appinventor.mit.edu>

Los requisitos de uso son:

- Disponer de un equipo PC (Windows, Mac, Linux) con conexión a Internet.
- Tener una cuenta Google: usuario y contraseña.
- Se recomienda utilizar un navegador web como Google Chrome o Mozilla Firefox .

El proceso de creación de una app con MIT App Inventor consta de 3 fases:

1.- Diseñador de pantallas. Se crean las distintas ventanas o pantallas que contendrá la aplicación. En ellas sitúan sus componentes: imágenes, botones, textos ... y se configuran sus propiedades.

2.- Editor de bloques. Permite programar de forma visual e intuitiva el flujo de funcionamiento del programa utilizando bloques. Cada objeto dispone de unos métodos específicos que es posible invocar personalizando sus parámetros de llamada.

3.- Generador de app. Al finalizar las fases de diseño y programación, se genera el instalador APK de la aplicación. Se puede obtener un código QR para su descarga temporal desde el móvil o bien el propio archivo APK para descargar, publicar en la nube y/o enviar a otros usuarios/as.



Imagen 2. Proceso de creación de una app con MIT App Inventor

4.1.- Pasos para creación de app

A continuación se muestran unos enlaces de interés:

- Guía de iniciación a App Inventor:
<https://codeweek.eu/docs/spain/guia-iniciacion-app-inventor.pdf>
- Kio4: tutoriales: <http://kio4.com/appinventor/index.htm>
- Cómo se programan apps para móviles con App Inventor:
<http://soloelectronicos.com/2014/04/21/como-se-programan-apps-moviles-con-mit-app-inventor-bloques-logicos/>

4.2.- Estructura de la app

La aplicación está compuesta por seis pantallas, cuyos nombres, ordenados por orden de diseño, son:

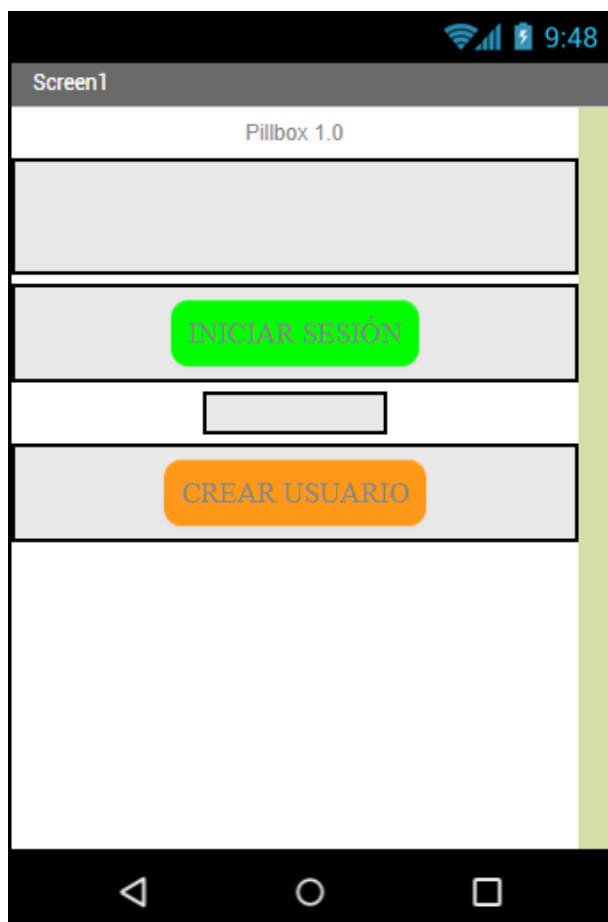
Screen1
Screen_ErrorLogin
Screen_LoginUsuario
Screen_NuevoUsuario
Screen_Pastillero
Eliminar_medicamento

A lo largo de este apartado, que será el más extenso de la memoria, se va a proceder a exponer y comentar cada una de dichas pantalla y bloques, tanto su diseño como su programación dentro de cada una de ellas.

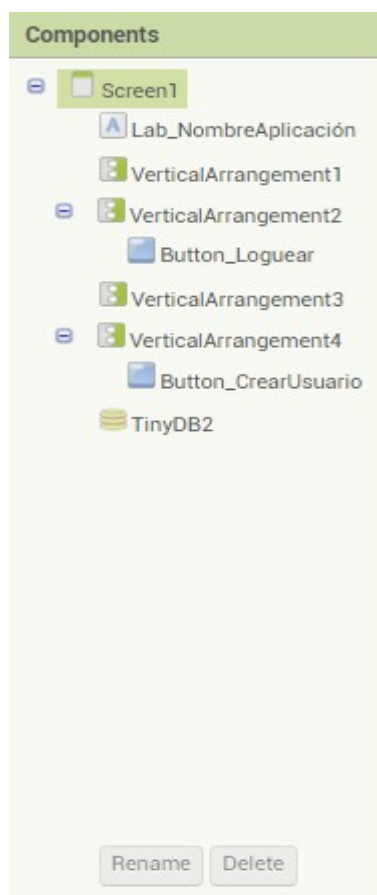
En App Inventor 2, la pantalla de inicio siempre se llama *Screen1*. Siempre debe tener ese nombre y no se puede renombrar. Las demás pantallas pueden tener cualquier nombre, aunque una vez creadas no se pueden renombrar. Esta es la razón por la cual la pantalla *Eliminar_medicamento* no tiene el nombre *Screen_EliminarMedicamento*.

Screen1

Ésta es la primera pantalla creada de la app, su diseño es el siguiente:



4.2.- Screen1, imagen 2



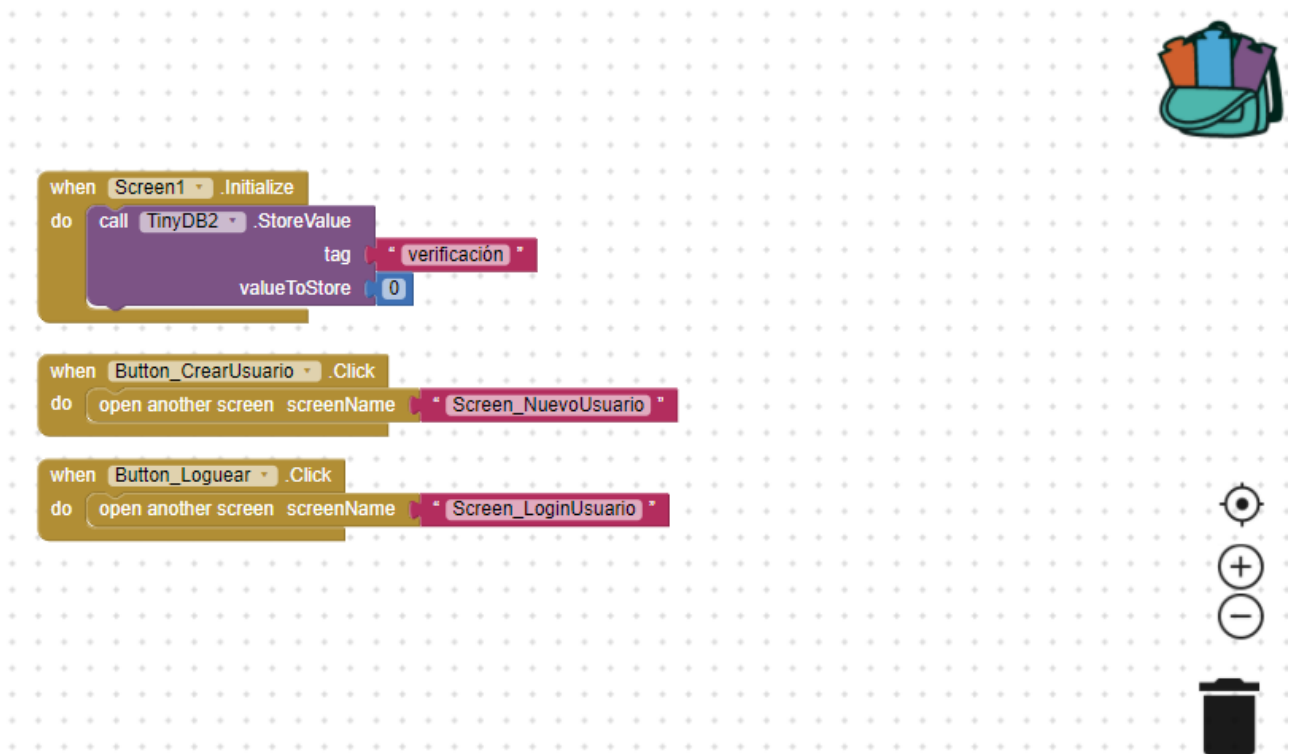
4.1.- Screen1, imagen 3

Básicamente, la pantalla *Screen1* está compuesta por los botones comentados en el apartado 3.1.- de esta memoria, así como por una etiqueta que le da nombre a la app. Como se puede observar en la imagen de "Componentes", se han añadido varios *VerticalArragment* con el único objetivo de situar los botones en una posición concreta de la pantalla. Asimismo, se ha incluido una base de datos TinyDB2 de nombre T2 para interactuar con otras pantallas y cuya función se comentará más adelante, en el apartado referente a la pantalla *Screen_Pastillero*.

Los nombres que se le han dado a los componentes con los que se puede programar son representativos de sus funcionalidades para facilitar la programación en bloque que se irá comentando en

esta y las siguientes pantallas de la app.

El diseño de bloques de esta primera pantalla quedaría tan sencillo como:

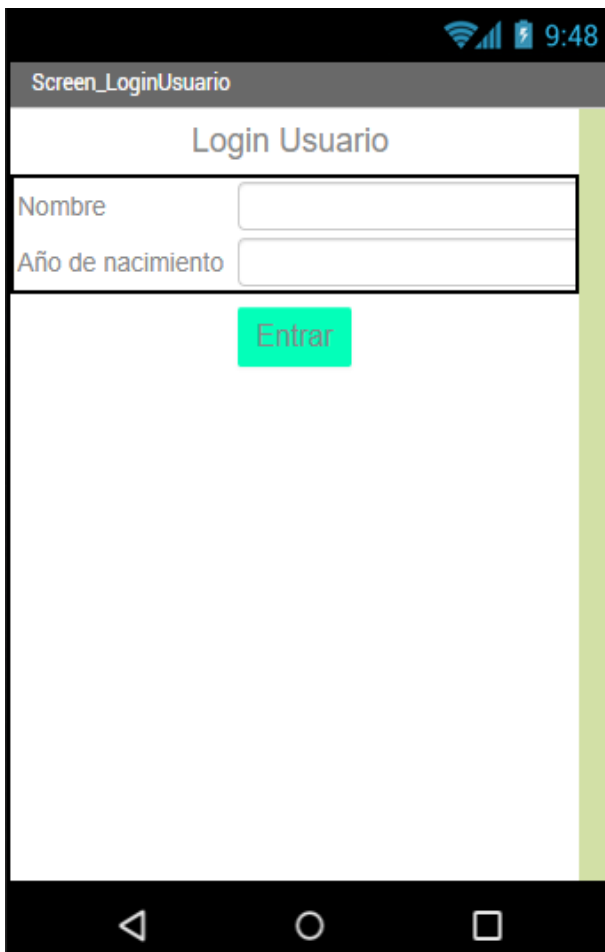


4.1.- Screen1 Diseño de bloques

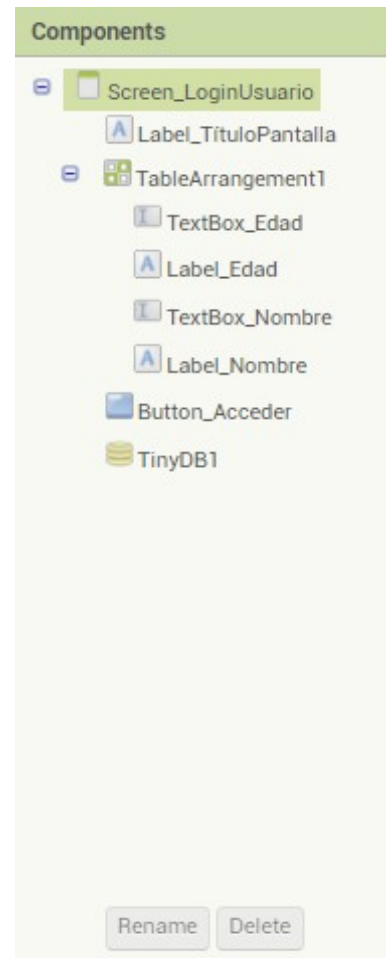
Cuando el usuario presione el botón “Crear usuario” (llamado *Button_CrearUsuario*), la app abrirá una nueva ventana, creada previamente, llamada *Screen_NuevoUsuario*, donde podrá introducir sus datos y obtener la planificación de sus tomas de medicamentos. De la misma manera, al presionar el botón “Iniciar sesión” en la pantalla de diseño (llamado éste *Button_Loguear*), se abrirá una nueva ventana llamada *Screen_LoginUsuario*, donde podrá el usuario loguearse para acceder a sus datos y planificación guardada.

Screen_LoginUsuario

A esta pantalla se accede al pulsar el botón “Iniciar sesión” de la pantalla principal. El diseño y componentes de la pantalla a comentar son los siguientes:



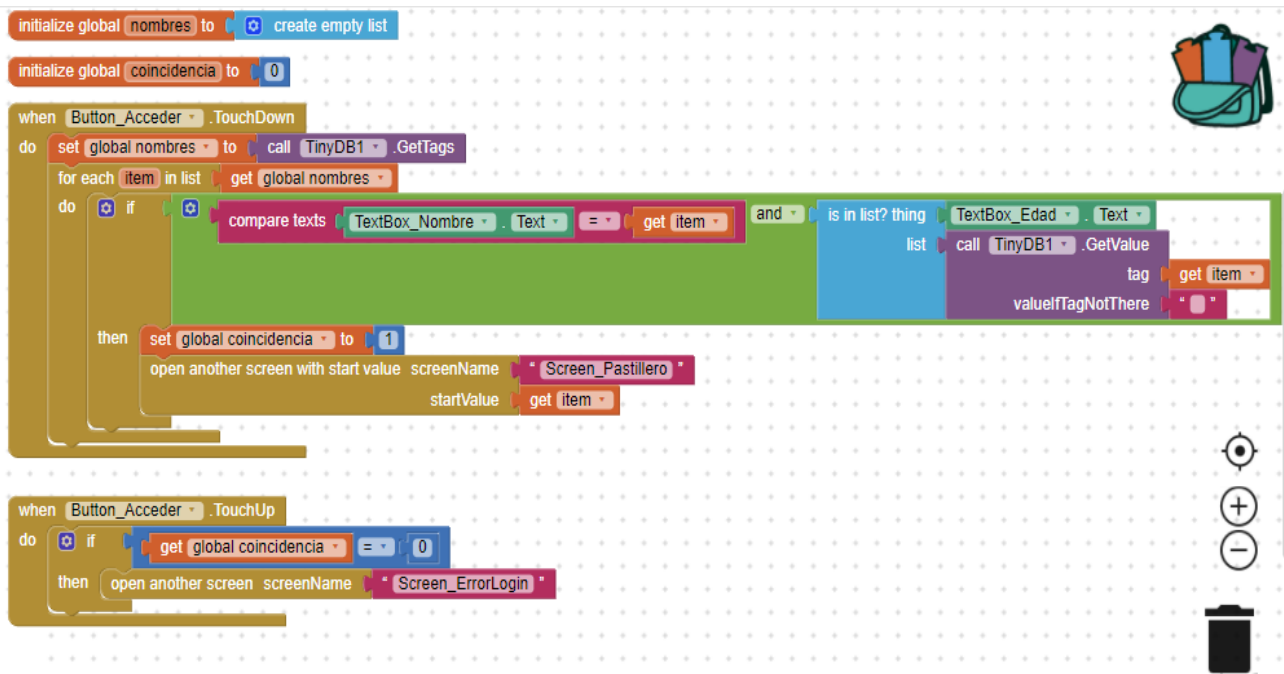
4.1.- Screen_loginUsuario, imagen 3



4.1.- Screen_loginUsuario, imagen 4

Básicamente la pantalla se compone de un título (*Login* usuario, usando una etiqueta), la información que solicita la pantalla (utilizando dos etiquetas) y un lugar para escribirla (utilizando dos cajas de texto), y un botón para poder acceder a la cuenta del usuario. Se puede observar también que también hay

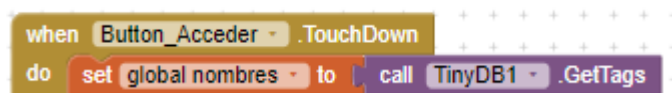
una base de datos TinyDB1 (de nombre T1). Ésta es necesaria para poder poder acceder a los datos del usuario una vez fuera creada su cuenta y tras haber cerrado la aplicación.



4.1.- Screen_loginUsuario, diseño de bloques

El diseño de bloques de esta pantalla es el siguiente:

Bloque 1.- Cada vez que el usuario presione el botón “Entrar”, una lista vacía (*nombres*) se cargará con los nombres de las etiquetas de la base de datos TinyDB1 (los nombres de las etiquetas se guardan con el mismo nombre que utilizó el usuario al crear su cuenta).



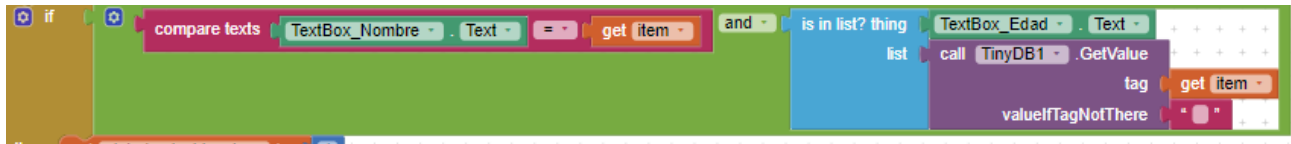
4.1.- Screen_loginUsuario, subbloques 1

Para cada uno de estos nombres se realiza una comprobación:



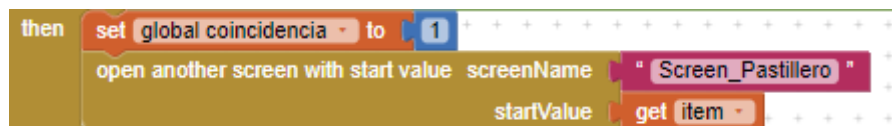
4.1.- Screen_loginUsuario, subbloques 2

Se comparan el nombre y edad introducidos por el usuario con todos los nombres de las etiquetas de TinyDB1 y valores de las mismas (la fecha de nacimiento es uno de los datos que se almacenan en las TinyDB1).



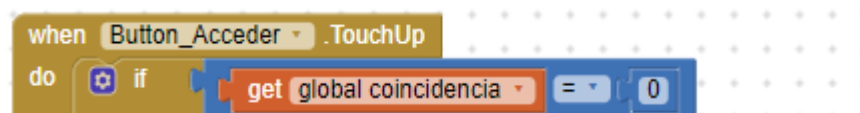
4.1.- Screen_loginUsuario, subbloques 3

Si coinciden el nombre y edad del usuario con los guardados en TinyDB1 se pondrá la variable *coincidencia* a 1 (ha sido creada con valor 0) y se abrirá una nueva ventana, dando paso a *Screen_Pastillero* y obteniendo un valor de comienzo correspondiente al nombre del usuario cuyo nombre estaba guardado en la aplicación.



4.1.- Screen_loginUsuario, subbloques 4

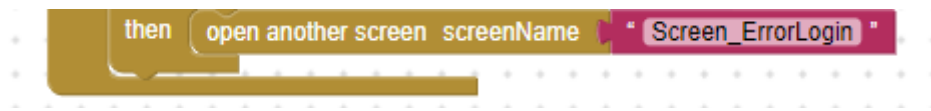
Bloque 2.- Al soltar el botón “Entrar” se hace una comprobación que consiste en comparar el valor de *coincidencia*.



4.1.- Screen_loginUsuario, subbloques 5

Si éste es 0 eso quiere decir que se han introducido datos que no coinciden con ningún usuario conocido por la app, ya que, como se mencionó anteriormente el valor es 1 si hay coincidencia, y se procede en este caso a abrir una nueva ventana

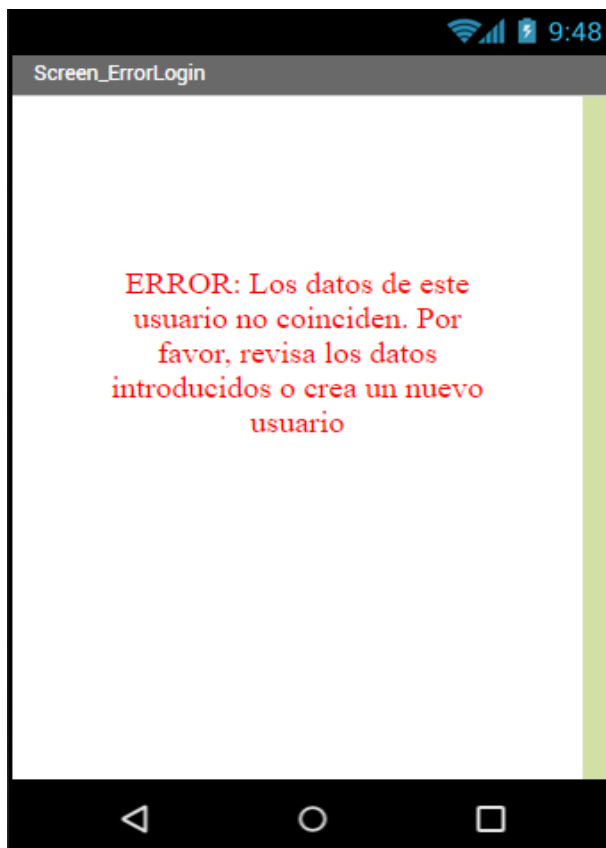
(*Screen_ErrorLogin*) de información advirtiéndole que ha habido un error.



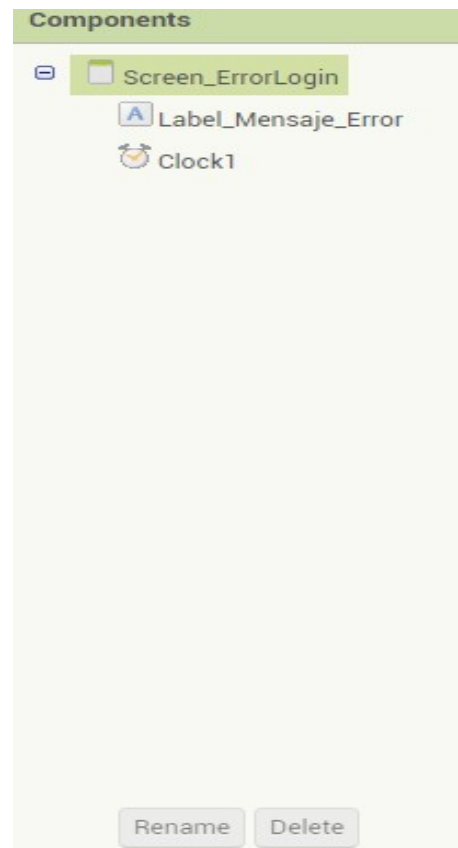
4.1.- *Screen_loginUsuario*, subbloques 6

Screen_ErrorLogin

A esta pantalla se podrá acceder solamente en el siguiente caso: Al intentar iniciar sesión un usuario que anteriormente no hubiera creado una cuenta o un usuario ya registrado en la app que hubiera introducido mal los datos de acceso a la misma.



4.1.- *Screen_ErrorLogin*, imagen 2

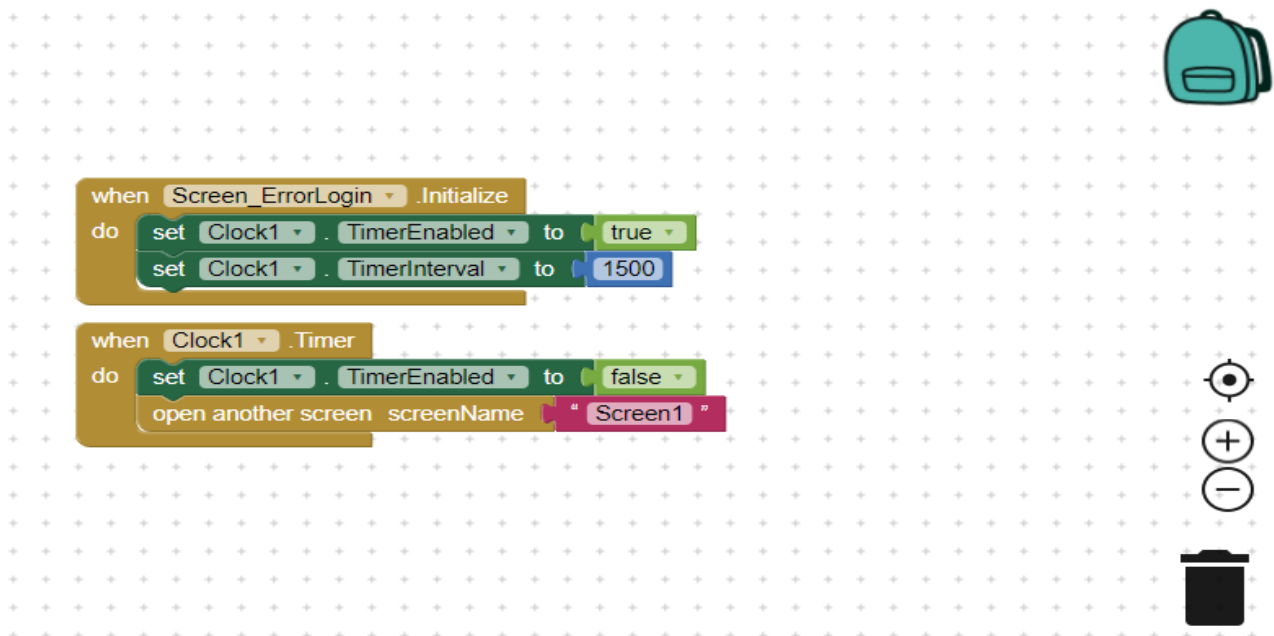


4.1.- *Screen_ErrorLogin*, imagen 3

El diseño de esta pantalla consiste solamente en una etiqueta con un mensaje de error, el cual permanecerá visible durante un breve período de tiempo para después regresar a la pantalla

anterior *Screen_LoginUsuario*. Se ha introducido como componente un reloj para realizar una de las funciones anteriores.

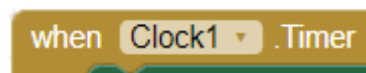
El diseño de bloques queda de la siguiente manera:



4.1.- *Screen_ErrorLogin*, diseño de bloques

Las instrucciones de los bloques serían las siguientes:

Cuando se abra la pantalla actual (*Screen_ErrorLogin*) se habilitará *Reloj1* y se establecerá un intervalo de tiempo para el mismo de 1500 ms, es decir, el reloj comenzará a contar desde 0 hasta llegar a los 1500 ms tras lo cual se activará el *Reloj1.Timer*



4.1.- *Screen_ErrorLogin*,
subbloques 1

Una vez activado se procederá a inhabilitar al mismo *Reloj1* y abrirá la pantalla *Screen1*, devolviendo al usuario así a la pantalla principal de la aplicación.

Screen_NuevoUsuario

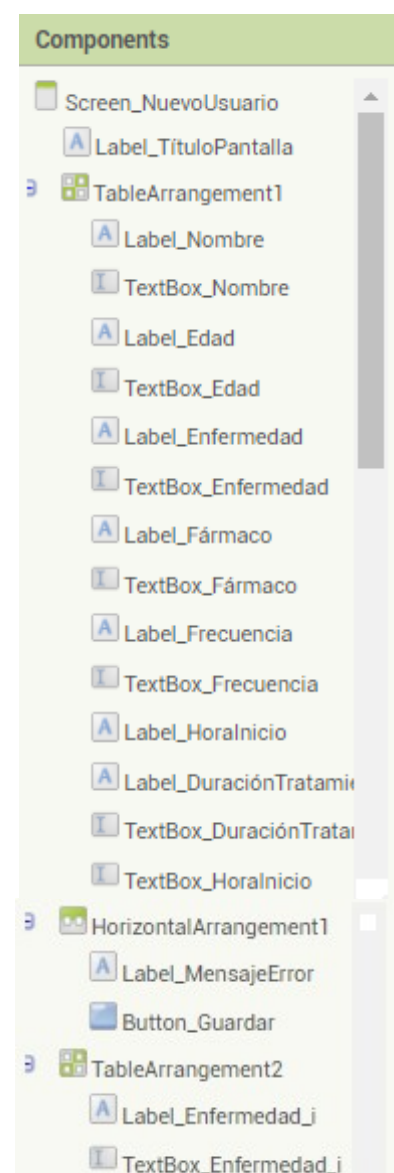
A esta pantalla se puede acceder de dos modos:

- 1.- Al pulsar el botón "Crear usuario", desde *Screen1*, o bien
- 2.- Al pulsar el botón "Añadir" fármaco desde *Screen_Pastillero*

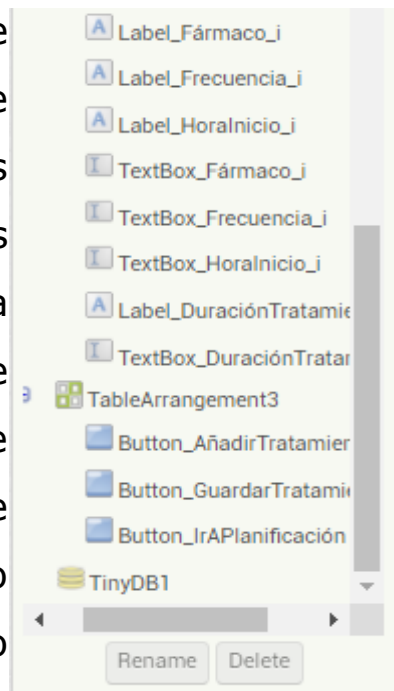
El diseño y componentes de la pantalla son los siguientes:

The screenshot shows a mobile application interface for creating a new user. At the top, the status bar displays signal strength, battery level, and the time 9:48. The app title bar reads "Screen_NuevoUsuario". Below this, the main title "NUEVO USUARIO" is centered. The form consists of two sections. The first section contains labels and text boxes for "Nombre", "Año de nacimiento", "Enfermedad", "Fármaco", "Frecuencia (h)", "Inicio (hora)", and "Duración tratamiento (d)". Below these is a green "Guardar" button. The second section contains labels and text boxes for "Enfermedad", "Fármaco", "Frecuencia (h)", "Inicio (h)", and "Duración tratamiento (d)". At the bottom of the form, there are three buttons: "Añadir tratamiento", "Guardar", and "Ir a Planificación". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

4.1.- *Screen_NuevoUsuario*, imagen 7

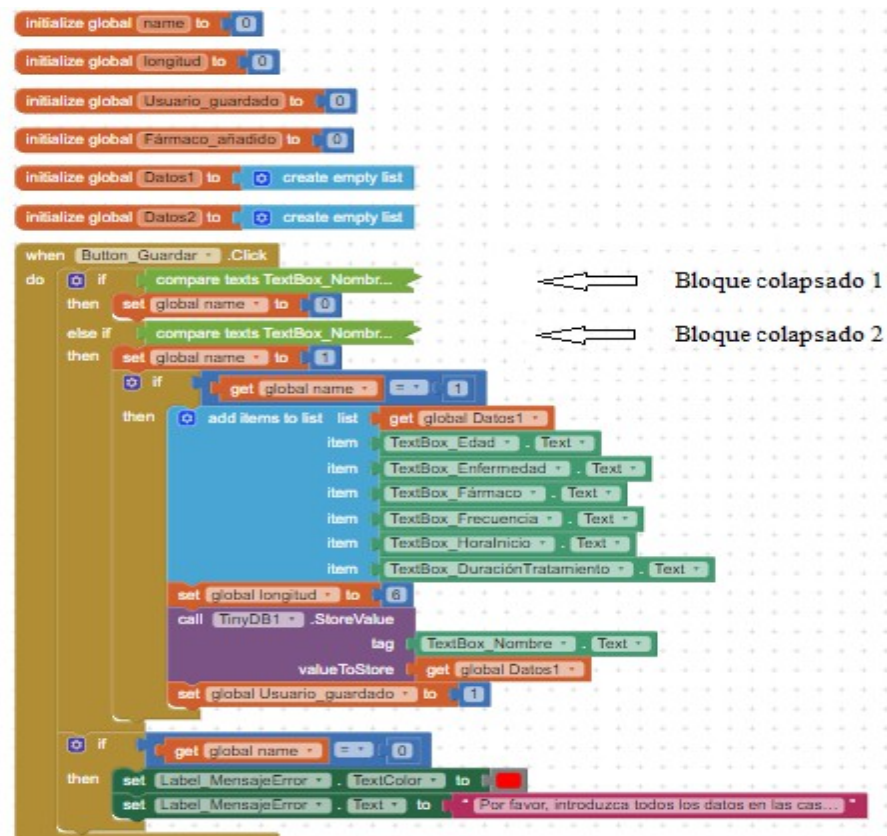


La pantalla está compuesta básicamente por un título (una etiqueta), la información que solicita la pantalla (doce etiquetas) con sus respectivos lugares para recibirla (doce cajas de texto), cuatro botones, una etiqueta para mostrar posibles mensajes de error y una base de datos TinyDB1, , cuya función es la de guardar todos los datos del tratamiento de cada usuario. Esta TinyDB tendrá, como etiqueta el nombre del usuario, y como valores, serán los siguientes (los cuales se guardan en modo de lista): Año de nacimiento, enfermedad, frecuencia, hora de inicio de la primera toma y duración del tratamiento. El diseño de bloques de esta ventana es el siguiente:



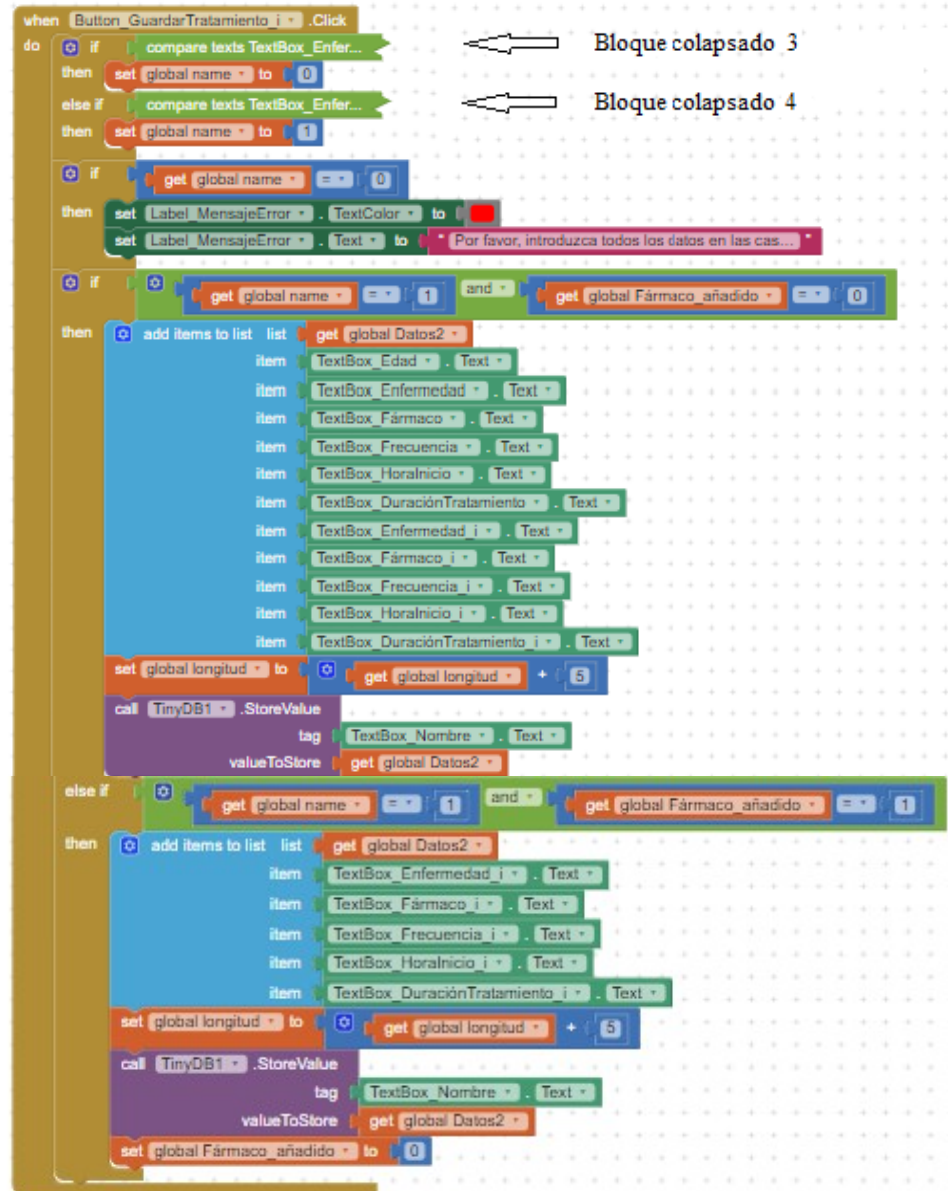
4.1.- Screen_NuevoUsuario, imagen 8

Bloque 1



4.1.- Screen_NuevoUsuario, diseño de bloques, imagen 1

Bloque 2



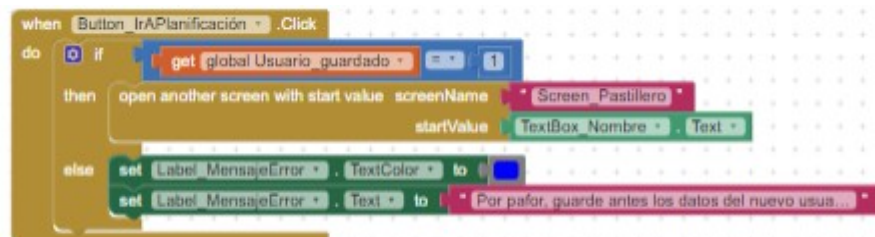
4.1.- Screen_NuevoUsuario, diseño de bloques, imagen 2

Bloque 3



4.1.- Screen_NuevoUsuario, diseño de bloques, imagen 3

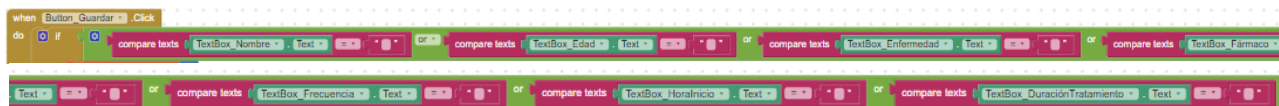
Bloque 4



4.1.- Screen_NuevoUsuario, diseño de bloques, imagen 4

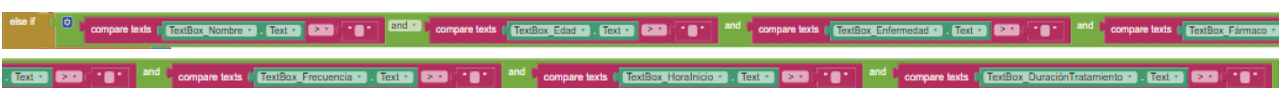
Donde los bloques colapsados 1,2,3 y 4 que aparecen en las imágenes anteriores, por falta de espacio, se han representado compactados y, mostrados de forma expandida son:

Bloque expandido 1



4.1.- Screen_NuevoUsuario, subbloques 1

Bloque expandido 2



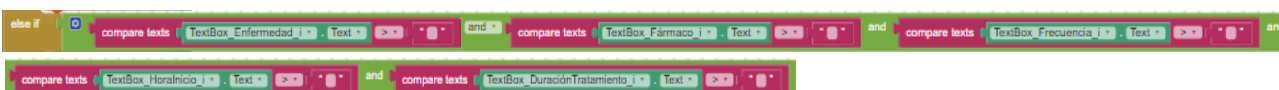
4.1.- Screen_NuevoUsuario, subbloques 2

Bloque expandido 3



4.1.- Screen_NuevoUsuario, subbloques 3

Bloque expandido 4



4.1.- Screen_NuevoUsuario, subbloques 4

Las instrucciones de los bloques son, básicamente, las siguientes:

Bloque 1.- Tras haber introducido el usuario los datos que solicita la pantalla y pulsar el botón “Guardar”, la aplicación comenzará realizando unas comprobaciones:

Bloque 1



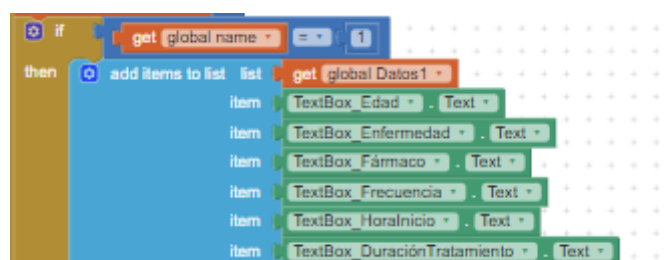
→ Bloque colapsado 1
→ Bloque colapsado 2

4.1.- Screen_NuevoUsuario, subbloques 5

Habiendo definido anteriormente la variable “name” con valor 0, si se cumple la condición del bloque compactado 1 (alguna o todas de las casillas están vacías) se tendrá que el usuario no ha rellenado todas las cajas de texto correspondientes a toda la

información necesaria para el primer fármaco , por lo que "name" seguirá valiendo 0. En el caso que todas las cajas de texto correspondientes a este primer fármaco hayan sido cumplimentadas (condición que viene representada por la condición del bloque compactado 2), "name" adoptará un valor 1. (Estos valores de "name" se utilizarán, o bien para guardar los datos introducidos por el usuario en la cadena vacía creada llamada "Datos1", o bien para mandar un mensaje de error indicando que no se han introducido todos los datos necesarios.

- *Name* = 1. Este valor lo toma "name" si se cumple la condición del bloque compactado 2, por lo que hay datos en todas las siete primeras cajas de texto, y se procede a guardar la información de dichas cajas (todas menos la correspondiente al nombre del usuario, ya que éste será le etiqueta de la base de datos TinyDB1) en una lista vacía creada de nombre "Datos1". Para ello se utiliza la función "agregar elementos a la lista", tomando los elementos (las cajas de texto) como argumentos individuales.



4.1.- Screen_NuevoUsuario, subbloques 6

El siguiente paso es guardar en una base de datos TinyDB1 la información que se tiene ahora en la lista "Datos1", y para ello se utilizará como etiqueta el propio nombre introducido por el usuario en la caja de texto correspondiente:



4.1.- Screen_NuevoUsuario, subbloques 7

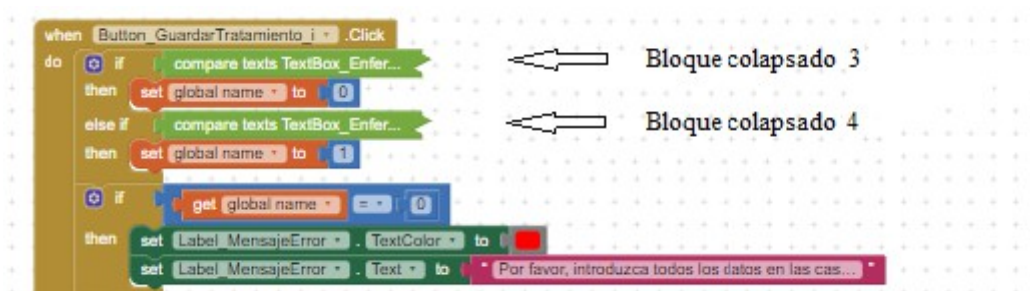
De esta manera, la información del usuario permanecería guardada y se podría consultar aunque se cerrara la aplicación. Se puede observar también que, tras guardar los datos en la TinyDB1, la variable creada "Usuario_guardado" e iniciada con valor 0, adquiere el valor 1 (condición que será necesaria para poder cambiar más adelante de pantalla, como se comentará en su bloque correspondiente).

- *Name* = 0. Este valor lo toma "name" si se cumple la condición del bloque compactado 1, significando entonces que al menos una de las cajas de texto está vacía y procediendo a mostrar un mensaje de error indicando esta situación.



4.1.- Screen_NuevoUsuario, subbloques 8

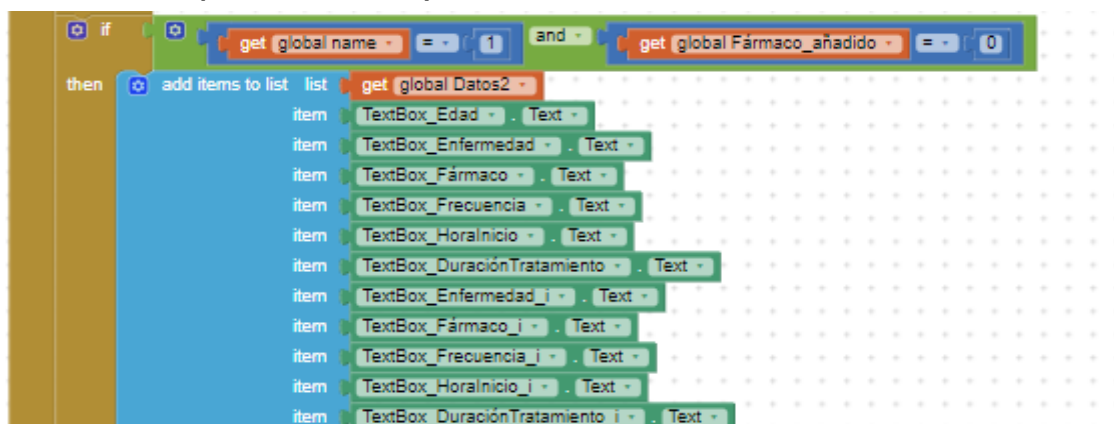
Bloque 2.- De la misma manera que ocurría en el bloque 1, al pulsar esta vez el botón "Guardar tratamiento" se realiza una comprobación de datos básica (si las últimas cinco casillas están todas o no cumplimentadas). Dicha comprobación es llevada a cabo por los Bloques colapsados 3 y 4, que pueden observarse extendidos en unas páginas más arriba.



4.1.- Screen_NuevoUsuario, subbloques 9

Y como ocurría también antes, mostrando un mensaje de error en caso que no se hayan escrito todos los datos antes de “Guardar tratamiento”.

A continuación, si se dan ciertas condiciones (“name” = 1, o que hayan datos en todas las casillas, y la variable “Fármaco_añadido” = 0) se añadirá a la lista vacía Datos2, los datos de las casillas de la pantalla debidamente cumplimentadas, exceptuando la primera (el nombre), ya que ésta será para nombrar la etiqueta de TinyDB1.

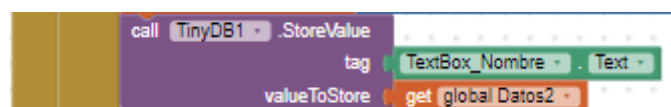


4.1.- Screen_NuevoUsuario, subbloques 10

La funcionalidad de la variable “Fármaco añadido” es el siguiente: si se guardan por primera vez todas las casillas de la pantalla (es decir, sólo los correspondientes a dos fármacos), “Datos2” comenzará a llenarse con ellas (excepto con la primera, el

nombre, como ya se ha comentado), siendo este caso el correspondiente a "Fármaco_añadido" = 0. Pero si ya se han guardado ambos fármacos y se introducen los datos para un tercero pulsando el botón "Añadir tratamiento", entonces se le irán añadiendo a la lista "Datos2" sólo los elementos compuestos por las últimas cinco casillas, siendo este otro el caso correspondiente a "Fármaco_añadido" = 1 (se mostrará cuando se trate el Bloque 3, ya que es aquí donde puede adquirir el valor 1).

A continuación, se guardan los valores de la lista "Datos2" en la base de datos TinyDB1, utilizando como etiqueta el nombre del usuario:



4.1.- Screen_NuevoUsuario, subbloques 11

Si, al pulsar el botón "Guardar tratamiento", las variables "name" = 1 y "Fármaco añadido" = 1 (porque se ha añadido un tercero al menos), se añadirá a la lista "Datos2" los elementos de las últimas cinco casillas de la pantalla



4.1.- Screen_NuevoUsuario, subbloques 12

Y, para finalizar el Bloque 2 se procede a guardar en la base de datos TinyDB1 los valores guardados en la lista "Datos2" utilizando, como siempre, de etiqueta el nombre del usuario:



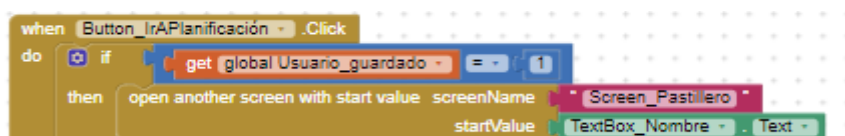
4.1.- Screen_NuevoUsuario, subbloques 13

Bloque 3.- Este bloque es muy sencillo: Al pulsar el botón “Añadir tratamiento”, las cinco últimas cajas de texto, correspondientes a los datos del tratamiento de un fármaco completo, se limpiarán para que el usuario pueda escribir más datos para otro fármaco si esa fuera su intención. Asimismo, la variable “Fármaco_añadido” toma el valor 1, para así poder añadir posteriormente en la lista “Datos2” solamente los valores correspondientes a las últimas cinco casillas mencionadas y no todos los datos introducidos que aparecen en la pantalla en ese momento:



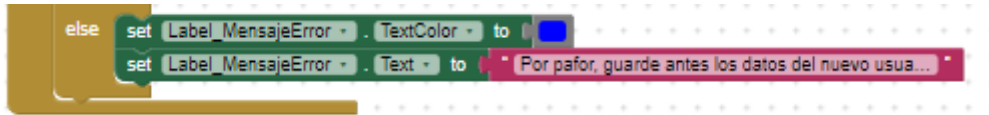
4.1.- Screen_NuevoUsuario, subbloques 14

Bloque 4.- Para finalizar con la pantalla *Screen_NuevoUsuario* es necesario pulsar el botón “Ir a Planificación”, el cual tiene asociadas las siguientes instrucciones: Al pulsarlo se comprobará si el usuario ha guardado, al menos, los datos correspondientes un fármaco relativo a las primeras casillas. Si es así, la variable “Usuario_guardado” tendrá el valor 1 y se cumplirá la condición del bloque procediendo a abrir otra ventana de nombre “Screen_Pastillero” y a la cual le enviará un valor de comienzo dado por el valor contenido en la caja de texto correspondiente al nombre del usuario.



4.1.- Screen_NuevoUsuario, subbloques 15

En el caso que el usuario no hubiera guardado los datos del primer fármaco al menos, aparecerá un mensaje de error indicando de la necesidad de guardar los datos antes de poder cambiar de pantalla.

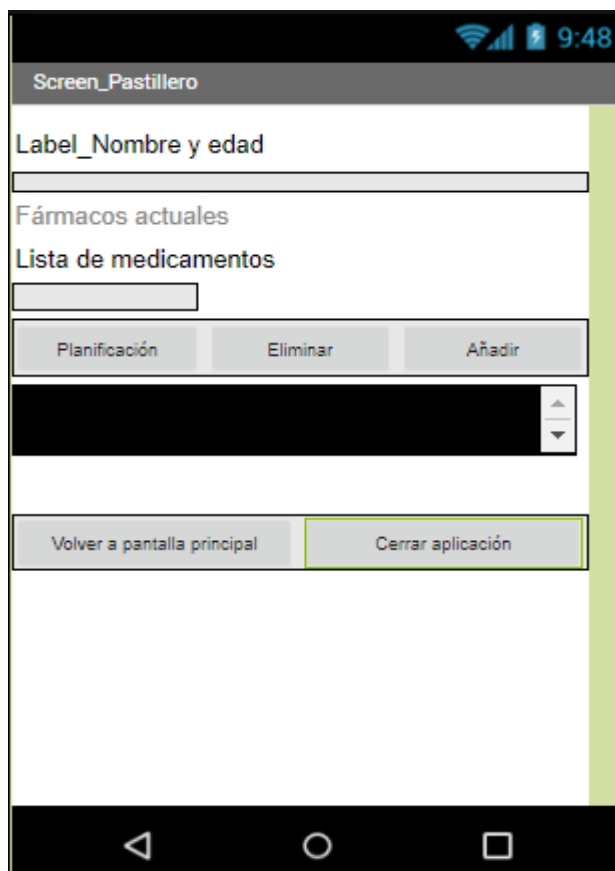


4.1.- Screen_NuevoUsuario, subbloques 16

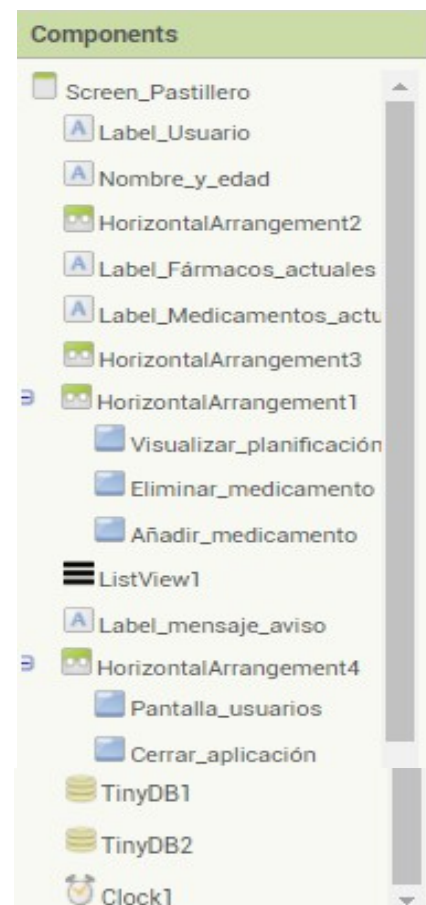
SCREEN_PASTILLERO

A esta pantalla es posible acceder a través de dos pantallas:

- 1.- Pulsando el botón “Ir a Planificación” desde *Screen_NuevoUsuario*,
- 2.- Desde *Eliminar_medicamento*, pulsando “Volver”.



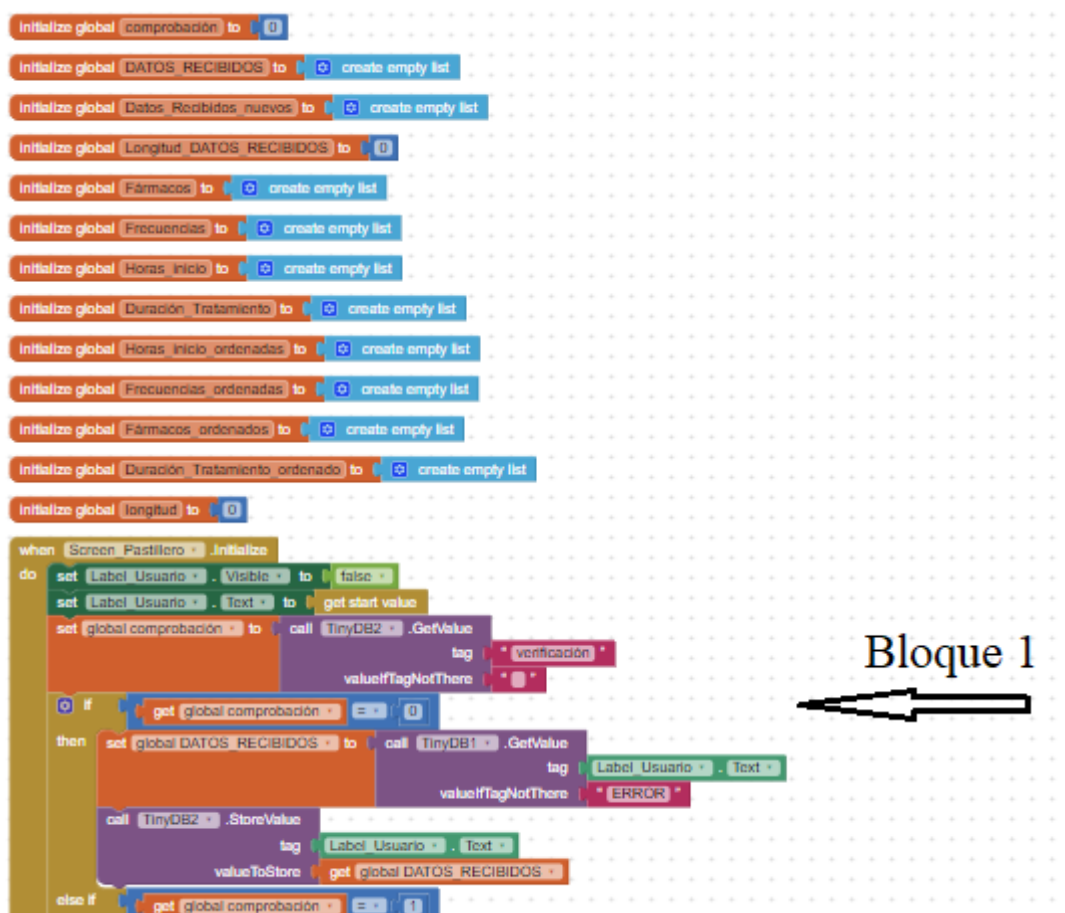
4.1.- Screen_Pastillero, imagen 6

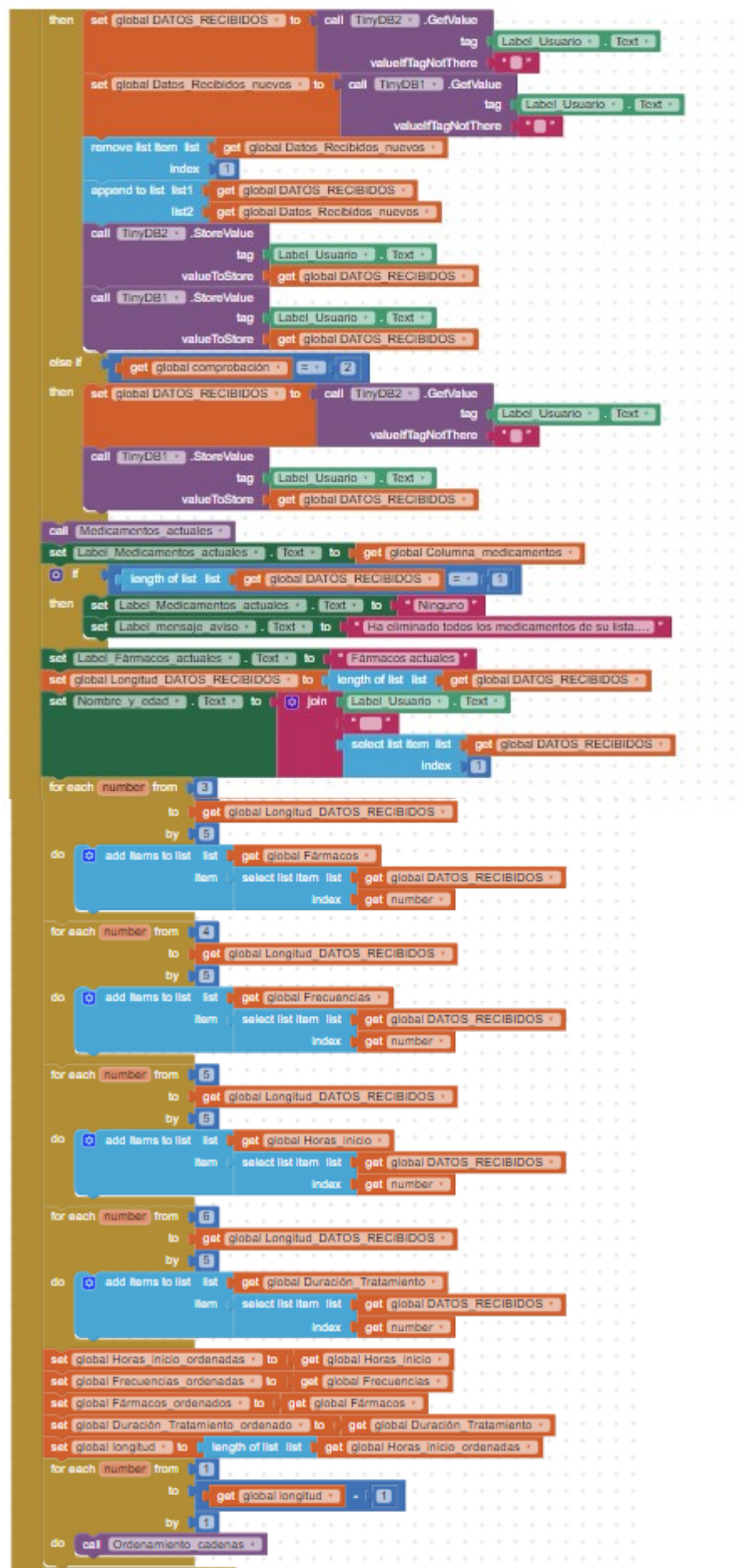


4.1.- Screen_Pastillero, imagen 7

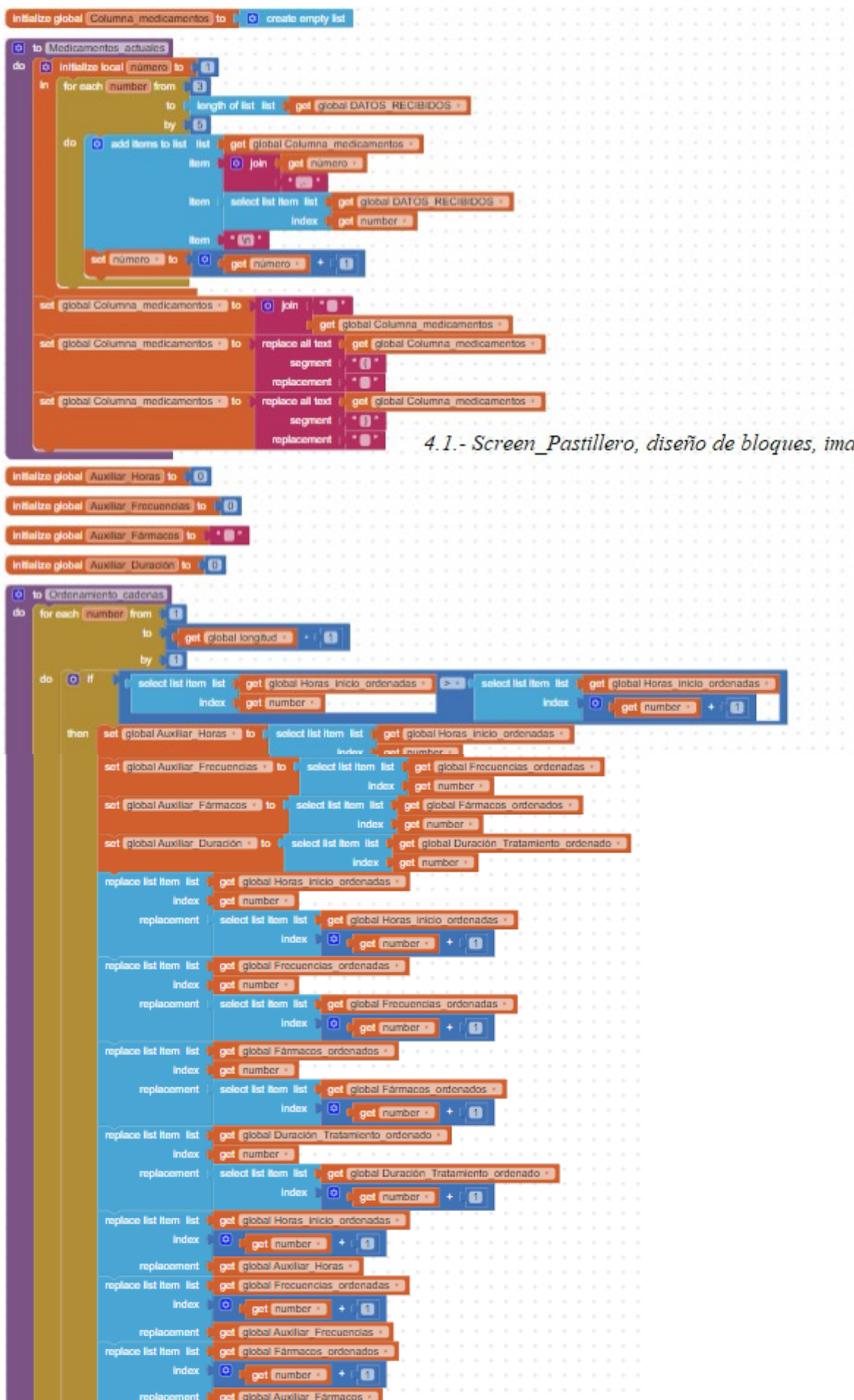
El diseño de la pantalla está compuesto, básicamente, por dos títulos, uno con el nombre y año de nacimiento (etiqueta) y el otro dando paso a la lista de medicamentos (etiqueta), una lista de los fármacos actuales (etiqueta), cinco botones para cada una de las opciones que ofrece la ventana, un "ListView", el cual ofrecerá una lista con toda la información necesaria para el usuario para sus tomas de medicamentos, otra etiqueta para mostrar posibles mensajes de error o información, dos TinyDB (TinyDB1 y TinyDB2) y un reloj Clock1, el cual será necesario para que la app calcule la planificación del tratamiento a seguir por el usuario.

A continuación se muestra el diseño de bloques, donde la creación de las variables están en orden de aparición en los bloques correspondientes.

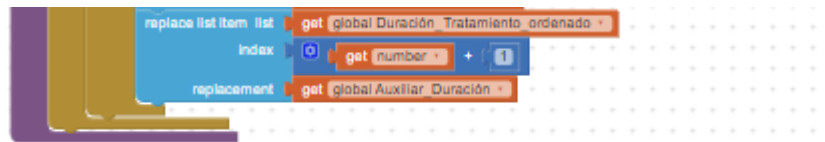




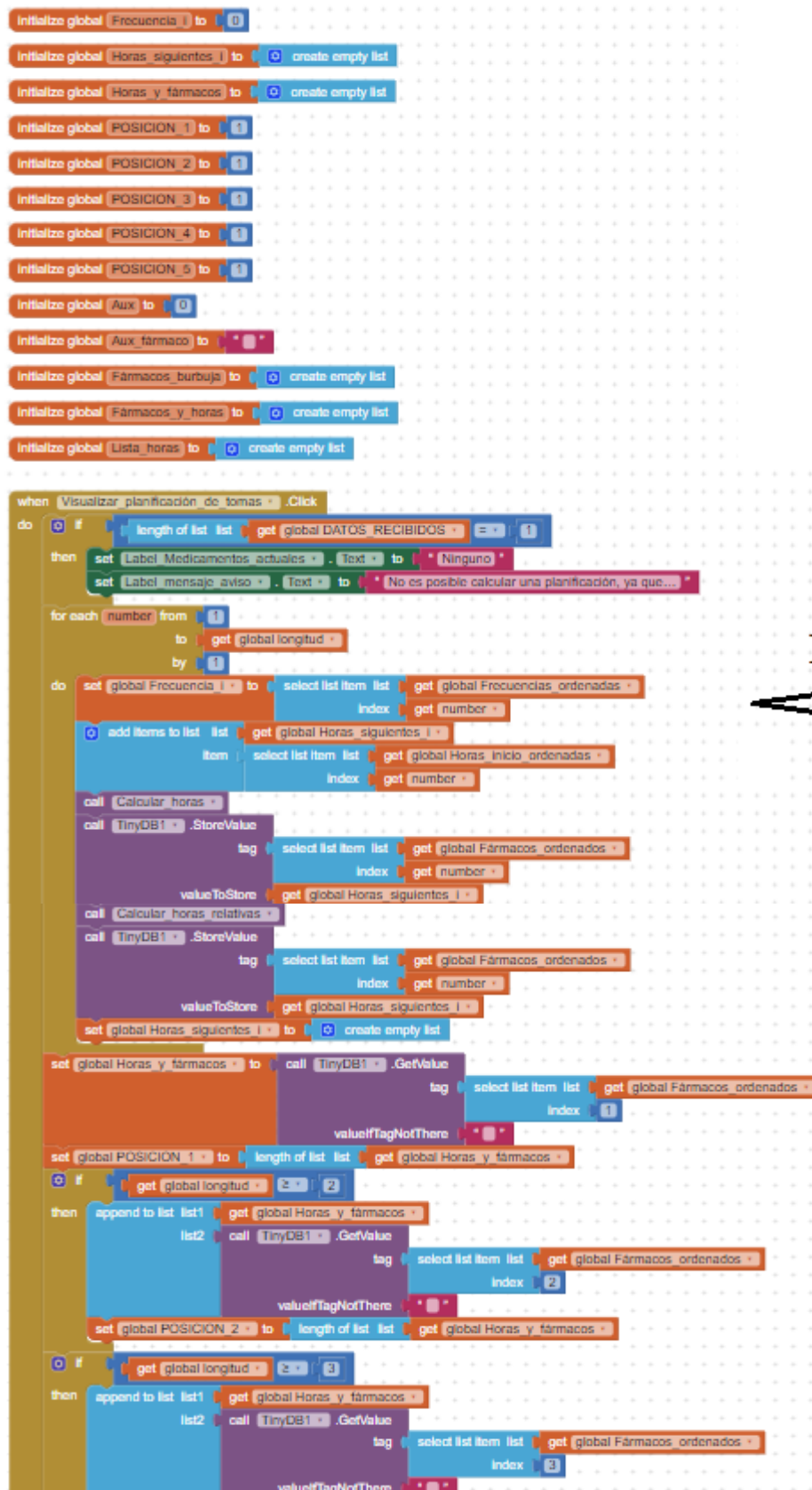
4.1.- Screen_Pastillero, diseño de bloques, imagen 1



4.1.- Screen_Pastillero, diseño de bloques, imagen 2

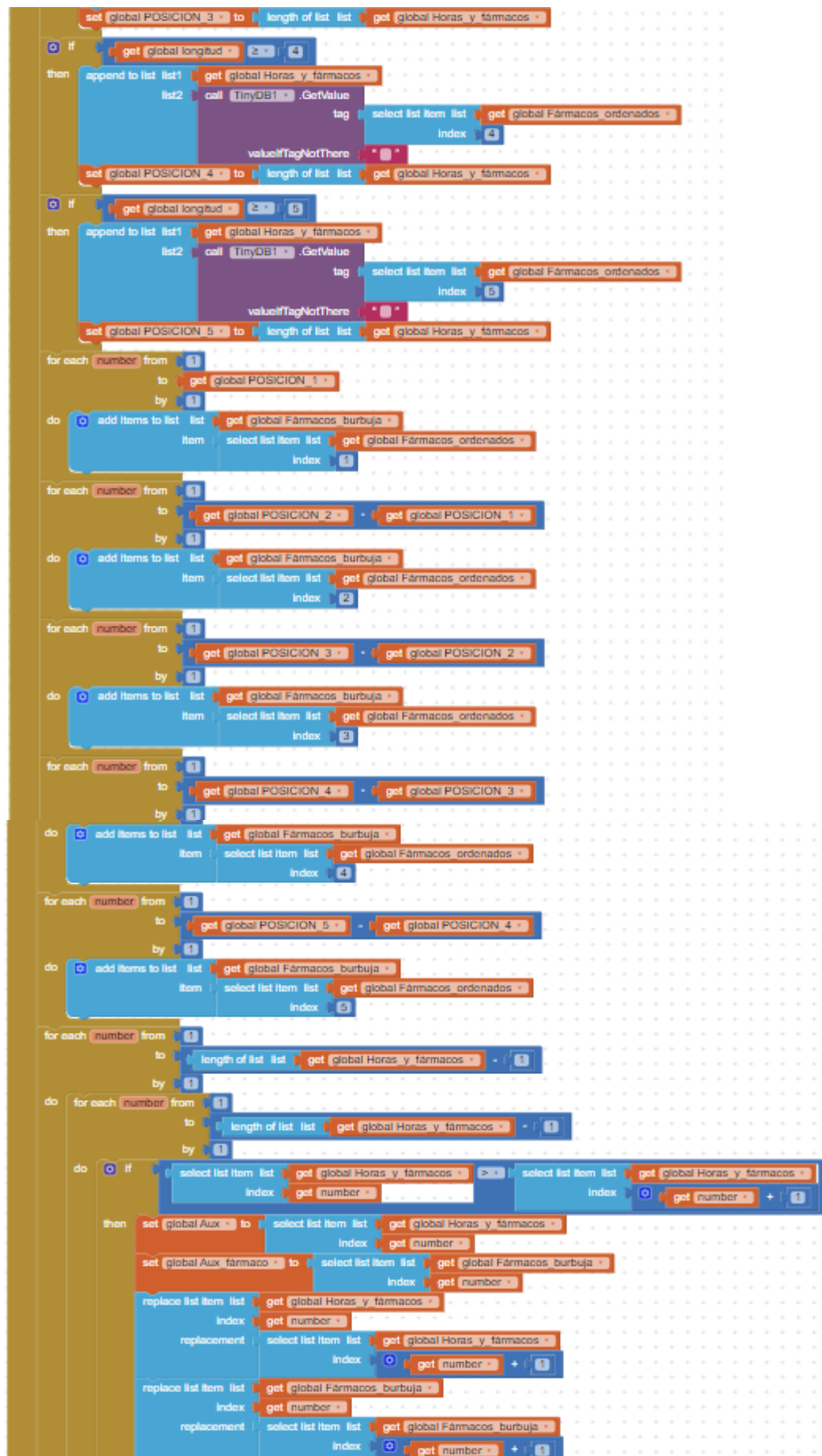


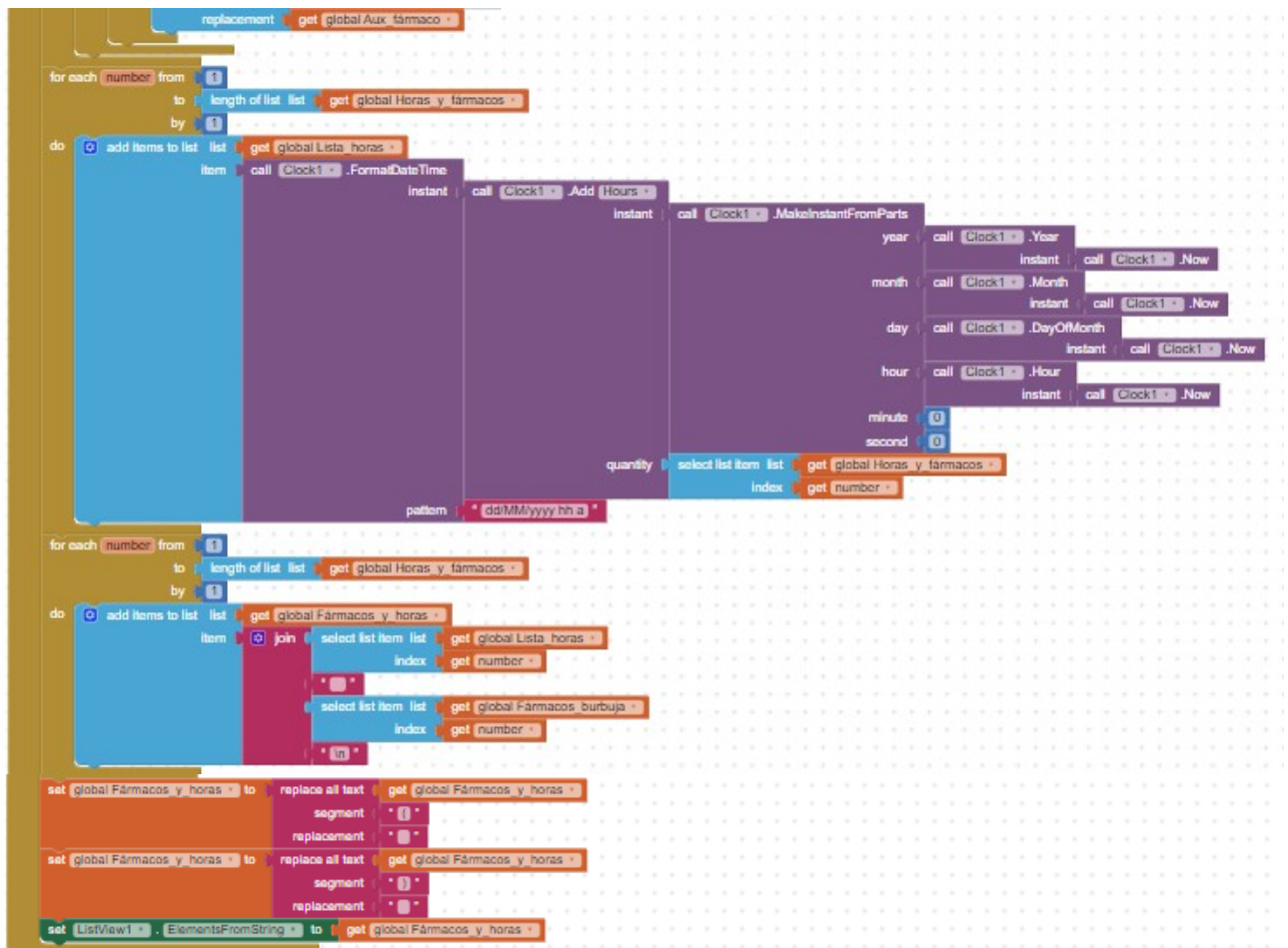
4.1.- Screen_Pastillero, diseño de bloques, imagen 3



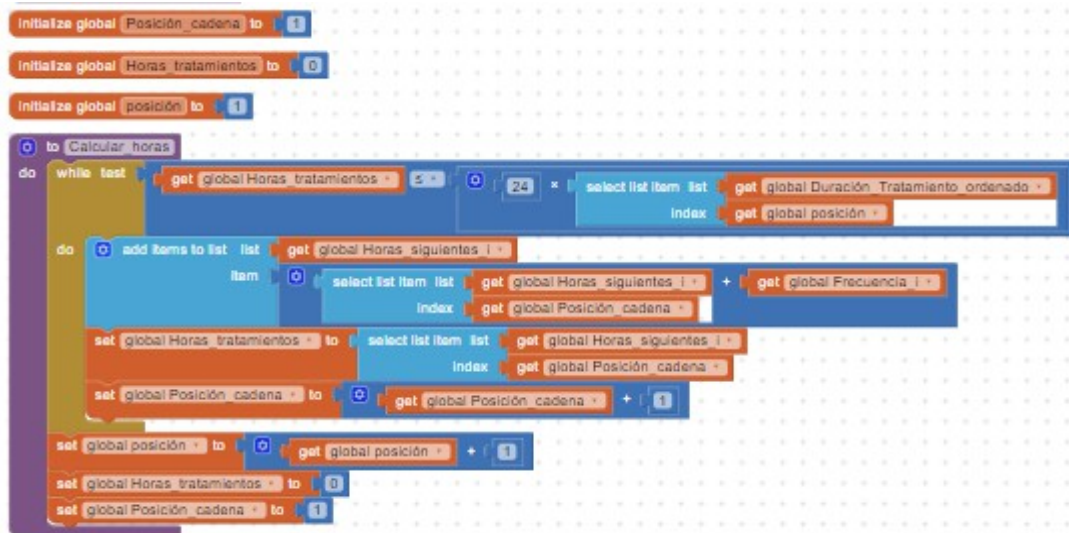
Bloque 2



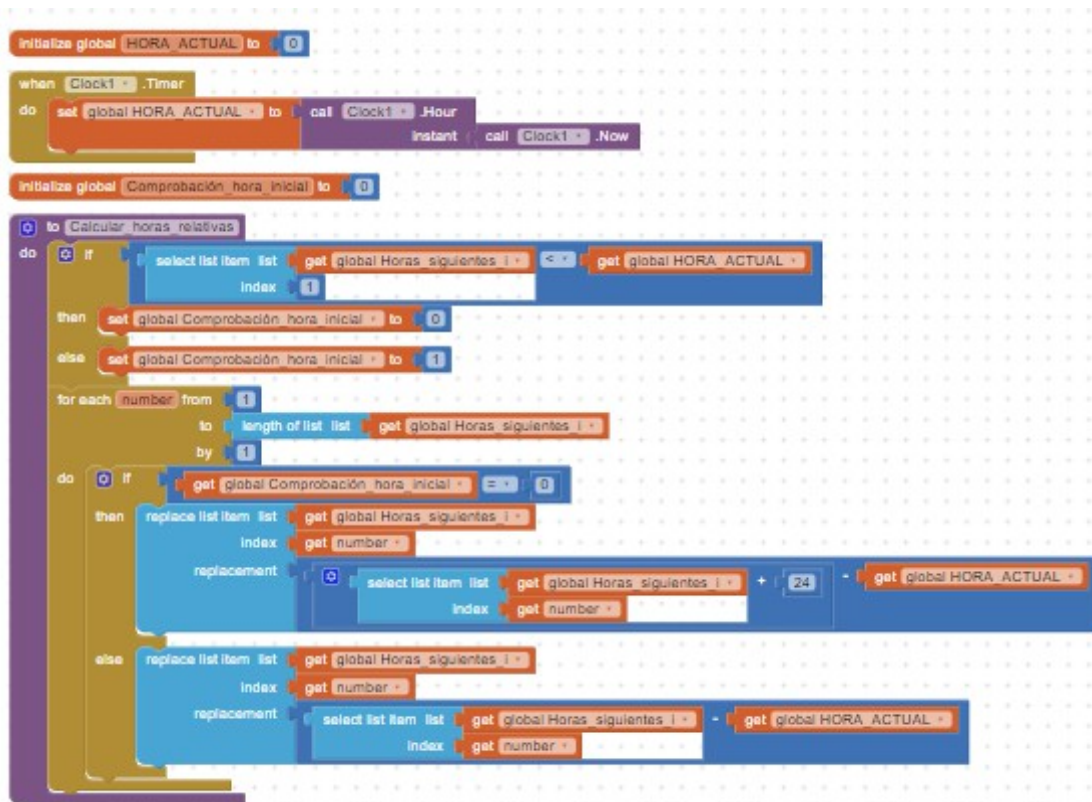




4.1.- Screen_Pastillero, diseño de bloques, imagen 4



4.1.- Screen_Pastillero, diseño de bloques, imagen 5



4.1.- Screen_Pastillero, diseño de bloques, imagen 6



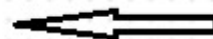
Bloque 3



4.1.- Screen_Pastillero, diseño de bloques, imagen 7



Bloque 4



4.1.- Screen_Pastillero, diseño de bloques, imagen 8



Bloque 5



4.1.- Screen_Pastillero, diseño de bloques, imagen 9



Bloque 6

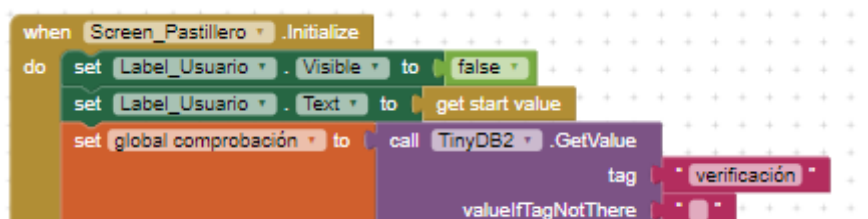


4.1.- Screen_Pastillero, diseño de bloques, imagen 10

Las instrucciones de los bloques son, básicamente, las siguientes:

Bloque 1.- Se va a comentar el bloque 1 a la vez que los procedimientos que incluye, y ésto se hará cuando aparezcan los mismos. La ejecución de este bloque sólo requiere como condición que se haya entrado en la pantalla *Screen_Pastillero*.

Al abrirse la pantalla, ésta recibe como datos, tanto si se proviene de *Screen_NuevoUsuario* tanto si es de *Eliminar_medicamento*, el nombre del usuario, adquiriendo dicho dato utilizando "get start value", y en este caso se asocia dicho valor a la etiqueta "Label_Usuario". También se asocia la variable "comprobación" a un valor importante almacenado en la base de datos TinyDB2 con etiqueta "verificación".

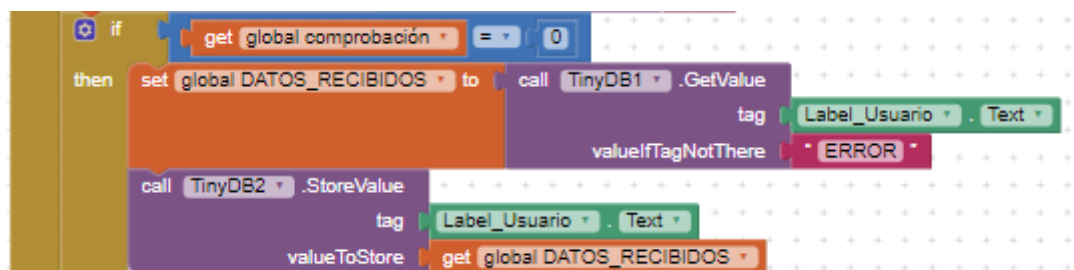


4.1.- *Screen_Pastillero*, subbloques 1

La funcionalidad de TinyDB2 y dicha etiqueta es la siguiente: Como se ha utilizado la misma pantalla *Screen_NuevoUsuario* tanto para introducir datos por primera vez como para añadir más tarde nuevos medicamentos, el programa debe conocer si es o no la primera vez que se introducen datos, ya que para el manejo de los mismos se han utilizado dos TinyDB (TinyDB1 y TinyDB2, con nombre de espacio T1 Y T2) para que no hubiera solapamiento de datos. De esta manera, si es la primera vez que se accede a la

pantalla *Screen_Pastillero* (ya fuera creando nuevo usuario o accediendo a través del inicio de sesión), el valor de **TinyDB2** de etiqueta "**verificación**" valdrá **0**. Este valor se le asigna en la pantalla principal *Screen1*.

De esta manera, si es la primera vez que se accede a esta pantalla (no habiendo entonces utilizado la misma para añadir o eliminar medicamentos), la variable "comprobación" vale 0, y por lo tanto se guardará en la lista "DATOS_RECIBIDOS" los elementos obtenidos de TinyDB1 con etiqueta "Label_Usuario" (en *Screen_NuevoUsuario* se guardaron los elementos en cuestión en dicha base de datos).



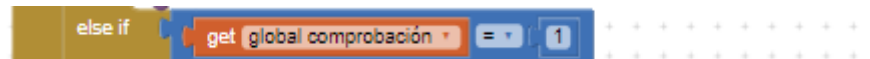
4.1.- *Screen_Pastillero*, subbloques 2

NOTA: "DATOS_RECIBIDOS" tiene el siguiente orden de datos: [Año de nacimiento, enfermedad 1, fármaco1, frecuencia1, inicio1, duración1, enfermedad2, fármaco2, frecuencia2, inicio2, duración2, ..., enfermedad5, fármaco5, frecuencia5, inicio5, duración5].

Asimismo se procede a guardar en TinyDB2 dichos datos utilizando como etiqueta la misma que en TinyDB1 para mayor simplicidad.

En el caso que se haya accedido, desde esta pantalla, a *Screen_NuevoUsuario* para añadir nuevos medicamentos (se comprobará en uno de los últimos bloques de la pantalla), el valor

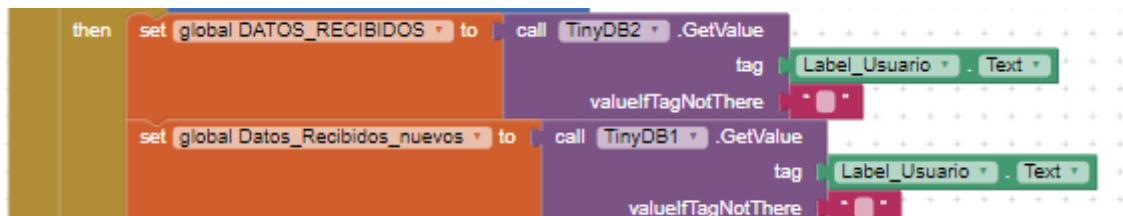
de **TinyDB2** de etiqueta **"verificación"** valdrá **1**, por lo que el bloque que se está analizando entraría en la siguiente condición (Recuérdese que se ha optado por grabar en la variable "comprobación" el valor guardado en la TinyDB2 de etiqueta "Verificación" para mayor simplicidad):



4.1.- Screen_Pastillero, subbloques 3

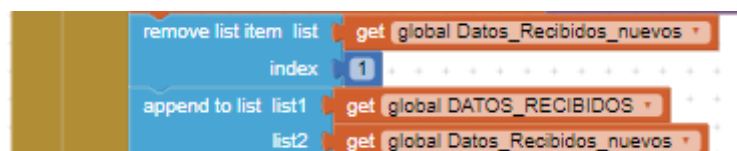
y las instrucciones que se ejecutarían en este caso serían:

a).- Copiar los datos de las etiquetas "Label_Usuario" (donde está escrito el nombre del usuario) TinyDB1 y TinyDB2 en las listas "DATOS_RECIBIDOS" y "Datos_Recibidos_nuevos". El objetivo de esto es el siguiente: Tener siempre disponible en la TinyDB2 los datos actuales del usuario y, cada vez que se desee añadir un nuevo medicamento guardar éste en TinyDB1.



4.1.- Screen_Pastillero, subbloques 4

b).- Eliminar de la lista "Datos_Recibidos_nuevos" el primer elemento, ya que éste corresponderá a la edad del mismo y este dato no se utiliza para calcular la planificación de sus tomas.



4.1.- Screen_Pastillero, subbloques 5

Asimismo también se añade al final de la lista "DATOS_RECIBIDOS" la lista "Datos_Recibidos_nuevos" para añadir los nuevos medicamentos que se hayan añadido.

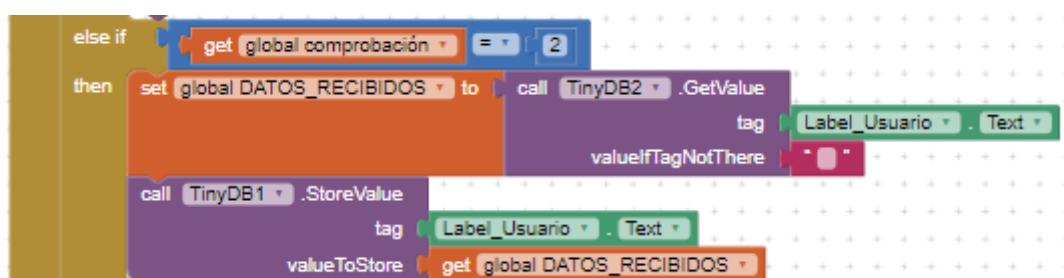
c).- Finalmente, la condición “comprobación” = 1 tiene también la siguiente secuencias asociada:



4.1.- Screen_Pastillero, subbloques 6

Donde TinyDB2 de etiqueta “Label_Usuario” se actualiza adquiriendo los datos completos tanto de los medicamentos que se tenían antes como los añadidos tras haber incluido nuevos.

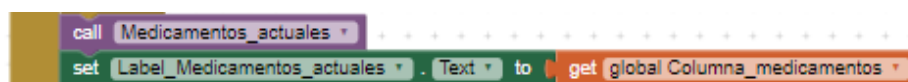
El último caso del valor de **TinyDB2** de etiqueta “**verificación**” se tiene al darse que “comprobación” valga **2**. Esto ocurrirá solamente si se ha accedido a la pantalla *Eliminar_medicamento* y se ha procedido a borrar alguno de ellos, procediendo a guardar en la variable “DATOS_RECIBIDOS” los elementos guardados (se procedió a guardarse en *Eliminar_medicamento* como se comprobará más adelante) en la TinyDB2 de etiqueta “Label_Usuario”, como se ha realizado en los dos casos anteriores (entrada por primera vez en la pantalla y por segunda tras haber añadido un nuevo medicamento).



4.1.- Screen_Pastillero, subbloques 7

La siguiente instrucción a ejecutarse corresponde a una llamada al procedimiento “Medicamentos actuales”, cuyo objetivo es

obtener una lista sólo de los medicamentos actualizados, para después copiar en un etiqueta dicha lista y mostrarla en pantalla, ya que "Label_Medicamentos_actuales" está habilitado y presente en la pantalla de diseño, de manera que así el usuario se asegura de ser todo correcto. A continuación se va a proceder a comentar dicho procedimiento para llevar un seguimiento más claro del programa.



4.1.- Screen_Pastillero, subbloques 8

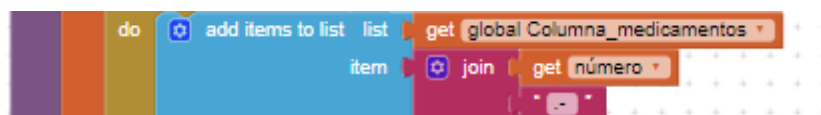
Procedimiento "Medicamentos actuales". Se trata de obtener una lista numerada de los medicamentos, de manera que se procede buscando primeramente los nombres de los medicamentos en la lista "DATOS_RECIBIDOS", del siguiente modo: con un bucle *for* solamente se pasan a una variable "Columna_medicamentos" el correspondiente al tercer elemento y a cada cinco elementos a partir de éste (ya que es en esa posición de la lista "DATOS_RECIBIDOS" donde se encuentran los nombres de los fármacos). Y esto se realiza hasta el final de la longitud de la lista "DATOS_RECIBIDOS"



4.1.- Screen_Pastillero, subbloques 9

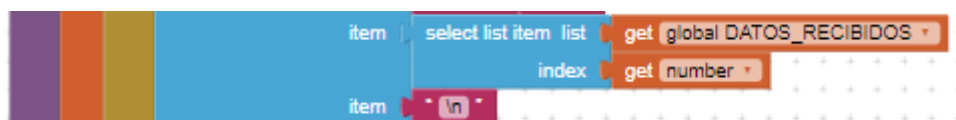
La forma de añadir los fármacos a "Columna_medicamentos" es utilizando bloques de cadena, añadiendo así los números correspondientes a cada fármaco que se presenta previamente al

mismo y en filas distintas. Se utiliza la función “añadir elementos a la lista” “Columna_medicamentos” escribiendo primeramente el número 1 (variable local “número” iniciada a 1) y sumando 1 en cada pasada del bucle (se añadiría en la primera ejecución “1.- “).



4.1.- Screen_Pastillero, subbloques 10

También se le añade el nombre del fármaco, obtenido de “DATOS_RECIBIDOS” y del lugar adecuado de la cadena (dado por “number”, el cual trabaja dentro del bucle “for” mencionado anteriormente. Y por último se le añade “\n” para que cada medicamento con su número correspondiente aparezca en otro renglón.



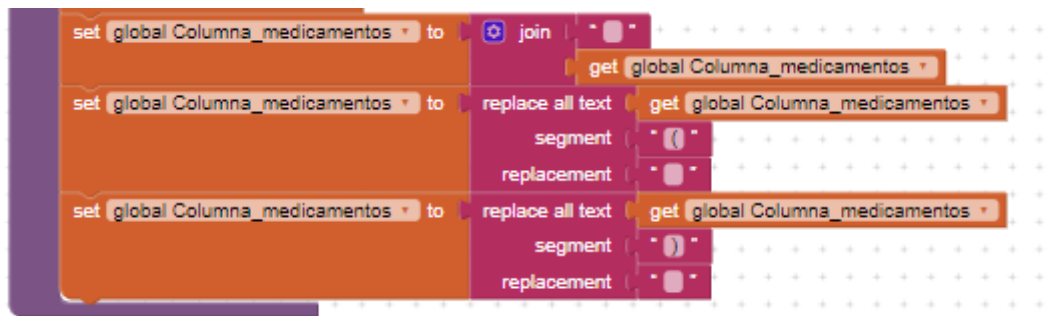
4.1.- Screen_Pastillero, subbloques 11

Asimismo, dentro del bucle “for” se incluye también la siguiente instrucción, para incrementar en 1 la numeración previa a los medicamentos, mientras no acabe dicho bucle.



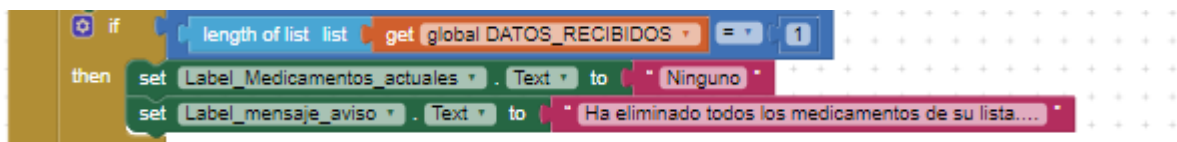
4.1.- Screen_Pastillero, subbloques 12

Para finalizar el procedimiento quedaría eliminar los paréntesis asociados a las lista por defecto, paréntesis que aparecen tanto al principio como al final de la lista (utilizando la función *replace all text* con sus dos argumentos respectivos). Esto se lleva a cabo reemplazando dichos paréntesis por espacios en blanco y grabando de nuevo el resultado en “Columna_medicamentos”.



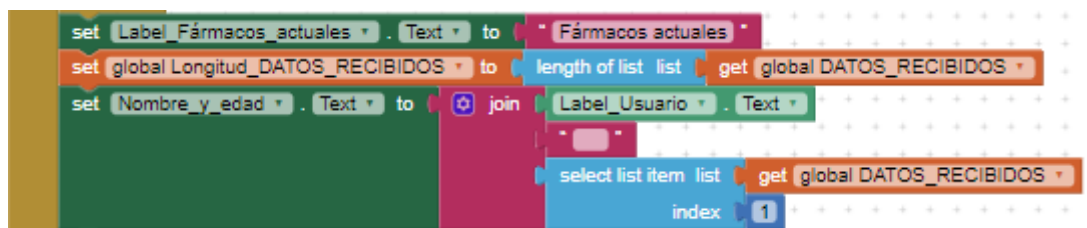
4.1.- Screen_Pastillero, subbloques 13

Continuando con el bloque 1, y considerando el caso que el usuario hubiera borrado todos los medicamentos en *Eliminación_medicamento* (esto se realizaría calculando la longitud de "DATOS_RECIBIDOS", de manera que si su valor es 1 significaría que sólo contiene el dato de la fecha de nacimiento y ningún fármaco entonces), se mostrará esta situación por pantalla mediante dos etiquetas ("Label_Medicamentos_actuales" y "Label_mensaje_aviso")



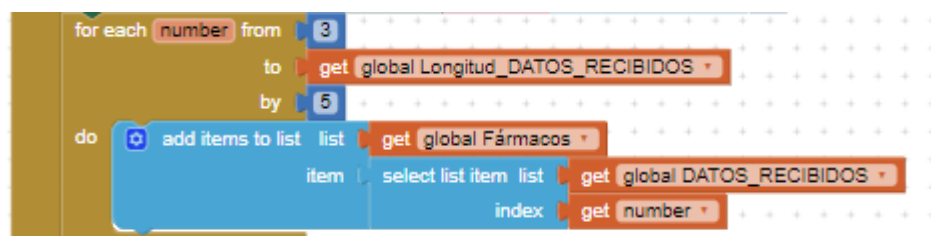
4.1.- Screen_Pastillero, subbloques 14

A continuación se escribe en una etiqueta que se muestra en pantalla un título: "Fármacos actuales" a partir de "Label_Fármacos_actuales". Después, en la variable "Longitud_DATOS_RECIBIDOS" se graba el resultado de obtener la longitud de la lista "DATOS_RECIBIDOS" y se procede a escribir en otra etiqueta ("Nombre_y_edad"), que se muestra por pantalla a modo de título mostrando el nombre del usuario (guardado en "Label_usuario") y su año de nacimiento (obtenido de seleccionar el elemento 1 de la lista "DATOS_RECIBIDOS").



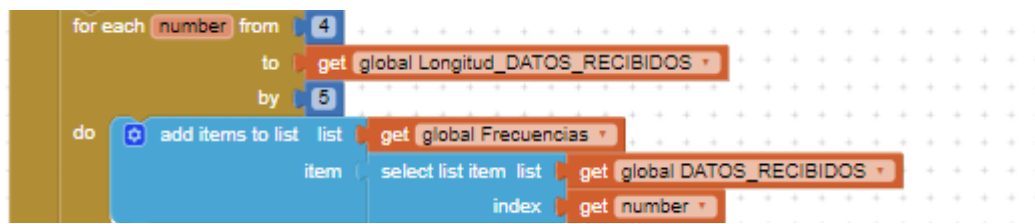
4.1.- Screen_Pastillero, subbloques 15

Se va a proceder a guardar, en la lista creada “Fármacos”, todos los fármacos incluidos en “DATOS_RECIBIDOS”, y esto se realizará utilizando un bucle “for”. Como el primer fármaco tiene la posición 3, y el siguiente aparecerá cinco posiciones consecutivas, se hace el bucle desde la posición tres, hasta la longitud de “DATOS_RECIBIDOS” y de cinco en cinco, y en dicho bucle se irán añadiendo a la lista “Fármacos” los elementos de la lista “DATOS_RECIBIDOS” donde están guardados los medicamentos (con la variable local “number” se escoge el índice correcto de la lista “DATOS_RECIBIDOS”).



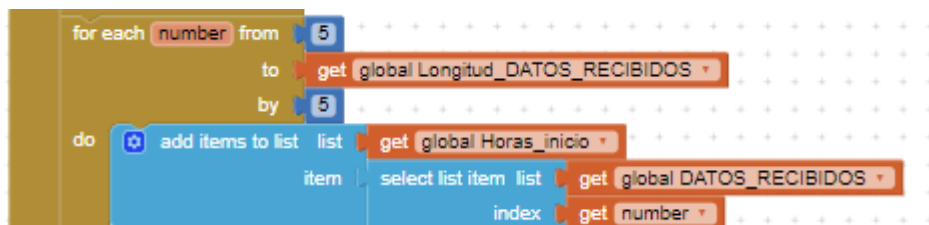
4.1.- Screen_Pastillero, subbloques 16

Asimismo se creó una variable en modo lista con el nombre “Frecuencias”, con el objetivo de guardar solamente las frecuencias de las tomas de fármacos. En este caso, en DATOS_RECIBIDOS, la primera frecuencia está en la posición cuatro, por lo que, procediendo como antes, se utilizará un bucle con iguales valores de incremento y condición de fin, así como la variable local “number” para escoger la posición correcta dentro de la lista DATOS_RECIBIDOS.



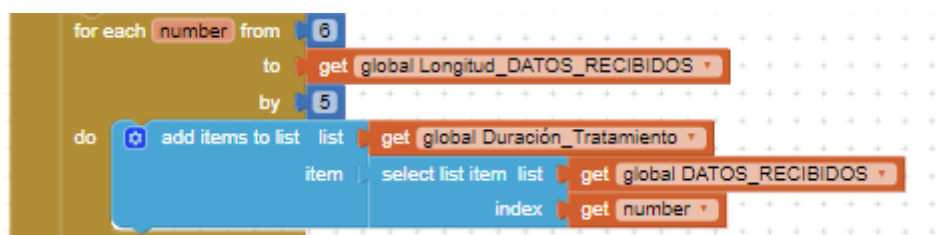
4.1.- Screen_Pastillero, subbloques 17

Procediendo de la misma manera para cumplimentar la lista creada "Horas_inicio" (en este caso el primer elemento de la lista "DATOS_RECIBIDOS" donde aparece la primera hora de inicio del tratamiento está en la posición cinco de la misma, valor que se utilizará en el bucle "for"):



4.1.- Screen_Pastillero, subbloques 18

Por último, en relación a las listas que se están creando, la lista "Duración_Tratamiento" albergará los valores de la duración de cada uno de los tratamientos. Tener en cuenta que el índice de la lista "DATOS_RECIBIDOS" donde se encuentra el primera valor de dichas duraciones de tratamientos es el seis.

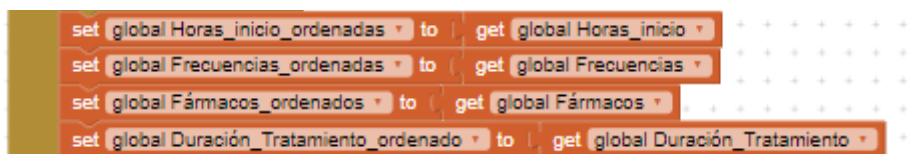


4.1.- Screen_Pastillero, subbloques 19

El paso siguiente ha sido ordenar todas las listas anteriores, con el fin de ordenar de menor a mayor la lista de "Horas_inicio". Las otras listas también se han ordenado de menor a mayor para que puedan seguir correspondiéndose datos entre listas diferentes

e índices.

Se ha optado utilizar unas copias de dichas listas para proceder a ordenar los datos. De esta manera, se crearon las listas "Horas_inicio_ordenadas", "Frecuencias_ordenadas", "Fármacos_ordenados" y "Duración_Tratamiento_ordenado" y se han guardado en ellas los datos que había en las listas con el mismo nombre excluyendo "_ordenado".



4.1.- Screen_Pastillero, subbloques 20

Después se realizará el ordenamiento a través de un bucle y un procedimiento, y para el bucle se utilizará como condición de fin la longitud de una (todas tienen la misma longitud) de las cadenas anteriores. Se graba en la variable "longitud" el valor de dicha longitud de lista:



4.1.- Screen_Pastillero, subbloques 21

Y el bucle que se comentaba sería el siguiente, en el cual, para cada valor desde uno hasta (longitud de la lista Horas_inicio_ordenadas - 1) en incrementos de uno, se ejecutará el procedimiento "Ordenamiento_cadenas", que será comentado a continuación.



4.1.- Screen_Pastillero, subbloques 22

Procedimiento "Ordenamiento_cadenas"

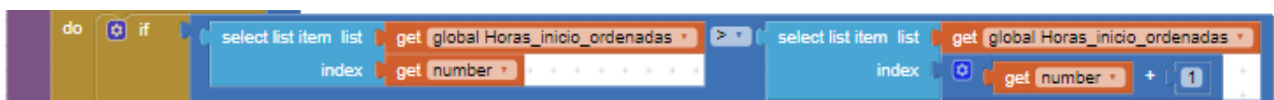
La razón por la cual se ha procedido a utilizar un procedimiento dentro del bucle anterior en vez de dos bucles anidados ha sido porque MIT App Inventor daba problemas con dichos bucles anidados.

El procedimiento comienza con la creación de un bucle for de las mismas características que el último que se mencionó del bloque 1, ya que el objetivo era crear dos bucles anidados de las mismas características para realizar el ordenamiento por el método de la burbuja. De esta manera, el procedimiento "Ordenamiento_cadenas" se ejecutará ("longitud" - 1) veces.



4.1.- Screen_Pastillero, subbloques 23

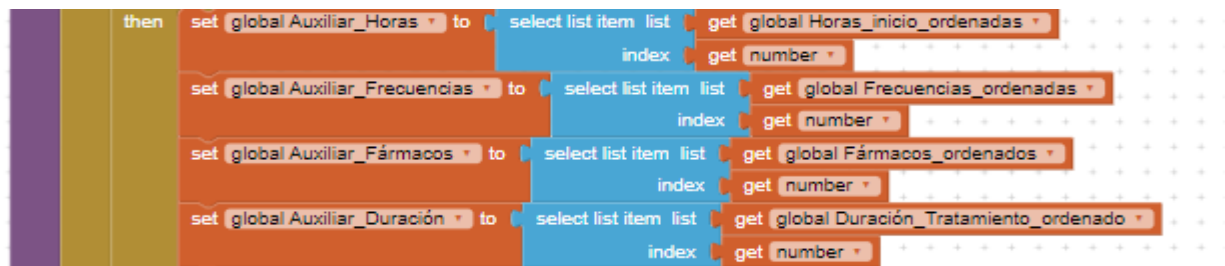
Dicho bucle "for", desde uno hasta la longitud de la lista Horas_inicio_ordenadas - 1, en incrementos de 1, comprobará si el valor de "Horas_inicio_ordenadas" de índice "number" de la lista Horas_inicio_ordenadas es mayor que el del siguiente índice de dicha lista:



4.1.- Screen_Pastillero, subbloques 24

Si así fuera entonces se grabarán en las variables creadas "Auxiliar_horas", "Auxiliar_Frecuencias", "Auxiliar_Fármacos" y "Auxiliar_Duración" el valor del elemento de índice "number" de "Horas_inicio_ordenadas", "Frecuencias_ordenadas",

“Fármacos_ordenados” y “Duración_Tratamiento_ordenado” respectivamente.



4.1.- Screen_Pastillero, subbloques 25

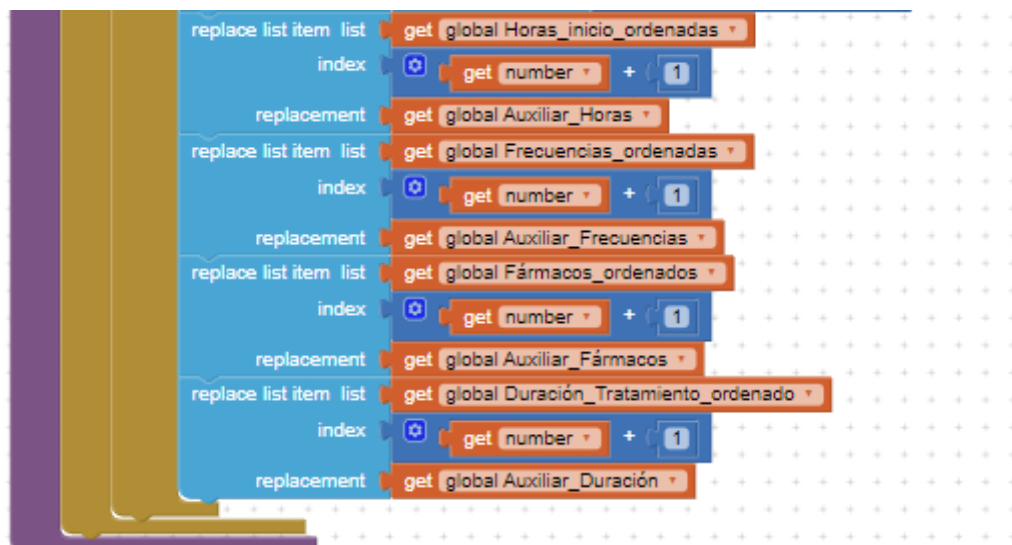
El objetivo es intercambiar los elementos de éstas últimas variables para ir pasando los elementos de mayor valor hacia el final de cada una de las listas. Para ello se necesitan unas variables auxiliares, que son las cuatro mencionadas anteriormente, para guardar los valores de los elementos mayores de la condición.

Siguiendo con la condición del “if” anterior y teniendo en cuenta el objetivo mencionado, se procede a intercambiar los valores de índice “number” con los de índice “number” + 1 dentro de cada una de las listas. De esta manera, de entre los dos valores comparados, el menor (en este caso sería el valor de índice “number” + 1) se guardaría en la posición “number” y el mayor en la posición “number” + 1:



4.1.- Screen_Pastillero, subbloques 26

Y, por último, ya que en la posición "number" se han guardado los valores de las posiciones "number" + 1 y será necesario recurrir a los valores que había en las posiciones "number" para guardarlos en las posiciones "number" + 1, se utiliza entonces los valores de las variables auxiliares comentadas anteriormente para realizar dicho intercambio de valores:

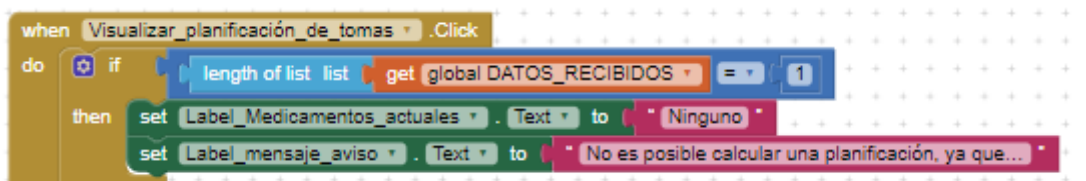


4.1.- *Screen_Pastillero*, subbloques 27

Bloque 2.- Para que se ejecute este bloque será necesario que el usuario pulse el botón "Planificación" de la pantalla que se está tratando *Screen_Pastillero*, y el objeto del mismo será, básicamente, proporcionar al usuario una planificación ordenada por fecha y hora de cada una de las tomas de fármacos que debe administrarse.

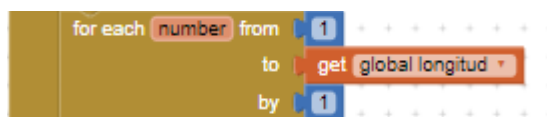
Comienza el bloque haciendo una comprobación de si hay o no fármacos guardados en la lista "DATOS_RECIBIDOS", calculando su longitud, ya que si ésta es igual a uno, sólo habrá como datos en dicha lista el año de nacimiento del usuario y por lo tanto no habrá fármacos guardados. En este caso se mostrará por

pantalla dos mensajes indicando que, por una parte, no hay ningún fármaco actualmente y por otra se indica la imposibilidad de calcular una planificación de tomas de medicamentos.



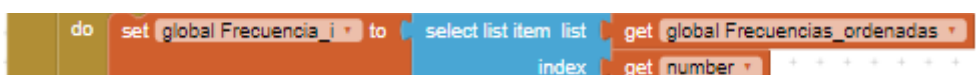
4.1.- Screen_Pastillero, subbloques 28

La siguiente parte del bloque está formada por un subbloque cuya misión es calcular las horas que quedan para empezar a tomar cada uno de los fármacos. Esto se realizará, básicamente, con un bucle "for" que recorre la longitud de cada una de las listas y utilizando dos procedimientos. Yendo por partes, el bucle mencionado se ejecutará, de uno en uno hasta llegar a la longitud de la lista "Horas_inicio_ordenadas" ("longitud") como se comentó anteriormente.



4.1.- Screen_Pastillero, subbloques 29

La lista "Frecuencias_ordenadas" contiene los valores de las frecuencias de administración de cada uno de los fármacos. De esta manera, la siguiente instrucción graba en la variable "Frecuencia_i" el valor del elemento de índice "number" de la lista "Frecuencias_ordenadas".

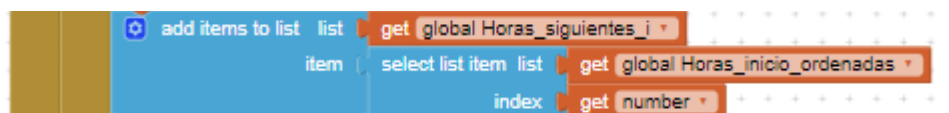


4.1.- Screen_Pastillero, subbloques 30

Al disponer del valor de la frecuencia de administración del fármaco de índice "number", el siguiente paso es conocer cuál es el hora de inicio de administración de dicho fármaco para poder calcular una

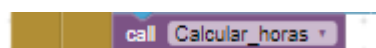
lista con las horas de administración del mismo.

Para ello, se graba en la lista creada "Horas_siguientes_i", primeramente, el valor del elemento de índice "number" de la lista "Horas_inicio_ordenadas":



4.1.- Screen_Pastillero, subbloques 31

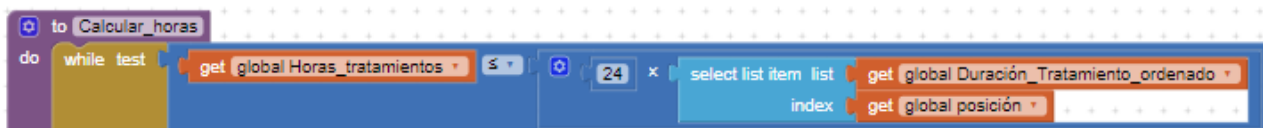
A continuación, dentro por supuesto del bucle "for", se llama a un procedimiento "Calcular_horas":



4.1.- Screen_Pastillero,
subbloques 32

Procedimiento "Calcular_horas"

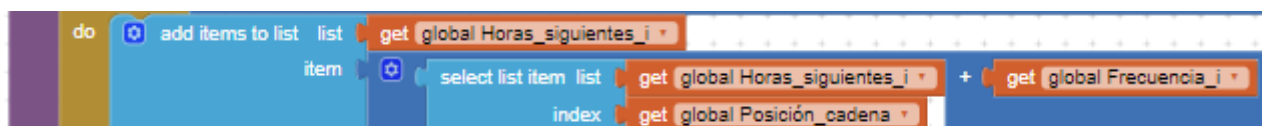
Teniendo en cuenta que los tratamientos tienen una duración determinada en días (si fuera muy largo podría darse un valor por ejemplo de siete días y volver a añadirse de nuevo ese mismo fármaco con otros siete días de duración), para calcular las horas de las tomas sucesivas se utilizará dicho valor de "Duración_Tratamiento_ordenado", donde está almacenada la duración en días del tratamiento para el fármaco i. Por lo tanto, se incluye un bucle "while" que observará cuándo se llega a esos días en concreto, quedando de la siguiente manera: Mientras "Horas_tratamientos" sea menor que $24 \times [\text{valor de la duración en días del índice adecuado de la lista "Duración_Tratamiento_ordenado"}]$, se ejecutarán las siguientes instrucciones (que llevarán como objetivo final al cálculo de las horas a las que deberán ser administrados los fármacos):



4.1.- Screen_Pastillero, subbloques 33

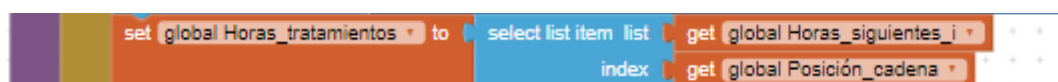
Como se ha comentado anteriormente, "Horas_siguientes_i" es una lista que tiene de momento un sólo valor, y es el correspondiente a la hora de inicio de un fármaco (el cual irá variando cuando se calculen todas las "Horas_siguientes_i", referentes al bucle "for" justo antes de comenzar el procedimiento.

Se irán añadiendo a la lista "Horas_siguientes" los siguientes elementos cada vez que se siga ejecutando "while": el valor que tiene el elemento de índice "Posición_cadena" de la propia lista "Horas_siguientes_i" más el valor de la frecuencia con la que debe tomarse dicho fármaco, "Frecuencia_i"). Por supuesto, "Posición_cadena" se irá incrementando conforme avance el procedimiento como se verá más tarde.



4.1.- Screen_Pastillero, subbloques 34

Siguiendo dentro del bucle "while" como se ve en los colores de la siguiente imagen, para controlar hasta cuándo se ejecuta dicho bucle, se irá grabando en la variable "Horas_tratamientos" el valor que tenga el elemento de índice "Posición_cadena" de la lista "Horas_siguientes_i".



4.1.- Screen_Pastillero, subbloques 35

La última instrucción del bucle consiste en incrementar en uno la variable "Posición_cadena" con el objetivo de poder ir

añadiendo los valores adecuados a la lista "Horas_siguientes_i".

(Ejemplo: los valores con cada ejecución del bucle "while" serían:

```
Horas_siguientes_i = [Horas_inicio_ordenadasfármaco_i];  
Horas_siguientes_i = [Horas_inicio_ordenadasfármaco_i, Horas_inicio_ordenadasfármaco_i +  
Frecuencia_i];  
Horas_siguientes_i = [Horas_inicio_ordenadasfármaco_i, Horas_inicio_ordenadasfármaco_i +  
Frecuencia_i, (Horas_inicio_ordenadasfármaco_i + Frecuencia_i) + Frecuencia_i]
```

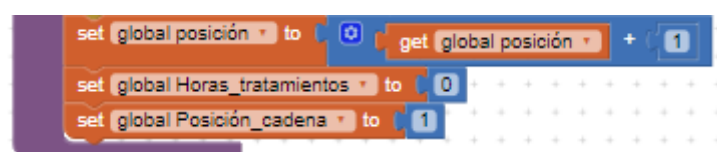
Y así sucesivamente.



4.1.- Screen_Pastillero, subbloques 36

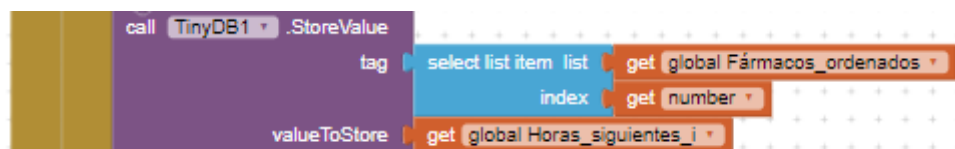
Fuera del bucle "while" ya, pero continuando con el procedimiento, y finalizando con él, las últimas instrucciones tienen los siguientes objetivos:

- A la variable "posición" se le suma 1. Esto se debe hacer para escoger el elemento de índice siguiente de la lista "Duración_tratamiento_ordenado" para poder ejecutar correctamente la condición de "while" correspondiente al fármaco adecuado (recuérdese que este procedimiento estaba dentro de un bucle "for").
- A "Horas_tratamientos" se le asigna el valor 0. Es necesario reiniciar este valor al salir del procedimiento para que, al entrar de nuevo desde el bucle, se pueda comenzar de nuevo con los datos de otro fármaco.
- El valor "Posición_cadena" se iguala a 1. La razón es exactamente la misma que la de la instrucción anterior.



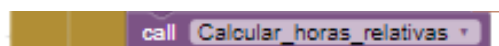
4.1.- Screen_Pastillero, subbloques 37

Continuación primera del Bloque 2.- Tras ejecutarse el anterior procedimiento se regresa a la siguiente instrucción de la llamada del mismo en el bloque 2. Se trata de grabar la lista "Horas_siguientes_i" que se ha calculado en el procedimiento anterior en una base de datos, TinyDB1, con la siguiente etiqueta: se utilizará el valor del elemento de índice "number" de la lista "Fármacos_ordenados" (en este caso corresponderá al nombre de uno de los fármacos; Recuérdese que "number" es la variable local del bucle "for" e irá incrementándose con cada pasada y señalando la posición donde se encuentra el siguiente fármaco a considerar).



4.1.- Screen_Pastillero, subbloques 38

Téngase en cuenta que dichos valores de la lista "Horas_siguientes_i" sólo muestran las horas de la siguiente manera: la primera hora será la que introdujo el usuario tal cual, de manera que no representa cuándo debe el usuario tomar el medicamento al que hace referencia. Para ello se empleará otro procedimiento "Calcular_horas_relativas", el cual se expondrá a continuación.



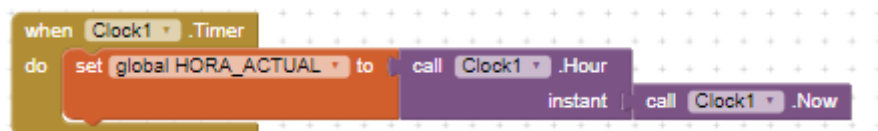
4.1.- Screen_Pastillero, subbloques 39

Procedimiento "Calcular_horas_relativas"

El objetivo de este procedimiento es transformar las horas de las tomas dadas por la lista "Horas_siguientes_i" en horas relativas, es decir, cada valor representará cuántas horas de diferencia hay

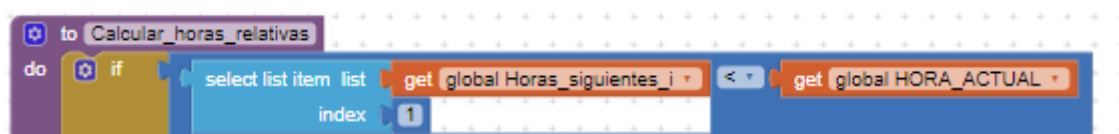
con respecto al momento actual. Más adelante esto se transformará en otra lista que indicará dentro cuántas horas se debe administrar cada fármaco.

Primeramente, a la variable creada al principio del procedimiento (no se han estado escribiendo éstas porque ya están reflejadas en la programación completa de la pantalla y no hacer más engorrosa la lectura de la memoria) "HORA_ACTUAL", se le asigna el valor de la hora actual, valga la redundancia, mediante unos procedimientos existentes en APP Inventor 2: el primero más a la izquierda devuelve las horas de la fecha indicada, y la segunda le indica que la fecha es ahora.



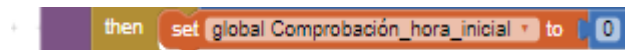
4.1.- Screen_Pastillero, subbloques 40

La primera instrucción comenzará con un "if" para comprobar si la "HORA_ACTUAL" es mayor que el primer valor de la lista "Horas_siguientes". Esto es así porque si fuera menor sería muy sencillo el cálculo de la nueva lista "Horas_siguientes_i", se trataría de una simple resta de horas para sacar las relativas (se verá un poco adelante), pero si es mayor habrá números negativos y esto llevaría a dificultar más tarde la obtención de las horas reales, por que lo hay que diferenciar las instrucciones a llevar a cabo mediante dicha condición "if".



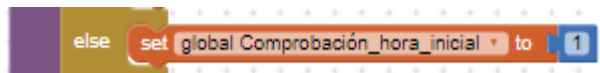
4.1.- Screen_Pastillero, subbloques 41

Si resultara que el valor del elemento de índice 1 de la lista "Horas_siguientes_i" fuera menos que el valor de la "HORA_ACTUAL", se adjudicará a la variable "Comprobación_hora_inicial" el valor 0.



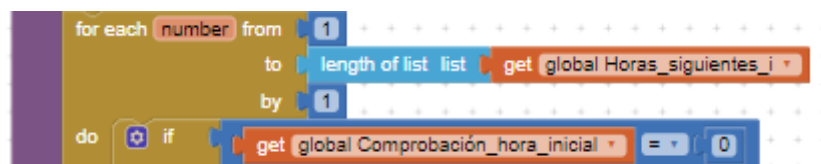
4.1.- Screen_Pastillero, subbloques 42

Sino fuera el caso, entonces se le adjudicará el valor 1.



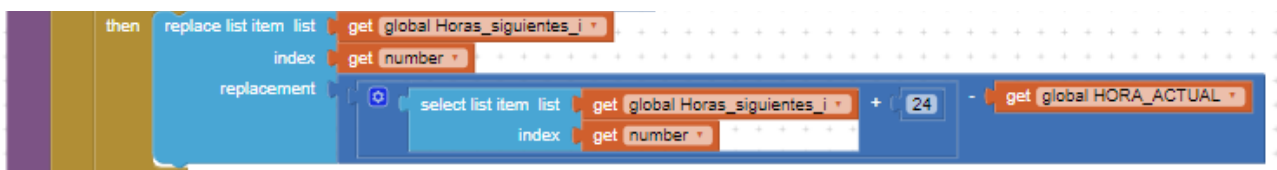
4.1.- Screen_Pastillero, subbloques 43

Esto se ha hecho así para poder aplicar una misma función a la transformación de la lista "Horas_siguientes_i", aumentando así la simplicidad de la implementación. Entonces se va a hacer un barrido de todos los elementos de la lista mediante un bucle "for" para ir grabando los nuevos valores relativos. Claro está, desde el índice 1, incrementando en 1 con cada pasada y hasta llegar a la longitud de la lista "Horas_siguientes_i", se empieza realizando la comprobación mencionada anteriormente: si la variable "Comprobación_hora_inicial" vale 0 (equivaldría a que el valor del elemento de índice "number" de la lista "Horas_siguientes_i" < "HORA_ACTUAL"):



4.1.- Screen_Pastillero, subbloques 44

Se ejecutarían las siguientes instrucciones:



4.1.- Screen_Pastillero, subbloques 45

Cuya función es la de sustituir el valor del elemento de índice

"number" de la lista "Horas_siguientes_i" por el resultado de sumar dicho valor a 24 y restarle el valor de "HORA_ACTUAL".

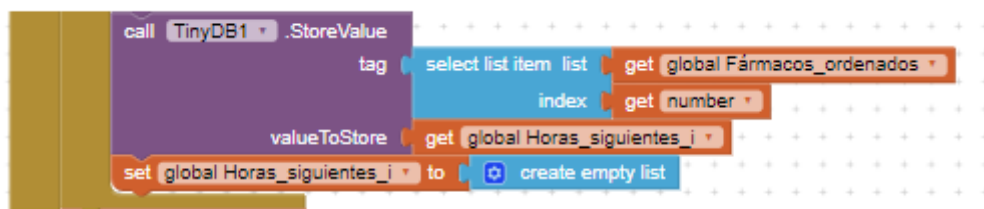
En el caso en que la variable "Comprobación_hora_inicial" valiera 1 (el valor del elemento de índice "number" de la lista "Horas_siguientes_i" > "HORA_ACTUAL"), se ejecutaría, siguiendo dentro del bucle "for" y siendo el último subbloque del procedimiento:



4.1.- Screen_Pastillero, subbloques 46

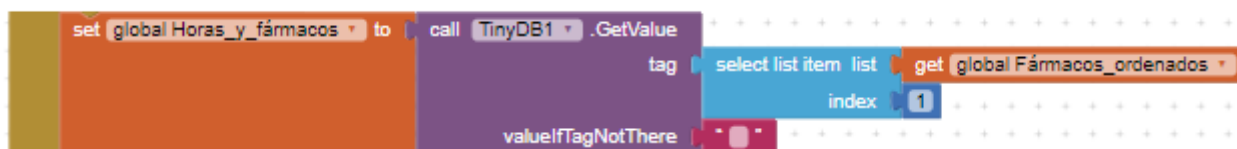
Con esto lo que se consigue es, simplemente, sustituir, en cada pasada, el valor del elemento del índice "number" por dicho valor menos el valor de "HORA_ACTUAL".

Continuación segunda del Bloque 2.- Al finalizar el procedimiento anterior se vuelve a la instrucción siguiente a la llamada al mismo del bloque 2, quedando solamente dos instrucciones para finalizar el bucle "for" dentro del cual se llamaron a los dos procedimientos. La primera de ellas tiene como funcionalidad grabar los nuevos datos "Horas_siguientes_i" calculados en el procedimiento "Calcular_horas_relativas", y se grabarán en TinyDB1 , concretamente en la etiqueta como nombre el fármaco correspondiente al elemento de índice "number" de la lista "Fármacos_ordenados". La segunda instrucción consiste en borrar la lista "Horas_siguientes_i" para poder ser utilizada en la siguiente pasada del bucle "for".



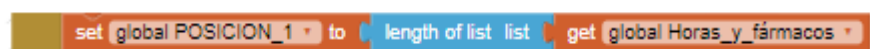
4.1.- Screen_Pastillero, subbloques 47

Una vez fuera del bucle "for" se copia en la variable "Horas_y_fármacos" (esta lista será la que se utilizará para construir la definitiva "Fármacos_y_horas") el valor del contenido de la TinyDB1 de etiqueta el elemento de índice 1 de la lista "Fármacos_ordenados" (serán las horas relativas del fármaco en cuestión).



4.1.- Screen_Pastillero, subbloques 48

A continuación se asigna a la variable "POSICIÓN_1" el valor de la longitud de la nueva lista de valores "Horas_y_fármacos" (longitud de la lista de horas relativas del fármaco ordenado de índice 1). Esto se ha realizado así porque será necesario saber dónde empieza y acaba cada lista de horas relativas de cada fármaco al colocarlas todas en una misma lista como se pretende.

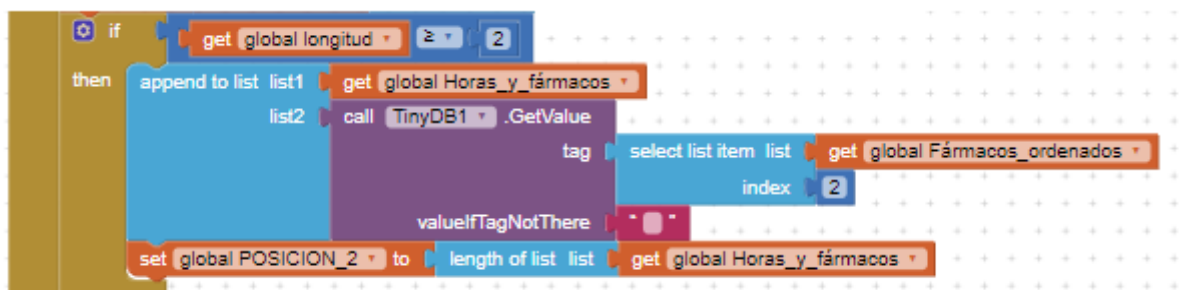


4.1.- Screen_Pastillero, subbloques 49

Vendrán una serie de comprobaciones mediante "if", cuyo objetivo será, por una parte comprobar cuántos fármacos hay y por otra, establecer unos valores que marcarán los límites entre fármacos en la lista donde se guardarán todas las horas relativas juntas.

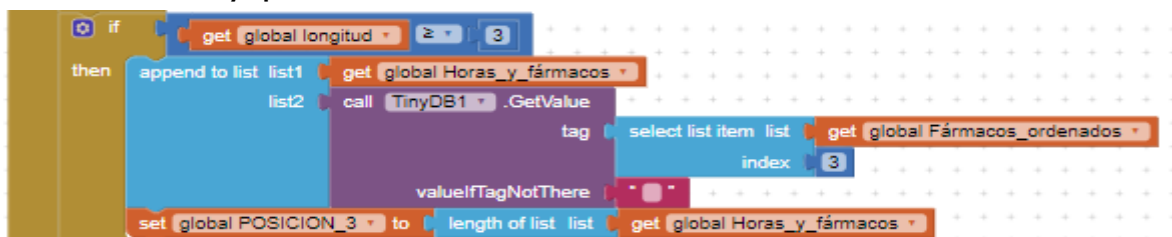
La condición "longitud" mayor o igual que 2 significaría que habría dos o más fármacos actuales guardados por el usuario, de

manera que si así fuera se procedería añadiendo a la lista "Horas_y_fármacos" otra lista: la obtenida a partir de los valores de la TinyDB1 de etiqueta el segundo elemento de la lista "Fármacos_ordenados". De esta manera la lista "Horas_y_fármacos" tendría todas las horas relativas a los dos primeros fármacos. Y acto seguido se guarda la longitud de dichas horas en la variable "POSICIÓN_2". (Recuérdese que el valor de "POSICIÓN_1" correspondía a la longitud de las horas relativas del fármaco 1.



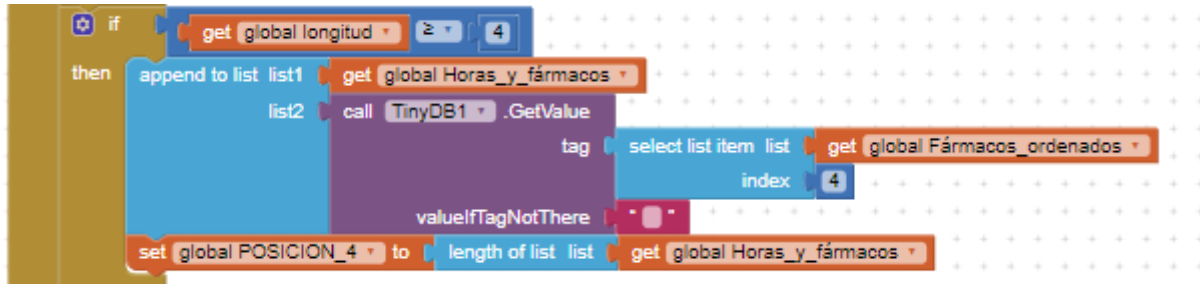
4.1.- Screen_Pastillero, subbloques 50

Se hará lo mismo para el caso que hubiera tres fármacos. Se escribe la condición de que "longitud" sea mayor o igual a tres y se añade a la lista "Horas_y_fármacos" las horas relativas calculadas para el fármaco que ocupe la tercera posición en la lista "Fármacos_ordenados. También se le da a la variable "POSICIÓN_3" el valor de la nueva longitud de "Horas_y_fármacos". (Es necesario que en las condiciones se utilice el símbolo mayor o igual para que se ejecuten todas las condiciones posibles con el objeto de asignar valores a las variables "POSICIÓN_i" y poder marcar unos límites como se verá en breve).



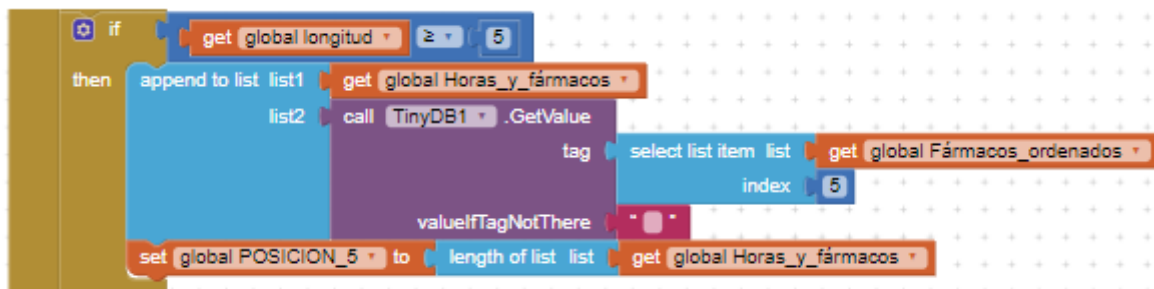
4.1.- Screen_Pastillero, subbloques 51

De la misma manera, para cuatro fármacos:



4.1.- Screen_Pastillero, subbloques 52

Y, como el límite de fármacos se ha puesto a cinco, el último bloque "if" de este tipo es:



4.1.- Screen_Pastillero, subbloques 53

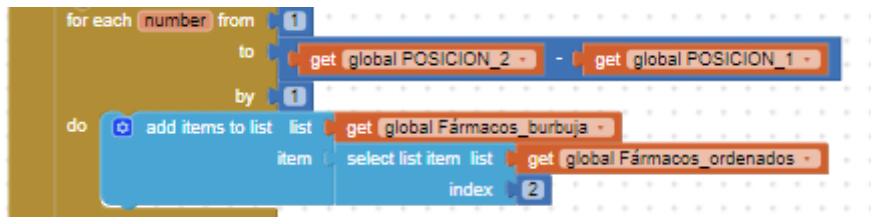
La estrategia que se ha seguido ahora es la de hacer listas con los nombres de los fármacos de igual longitud que las de las horas relativas de cada uno de ellos para unirlos. Dicha longitud de cada grupo queda determinada por las variables mencionadas anteriormente ("POSICIÓN_1", ..., "POSICIÓN_5"). Dichas lista, llamada "Fármacos_burbuja" no será más que la repetición y unión de los datos almacenados en "Fármacos_ordenados"



4.1.- Screen_Pastillero, subbloques 54

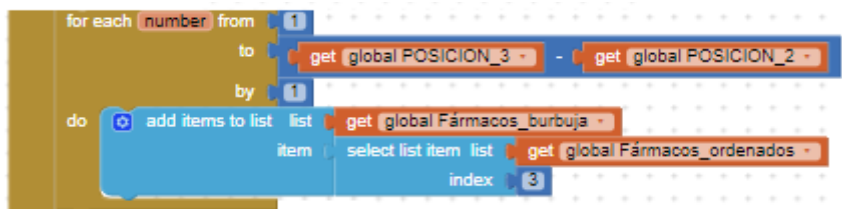
Con este bloque se obtiene una lista con el mismo nombre de fármaco repetido "POSICIÓN_1" veces, la cual se unirá a la lista de horas relativas correspondiente a este fármaco.

Haciendo lo mismo para el segundo fármaco, y teniendo en cuenta que el número de elementos de su lista de horas relativas correspondiente es de "POSICIÓN_2" - "POSICIÓN_1":



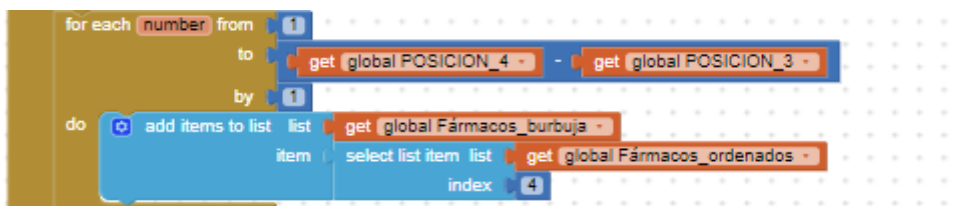
4.1.- Screen_Pastillero, subbloques 55

De la misma manera, se añade a la lista "Fármacos_burbuja" (si hubiera tres fármacos), una lista formada exclusivamente por el nombre del tercer fármaco una cantidad de veces "POSICIÓN_3" - "POSICIÓN_2":



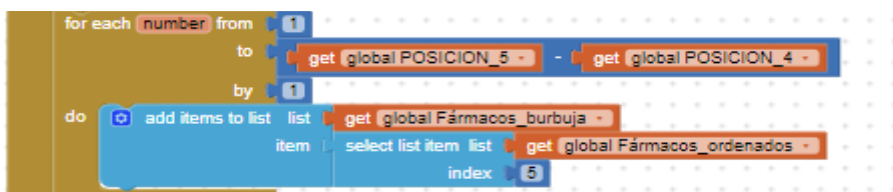
4.1.- Screen_Pastillero, subbloques 56

Téngase en cuenta que, si hubiera menos fármacos, este bucle y los siguientes dos (hasta cinco fármacos, el límite que permite la app) no se ejecutaría, ya que "POSICIÓN_i" valdría 1 por no haber recibido otro valor y por tanto el bucle no comenzaría. Para el siguiente caso:



4.1.- Screen_Pastillero, subbloques 57

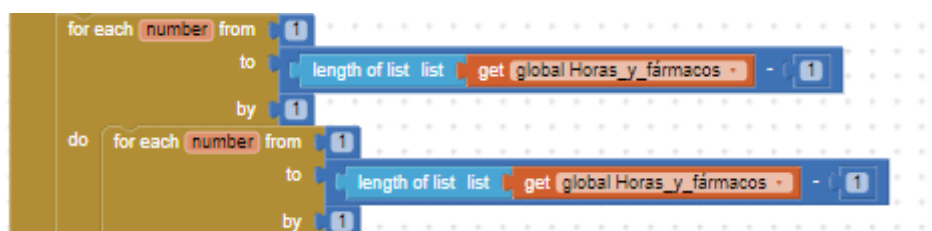
Y, finalmente, para el último (por supuesto en caso que existiera):



4.1.- Screen_Pastillero, subbloques 58

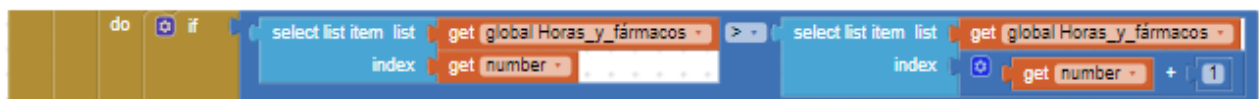
Ahora se tienen dos listas importantes: "Horas_y_fármacos" (con las horas relativas para las tomas de los fármacos, primero todas para un fármaco, después todas para el siguiente, etc.) y "Fármacos_burbuja" (con los nombres de los fármacos repetidos tantas veces como horas relativas hay en la lista anterior, primero un fármaco, después el siguiente, etc.).

El objetivo ahora es ordenar todas las horas relativas de todos los fármacos de menor a mayor para poder ofrecer un tratamiento ordenado, pero para ello es necesario también ordenar la lista de todos los fármacos exactamente de la misma forma para mantener la correspondencia entre ambos datos de forma correcta. Por lo tanto, se procede a realizar dicho ordenamiento y se aplicará el método de la burbuja (dos bucles "for" para recorrer la longitud de la lista "Horas_y_fármacos"):



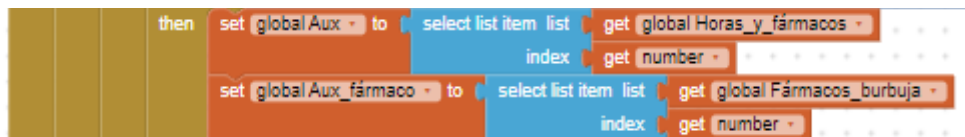
4.1.- Screen_Pastillero, subbloques 59

Y dentro de estos bucles se comprueba primero si el elemento "number" de la lista "Horas_y_fármacos" es mayor que el siguiente:



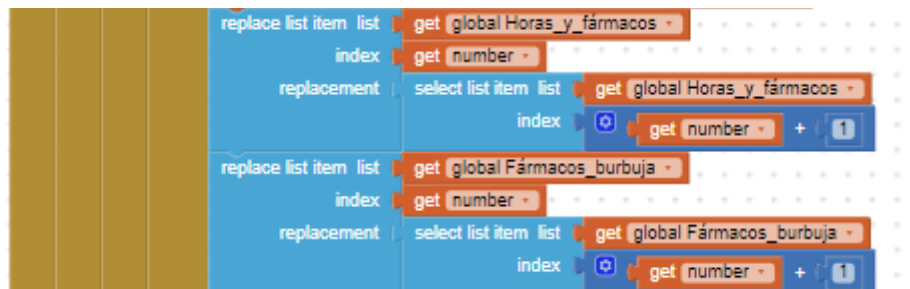
4.1.- Screen_Pastillero, subbloques 60

Si así fuera, se utilizan dos variables auxiliares creadas al principio del bloque ("Aux" y "Aux_fármaco") para guardar tanto el elemento "number" de la lista "Horas_y_fármacos" como el de la lista "Fármacos_burbuja":



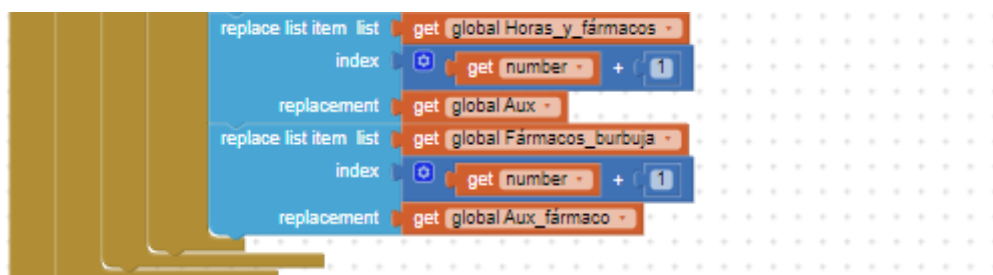
4.1.- Screen_Pastillero, subbloques 61

El siguiente paso es reemplazar el elemento "number" de las listas que se están tratando por el siguiente:



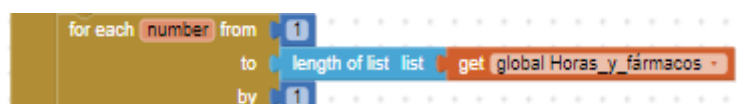
4.1.- Screen_Pastillero, subbloques 62

Y ahora resta por reemplazar el elemento "number"+1 de dichas listas por el anterior "number", utilizando para ello las variables auxiliares "Aux" y "Aux_fármaco" que tienen los valores originales de los elementos de índice "number" de las listas "Horas_y_fármacos" y "Fármacos_burbuja":



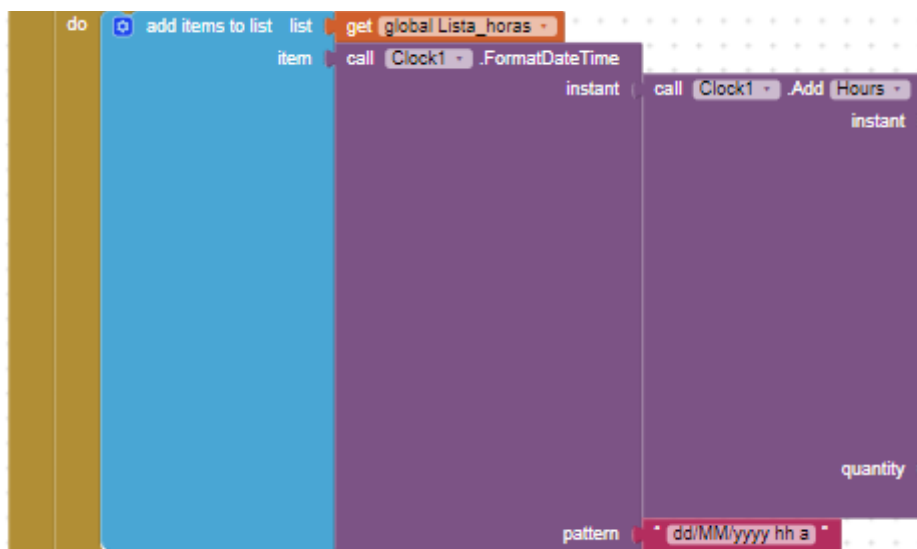
4.1.- Screen_Pastillero, subbloques 63

Ahora se tienen las dos listas correlacionadas y ordenadas de menor a mayor (esto último, para "Horas_y_fármacos", claro está), quedando pasar las horas relativas a fechas absolutas. Para ello se utilizará un bucle "for" para ir modificando cada una de las dichas horas relativas, desde uno y con incrementos de uno hasta la longitud de "Horas_y_fármacos".



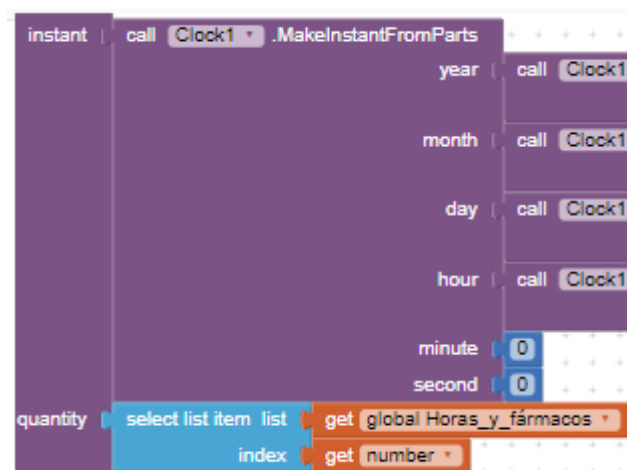
4.1.- Screen_Pastillero, subbloques 64

Se utilizará una nueva lista para ir añadiendo en ella dichas conversiones, que serán elementos obtenidos de la función "Clock1.FormatDateTime", cuyo formato de salida será tipo "dd/MM/yyyy hh" y con un argumento que se pasa a través de "instant". Dicho argumento será otra llamada a la función Clock1, en este caso "Clock1.Add(Hours)", cuya función será sumar las horas relativas a la actual para determinar el momento correcto de las tomas.



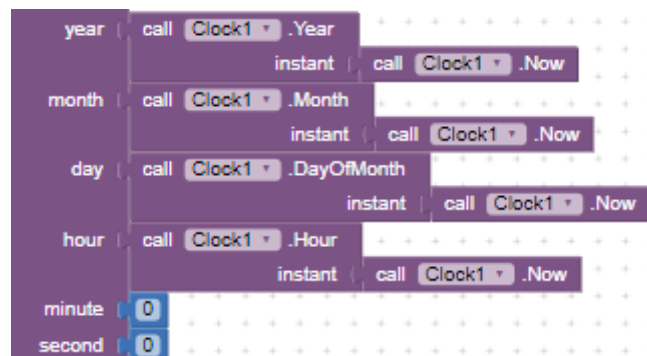
4.1.- Screen_Pastillero, subbloques 65

Los argumentos que recibe esta última función son "instant" y "quantity", éste último indica la cantidad que se va a añadir a "instant". La cantidad será, claro está, las horas relativas, es decir, las horas que quedan desde el instante marcado, para así convertirse en horas reales, escogiendo de la lista "Horas_y_fármacos" una a una las horas (recuérdese que se está dentro de un bucle "for"). Por otra parte, el instante va a venir marcado por el momento actual, utilizando antes para ello la función "Clock1.MakeInstantFromPart" y con los argumentos de datos de la fecha (año, mes, día, hora, minuto, segundo).



4.1.- Screen_Pastillero, subbloques 66

A dos de estos último argumentos (minuto y segundo) se les dará el valor 0 y al resto se les pasará otra función "Clock1.Yearinstant", "Clock1.Monthinstant", "Clock1.DayOfMonthinstant" y "Clock1.Hourinstant" con objeto de calcular los datos a los que hacen referencia, funciones a las que se les pasará otras que indicarán el momento actual (Clock1.Now) para ofrecerles dichos datos.

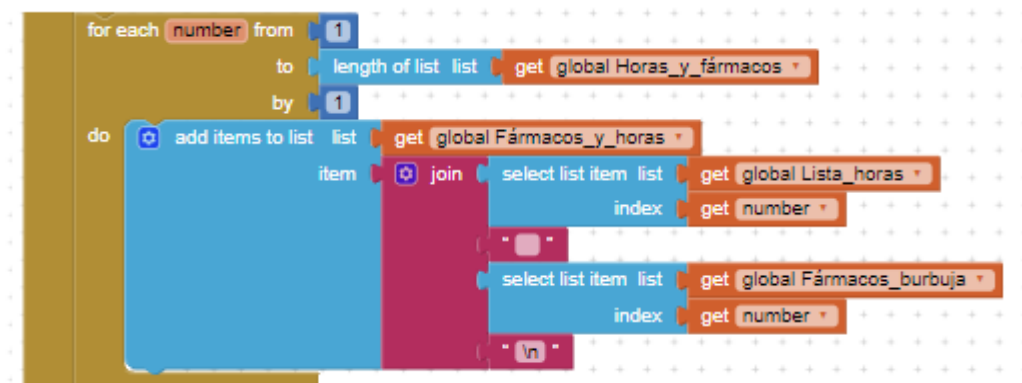


4.1.- Screen_Pastillero, subbloques 67

De esta manera, se tendría ahora en la lista "Lista_horas" todas las horas reales de las tomas ordenadas de menor a mayor.

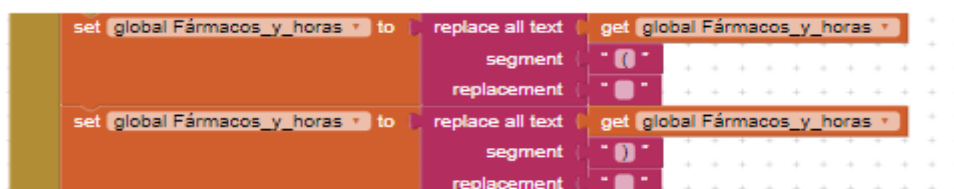
Se ha utilizado una nueva lista "Fármacos_y_horas" para unir las horas reales de cada toma de fármaco con este último y mostrar los datos en filas, separadas por hora y fármaco. Para ello se crea

un bucle "for" de longitud "Horas_y_fármacos" (esta longitud es la cantidad de elementos de horas para todos los fármacos) y se añade a la nueva lista creada "Fármacos_y_horas" en cada pasada del bucle: un elemento de "Lista_horas", seguido de un espacio en blanco, su correspondiente fármaco obtenido del elemento del mismo índice de "Fármacos_burbuja" seguido finalmente por "\n" para cambiar de línea.



4.1.- Screen_Pastillero, subbloques 68

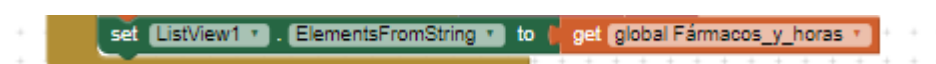
Sólo falta en este punto quitar los dos paréntesis, uno al principio y otro al final, de la lista "Fármacos_y_horas", y para ello se utiliza las siguientes instrucciones:



4.1.- Screen_Pastillero, subbloques 69

Donde solamente se reemplazan ambos paréntesis por espacios en blanco.

Para finalizar el Bloque 2 se utiliza una "ListView1" para mostrar todos los datos almacenados en la lista final "Fármacos_y_horas"



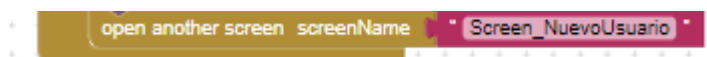
4.1.- Screen_Pastillero, subbloques 70

Bloque 3.- Este bloque tiene como función ir a la pantalla *Screen_NuevoUsuario*. Al pulsar el botón “Añadir”, se realizan las siguientes operaciones: Primeramente, la variable “comprobación” se pone a valor 1 para después grabar dicho valor en la TinyDB1 de etiqueta “verificación”. Como se mencionó con anterioridad, existe un bucle en el Bloque 1 donde se comprueba dicho valor cada vez que se regresa a esta pantalla para guardar los datos de una manera u otra.



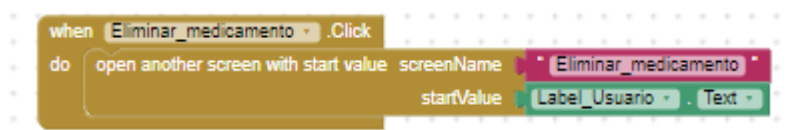
4.1.- *Screen_Pastillero*, subbloques 71

Y como última instrucción se procede a abrir la ventana *Screen_NuevoUsuario*.



4.1.- *Screen_Pastillero*, subbloques 72

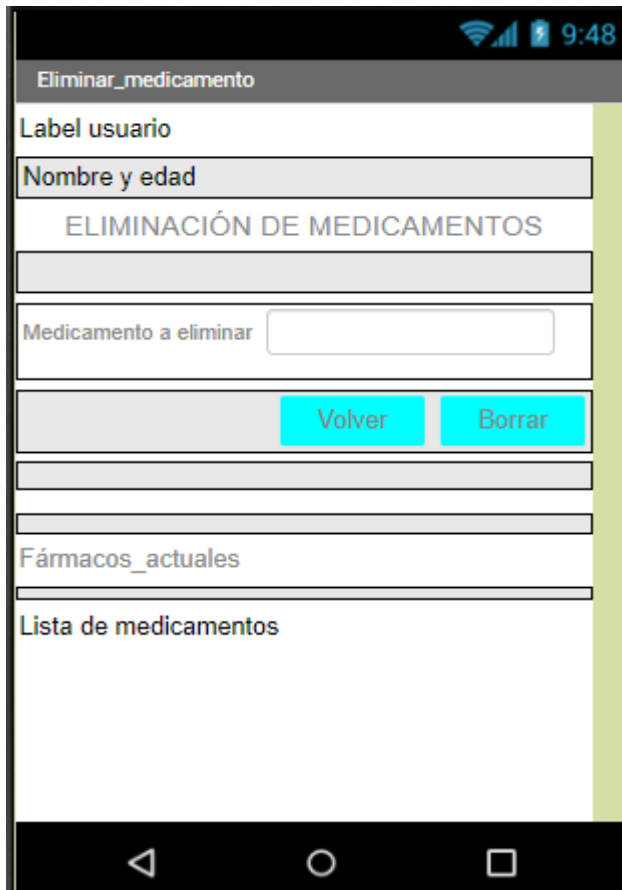
Bloque 4.- Este bloque tiene como función abrir la ventana *Eliminar_medicamento*. Cuando se pulse el botón “Eliminar” se abrirá dicha ventana a la vez que se le pasará el valor “Label_Usuario”, que será necesario para poder trabajar con la base de datos TinyDB2 de dicha etiqueta (donde se encuentran almacenados los “DATOS_RECIBIDOS”).



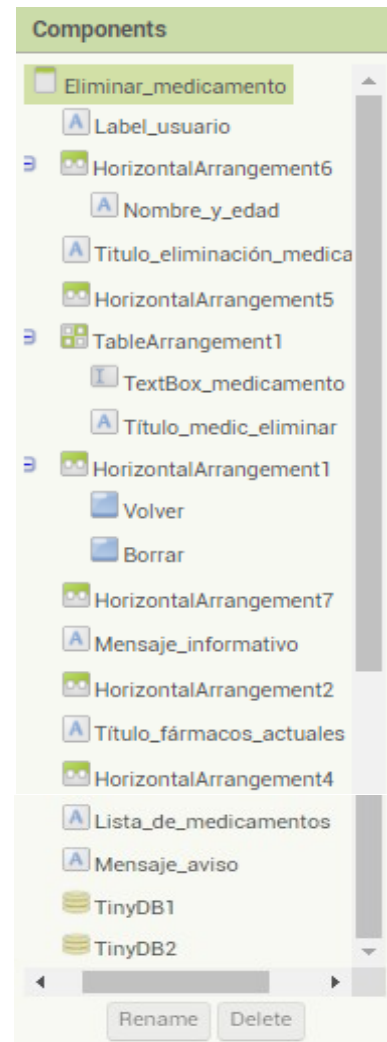
4.1.- *Screen_Pastillero*, subbloques 73

Eliminar_medicamento

A esta pantalla sólo se puede acceder a través de *Screen_Pastillero* al pulsar el botón “Eliminar” y su diseño es:



4.1.- *Eliminar_medicamento*, imagen 4

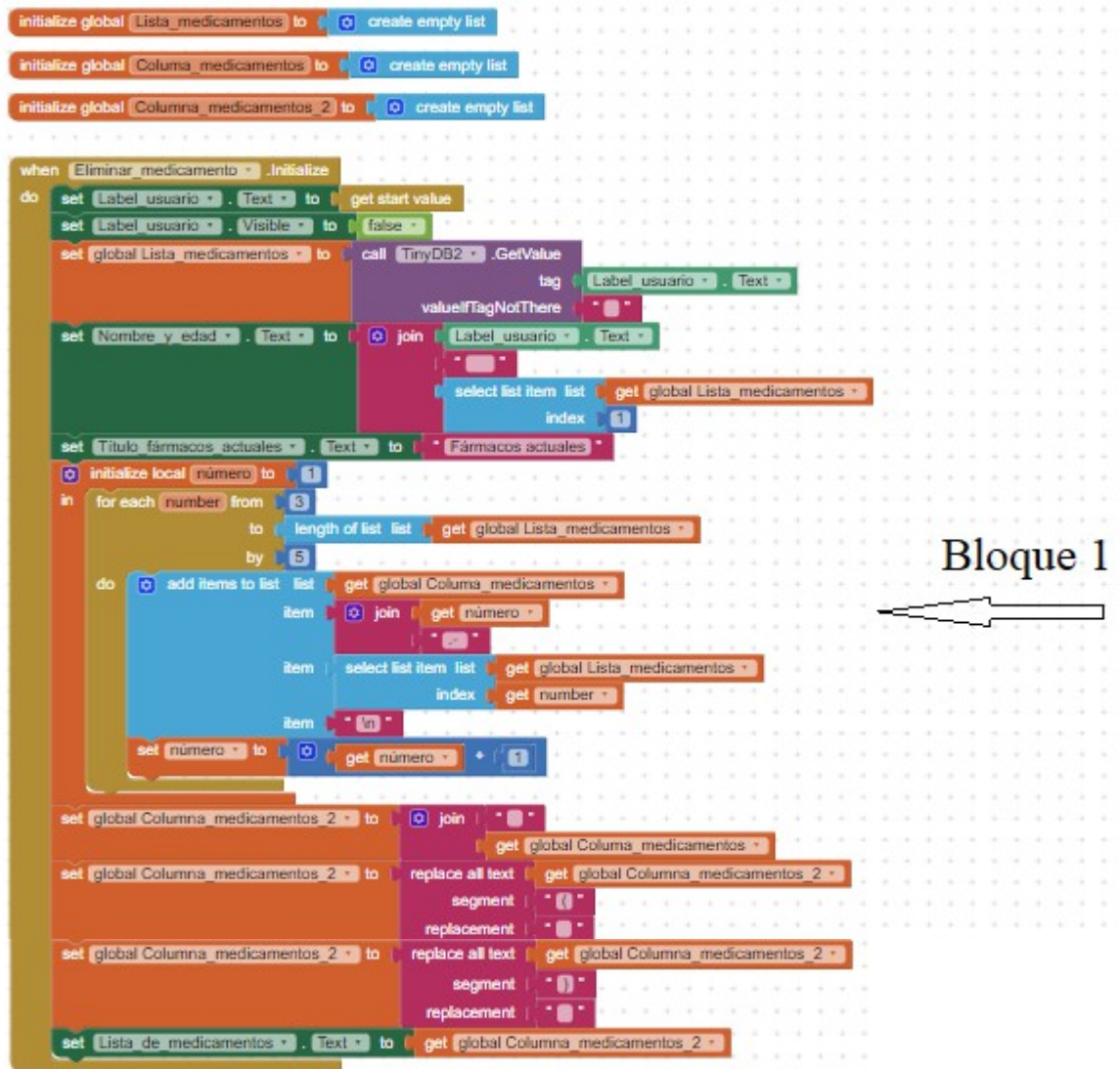


4.1.- *Eliminar_medicamento*, imagen 5

Básicamente está compuesta por ocho etiquetas, una de ellas se ha escogido que no sea visible, ya que su función es la de pasar datos de otra pantalla a otra, tres de ellas son títulos fijos que aparecen en la pantalla y las otras cuatro servirán para mostrar información en la pantalla relativa al usuario y sus datos. Asimismo, hay también una caja de texto (que será donde el usuario pueda escribir el medicamento que deseara borrar), dos botones (para

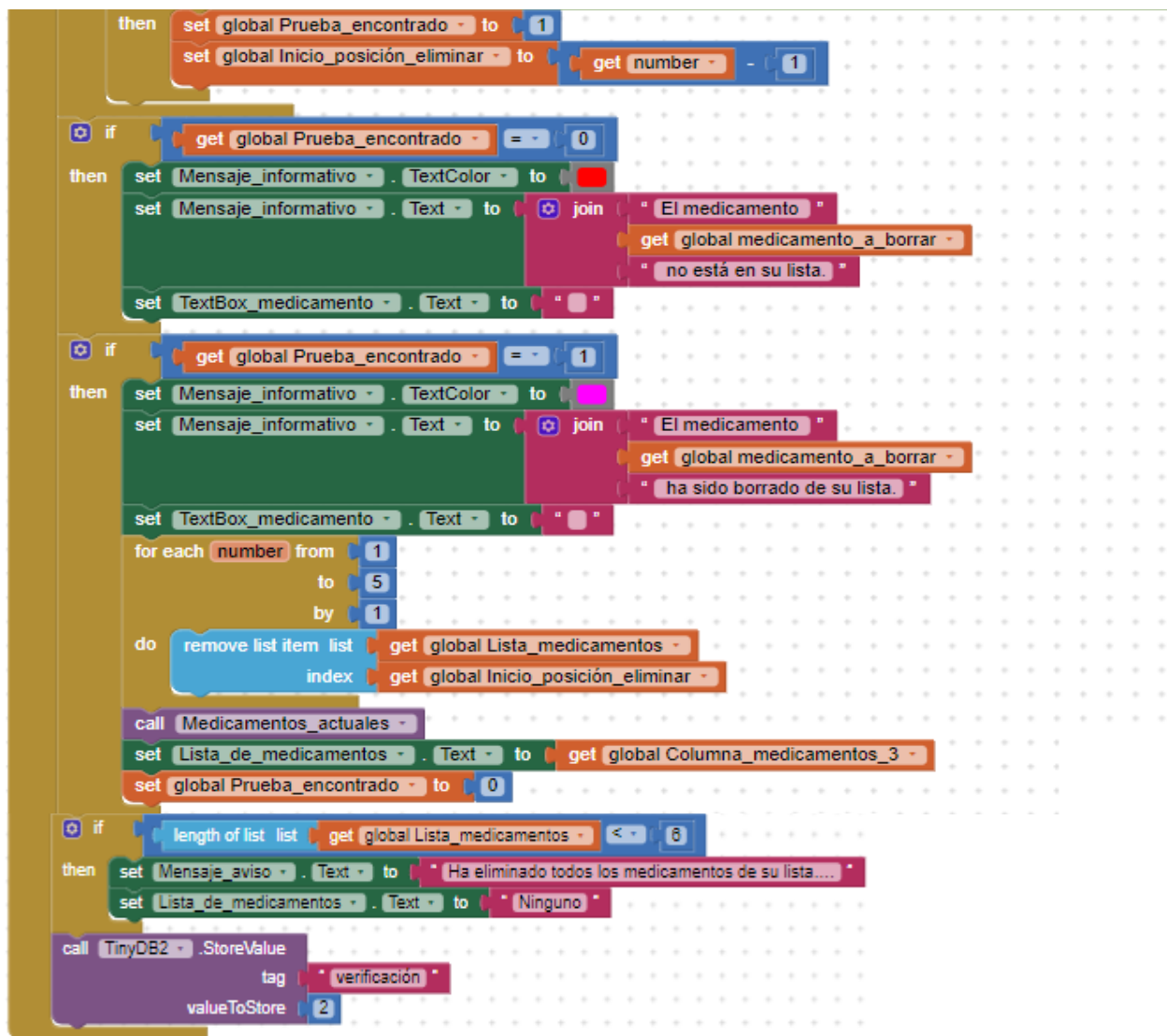
borrar el medicamento escrito en la caja de texto o para regresar a la pantalla anterior *Screen_Pastillero*) y dos TinyDB (TinyDB1 y TinyDB2).

Las instrucciones de los bloques son, básicamente, las siguientes:

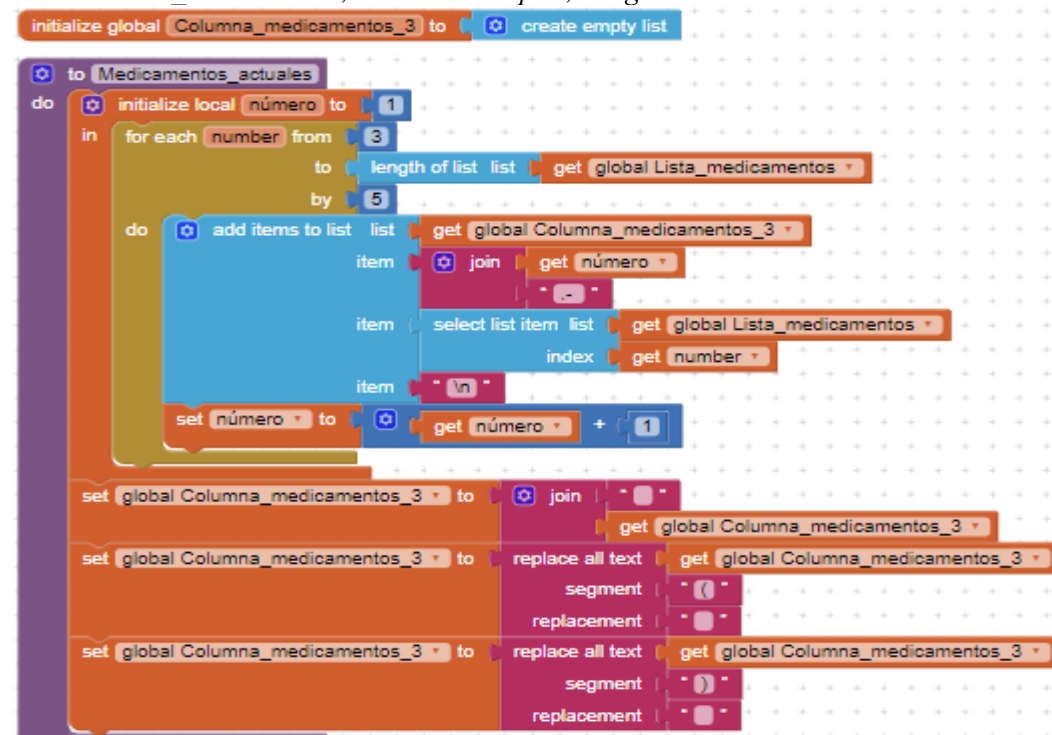


4.1.- *Eliminar_medicamento*, diseño de bloques, imagen 1

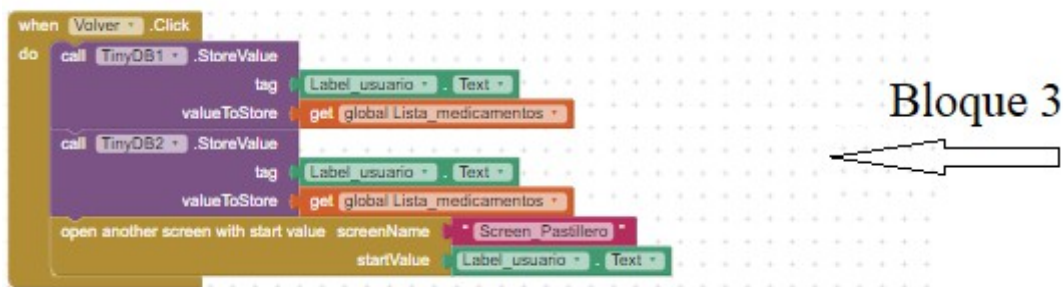




4.1.- Eliminar medicamento, diseño de bloques, imagen 2

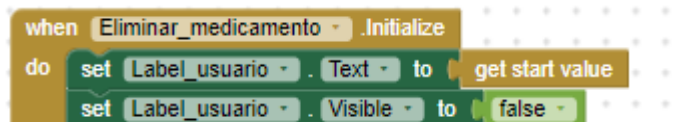


4.1.- Eliminar medicamento, diseño de bloques, imagen 3



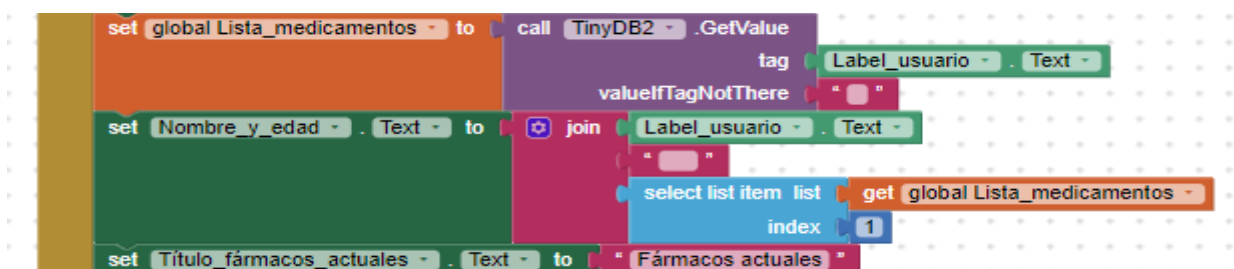
4.1.- Eliminar_medicamento, diseño de bloques, imagen 4

Bloque 1.- Al pulsar el botón “Eliminar” de *Screen_Pastillero*, se abrirá esta pantalla, a la vez que recibe un valor de aquella. Este valor era “Label_Usuario”, el cual se guardará en la etiqueta de esta pantalla “Label_usuario”. Asimismo, se desactivará su visibilidad:



4.1.- Eliminar_medicamento, subbloque 1

En una variable creada de tipo lista, “Lista_medicamentos” se guarda el valor obtenido de la base de datos TinyDB2 de etiqueta “Label_usuario” (el correspondiente “DATOS_RECIBIDOS” de *Screen_Pastillero*). A la vez, a la etiqueta “Nombre_y_edad” se le pasan los datos del nombre del usuario (guardado en la etiqueta “Label_usuario”), un espacio en blanco y el año de nacimiento (el primer elemento de “Lista_medicamentos”). Asimismo, a la etiqueta “Título_fármacos_actuales” se le pasa la cadena “Fármacos actuales” para que permanezca siempre en pantalla.

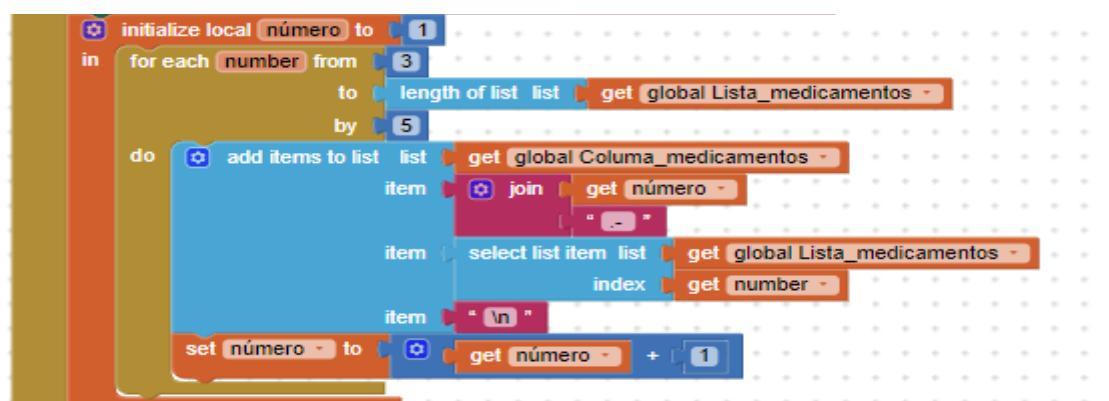


4.1.- Eliminar_medicamento, subbloque 2

El siguiente subbloque con sus subbloques tiene como objetivo

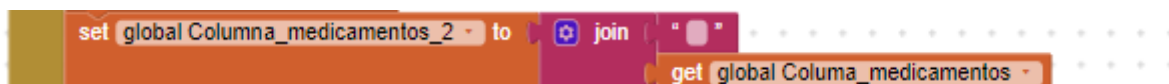
grabar en una variable tipo lista "Columna_medicamentos" todos los nombres de los medicamentos actuales numerados y en filas distintas. Para ello se crea una variable local "número" que se irá incrementando de uno en uno mientras se cumpla el bucle interior "for" para enumerar los medicamentos. Dentro de dicho bucle "for" se escogen como condiciones de inicio e incrementos los índices de "Lista_medicamentos" donde se encuentran los nombres de los fármacos (3 y 5 respectivamente) y como condición de fin, la longitud de "Lista_medicamentos".

Mientras se ejecute el bucle "for" se irán añadiendo a la lista "Columna_medicamentos" los siguientes elementos: la variable local "número", que irá contando las veces que se ha ejecutado el bucle "for" para así enumerar todos los fármacos actuales, los símbolos ".-" , el elemento de índice "number" de "Lista_medicamentos" (es donde se encuentran los nombres de éstos) y "\n" para cambiar de línea entre cada medicamento. La siguiente instrucción dentro del bucle será actualizar el valor de "número" para seguir enumerando los fármacos si aún no ha finalizado el bucle.



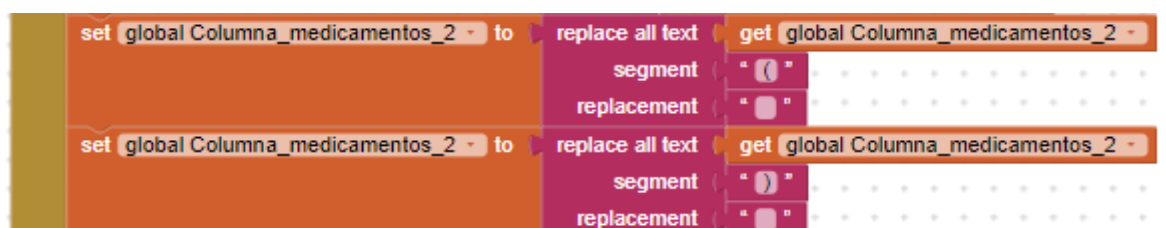
4.1.- Eliminar_medicamento, subbloque 3

Las filas de los fármacos almacenadas así en "Columna_medicamentos" tienen dos paréntesis y quiere separar por un espacio de la pantalla para que no esté pegada a ésta. Para conseguir esto último se le unirá, a la lista "Columna_medicamentos_2" un espacio en blanco " " y la lista "Columna_medicamentos".



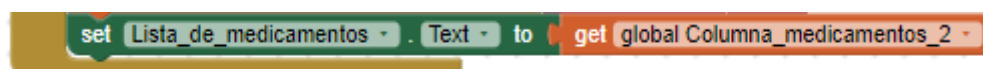
4.1.- Eliminar_medicamento, subbloque 4

Para eliminar los paréntesis se utilizará la función de reemplazar en todo el texto un segmento por otro: se graba Columna_medicamentos_2 con la información obtenida de su misma lista reemplazando un paréntesis "(" por un espacio en blanco " ". Y para el otro paréntesis ")" exactamente lo mismo.



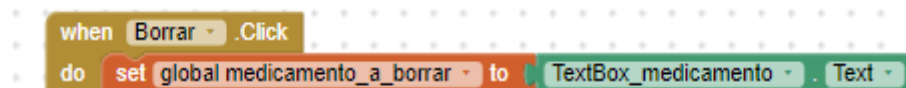
4.1.- Eliminar_medicamento, subbloque 5

Para finalizar el bloque, se escribe, en la etiqueta "Lista_de_medicamentos" el contenido de "Columna_medicamentos_2), mostrando así por pantalla los fármacos actuales enumerados.



4.1.- Eliminar_medicamento, subbloque 6

Bloque 2.- Al pulsar el botón "Borrar", la variable "medicamento_a_borrar" recibirá el valor obtenido de "TextBox_medicamento, que es el medicamento que el usuario ha escrito para eliminar.



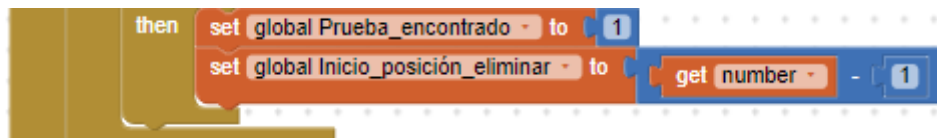
4.1.- Eliminar_medimento, subbloque 7

A continuación viene un bucle "for", cuyo objetivo será comprobar si el usuario ha introducido un medicamento que tiene en un lista actual de fármacos o no, dando un valor a una variable de proceder de una manera u otra. El bucle "for", para poder comparar todos los medicamentos actuales con el escrito por el usuario, tendrá una longitud de "Lista_medicamentos". Y la comprobación se efectuará con un "if" comprobando si la variable "medicamento_a_borrar" (el contenido de la casilla rellena por el usuario) es igual al elemento de índice "number" de Lista_medicamentos. (De esta manera en cada pasada comprobará el texto escrito por el usuario con cada fármaco).



4.1.- Eliminar_medimento, subbloque 8

Si se cumple la condición anterior, entonces se realizará las siguientes acciones: a).- "Prueba_encontrado" obtiene el valor 1 (este caso se utilizará para proceder a borrar el medicamento), y b). Se graba en la variable "Inicio_posición_eliminar" el valor donde se encuentra la enfermedad del fármaco asociado a ésta, para proceder a eliminarlo. La razón de esta forma de actuar es la siguiente: se ha decidido situar en la cadena "Lista_medicamentos" el principio de los datos referentes al fármaco a eliminar para después utilizar un bucle para llevar a cabo el borrado (se borrarían los cinco elementos asociados a los datos del fármaco en cuestión).



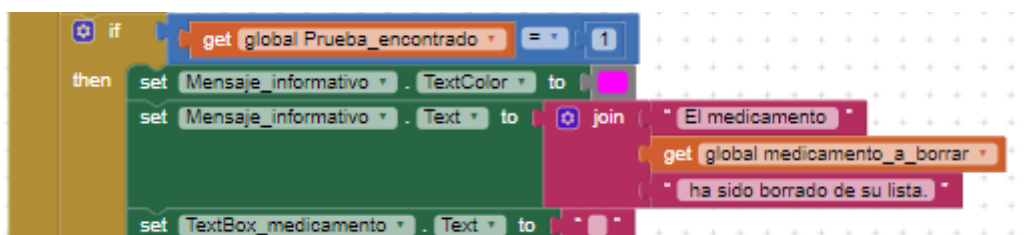
4.1.- Eliminar_medimento, subbloque 9

A continuación se hace una valoración con un "if" de la variable "Prueba_encontrado". Si ésta vale 0 quiere decir que el medicamento escrito por el usuario no está en su lista, por lo que se manda a la pantalla un mensaje de error indicando esto mismo (a través de la etiqueta "Mensaje_informativo") a la vez que muestra lo que el usuario había escrito (almacenado en la variable (medicamento_a_borrar). Esto se hace para que pueda observar si ha cometido algún error al escribirlo.



4.1.- Eliminar_medimento, subbloque 10

En el caso que sí coincidiera el texto escrito por el usuario (almacenado en "medicamento_a_borrar") con alguno de los medicamentos actuales (condición "Prueba_encontrado" = 1), se escribe igualmente un mensaje por pantalla utilizando la misma etiqueta anterior "Mensaje_informativo" e indicando que el medicamento en cuestión escrito por el usuario ha sido borrado de su lista (el borrado del medicamento se realiza a continuación de estas instrucciones, dentro aún del bloque "if", claro está).



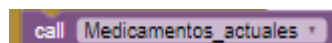
4.1.- Eliminar_medimento, subbloque 11

Para el borrado del fármaco se utilizará la "Lista_medicamentos" y se eliminarán de ahí los datos relativos al fármaco. Recuérdesse que los datos almacenados en dicha lista son del tipo: *[Año de nacimiento, enfermedad 1, fármaco1, frecuencia1, inicio1, duración1, enfermedad2, fármaco2, frecuencia2, inicio2, duración2, ..., enfermedad5, fármaco5, frecuencia5, inicio5, duración5]* (como máximo). De esta manera, es necesario eliminar cinco datos de dicha lista. "Inicio_posición_eliminar" indica el índice del elemento a partir del cual se deben borrar datos. Para ello se hace un bucle "for" que realice cinco veces la misma instrucción: Eliminar de la lista "Lista_medicamentos" el elemento de índice "Inicio_posición_eliminar", ya que una se comience a eliminar dichos elementos, el siguiente (sin llegar a cinco en el bucle) que debe eliminarse, vuelve a tener el susodicho índice.



4.1.- Eliminar_medicamento, subbloque 12

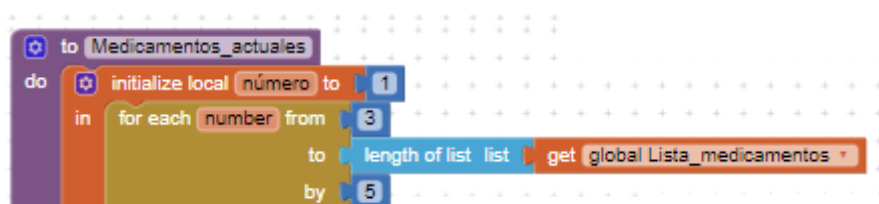
A continuación se hace una llamada a un procedimiento: "Medicamentos_actuales", el cual se comenta a continuación.



4.1.- Eliminar_medicamento, subbloque 13

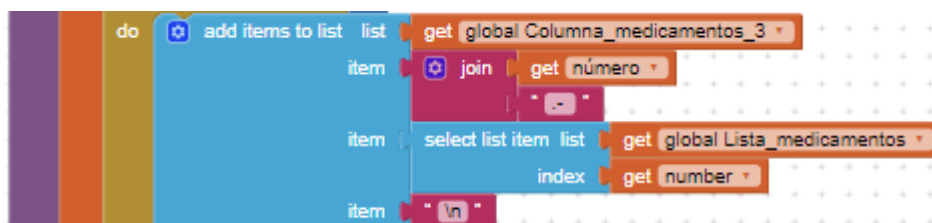
Procedimiento "Medicamentos_actuales". El objetivo de este procedimiento es mostrar por pantalla los medicamentos que tiene en su lista el usuario una vez se ha procedido al borrado

del medicamento elegido. El procedimiento realmente es similar al realizado en el bloque 1. La variable local "número" tiene como función enumerar los fármacos que saldrán por pantalla, y se incrementa su valor en uno cada vez que se ejecuta el bucle "for" de su interior como se verá más adelante. Se hace un bucle "for" que trabajará sólo con las posiciones de los fármacos en "Lista_medicamentos", como se ha explicado en otras ocasiones (el primer fármaco está en la posición tres de dicha lista y cada cinco índices posteriores se encuentra otro):



4.1.- Eliminar_medicamento, subbloque 14

A continuación se añade a la lista creada "Columna_medicamentos_3" los siguientes elementos: el valor de la variable local "número" para enumerar los fármacos (al comienzo será 1, en la siguiente pasada del bucle 2, ...), los símbolos punt y guión "-.", el nombre del fármaco que se haya en el índice "number" de "Lista_medicamentos" y el símbolo cambio de línea "\n" para que los fármacos actuales aparezcan en filas distintas.



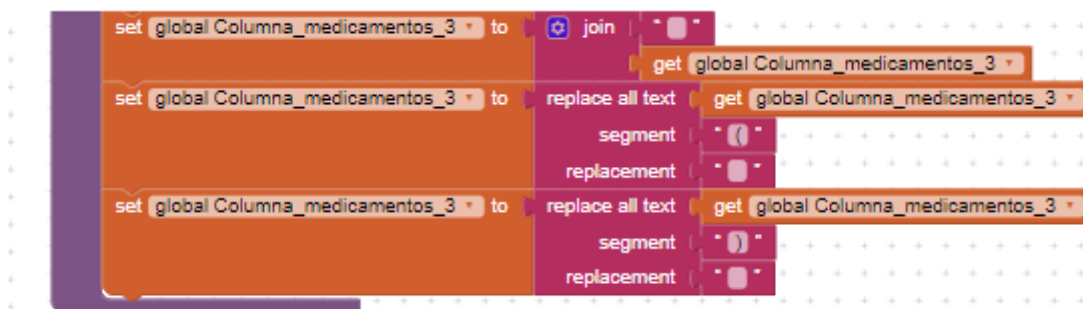
4.1.- Eliminar_medicamento, subbloque 15

Dentro todavía del bucle para poder actualizar el valor de la variable local "número", se le añade uno a su valor para la enumeración comentada de los fármacos:



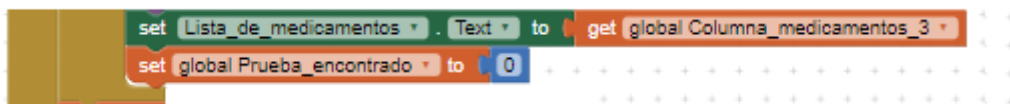
4.1.- Eliminar_medimento, subbloque 16

Para finalizar el procedimiento, se eliminan los dos paréntesis que tiene al principio y final de la variable "Columna_medamentos_3" y se le da un espacio en blanco al principio para que quede justificado el párrafo de las filas de "Columna_medamentos_3".



4.1.- Eliminar_medimento, subbloque 17

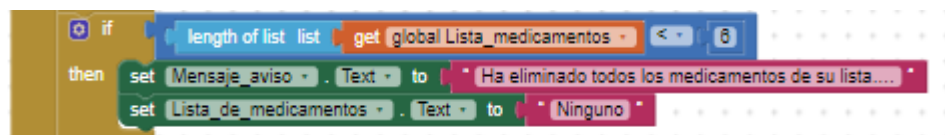
Continuación bloque 2.- Tras finalizar el procedimiento anterior se regresa al bloque 2 y se ejecuta la siguiente instrucción a la llamada de dicho procedimiento, cuyas instrucciones serán, por una parte, pasar los datos almacenados en "Columna_medamentos_3" a la etiqueta "Lista_de_medamentos", que será la que muestre por pantalla los fármacos actuales que tiene grabados el usuario en su cuenta. Por otra parte, se pone a 0 el valor de la variable "Prueba_encontrado", para así dejarlo preparado para cuando se desee borrar otro medicamento y que el programa lo pueda detectar (debido a que no se ha incluido en el "if" del Bloque 2 dicha instrucción en caso de escribirse incorrectamente el medicamento o no exista).



4.1.- Eliminar_medimento, subbloque 18

A continuación viene un bloque "if" que tiene como objetivo informar al usuario en caso de haber borrado todos los medicamentos. Para ello se comprueba la longitud de "Lista_medicamentos", si ésta es menor de 6 es indicativo que no hay medicamentos guardados, porque recuérdese que dentro de la lista "Lista_medicamentos" son los 6 primeros elementos los que contienen el primer fármaco.

Si de cada situación entonces se mostrarán por pantalla dos mensajes: el primero, a través de una etiqueta, advertirá que se han eliminados los medicamentos de la lista del usuario, y el segundo, a través de la etiqueta "Lista_medicamentos" mostrará la palabra "ninguno".



4.1.- Eliminar_medimento, subbloque 19

En la última parte del Bloque 2, se guardará en la etiqueta "verificación" de la base de datos TinyDB2 el valor 2. Esto se hace así para que la pantalla *Screen_Pastillero* sepa que los datos que recibe proceden de esta pantalla.



4.1.- Eliminar_medimento, subbloque 20

Bloque 3.- Finalizando con el diseño de la pantalla

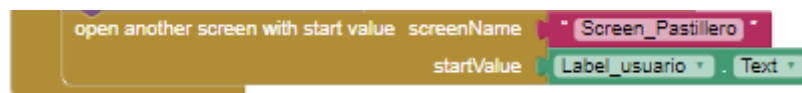
Eliminar_medicamento aparece el siguiente bloque, que se ejecuta al pulsar el botón “Volver” en dicha pantalla. Esto llevará a la ejecución de las siguientes instrucciones:

Deben guardarse los datos obtenidos en esta pantalla para poder utilizarse en *Screen_Pastillero*, y para ello se utilizará la TinyDB1 y TinyDB2, utilizando el modo grabar, como etiqueta “Label_usuario” y como datos “Lista_medicamentos” (donde estarán los actuales, tras haber eliminado los deseados).



4.1.- *Eliminar_medicamento*, subbloque 21

Continuando con la condición de pulsar el botón, la siguiente y última instrucción es abrir la ventana *Screen_Pastillero* y que ésta tome de “valor inicial” el contenido de la etiqueta “Label_usuario”.



4.1.- *Eliminar_medicamento*, subbloque 22

5.- CONCLUSIONES

Ha sido muy interesante programar con App Inventor 2, ha resultado un lenguaje ameno e interesante a la vez que potente, al menos para esta app y sin duda ha despertado en mí un interés por la programación que deseo llevar más allá. Puedo decir que aconsejo su uso a cualquiera que sienta la curiosidad de iniciarse en el mundo de las aplicaciones, desde mi punto de vista no se sentirá defraudado.

5.1.- Resultados

Se ha obtenido una aplicación útil que cumple el objetivo de poder ayudar al usuario evitándole errores y que a su vez le ahorra tiempo de organización.

La aplicación puede ser utilizada para llevar el seguimiento de todos los usuarios que desee, siempre y cuando no coincidan tanto el nombre como el año de nacimiento de los mismos, algo poco probable, y que podría ser solventado si se diera el caso añadiendo algún apellido al nombre del usuario. Asimismo incluye una posibilidad que se ha considerado interesante, que es la información relativa a la hora de comienzo de las tomas para darle más libertad: o bien a la hora de evitar horas nocturnas ajustando o empezando con el tratamiento adecuadamente (siempre podrá consultar en la planificación si le parece adecuado y sino fuera así cambiar dentro de lo posible la hora de comienzo de la toma del fármaco) o bien porque prefiera juntar dentro de lo posible las

tomas de varios fármacos.

Por otra parte, se ha obtenido un producto de fácil uso, muy intuitivo, y con posibilidades de crear una organización muy adecuada de hasta cinco fármacos para el tratamiento del paciente, además de la posibilidad de rectificar eliminando todos los medicamentos que desee desde la misma pantalla y volviendo a añadir los que considere sin sobrepasar el límite establecido de cinco fármacos simultáneos.

El diseño de las ventanas, aunque sencillo, es claro y no está sobrecargado, llevando al usuario a un manejo rápido desde el principio básicamente, algo que no siempre ocurre en las aplicaciones. A continuación se muestran algunas capturas juntas de la app que se han ido mostrando a lo largo de la memoria a modo de resumen.



INICIAR SESIÓN

CREAR USUARIO



Entrar

5.1.- Captura 1

5.1.- Captura 2

14:08 E

Screen_NuevoUsuario

NUEVO USUARIO

Nombre: Daniel López

Año de nacimiento: 1960

Enfermedad: Anemia

Fármaco: Glutaferro

Frecuencia (h): 12

Inicio (hora): 10

Duración tratamiento (d): 14

Guardar

Enfermedad:

Fármaco:

Frecuencia (h):

Inicio (h):

Duración tratamiento (d):

Añadir tratamiento Ir a Planificación

Guardar

5.1.- Captura 3

17:23 E

Screen_Pastillero

Daniel López 1960

Fármacos actuales

1.- Aspirina 500 mg

2.- Ibuprofeno

3.- Biodramina 50 mg

Planificación Eliminar medicamento Añadir

20/02/2021 08 a. m. Ibuprofeno

20/02/2021 09 a. m. Aspirina 500 mg

20/02/2021 12 p. m. Biodramina 50 mg

20/02/2021 04 p. m. Ibuprofeno

20/02/2021 09 p. m. Aspirina 500 mg

21/02/2021 12 a. m. Ibuprofeno

21/02/2021 08 a. m. Ibuprofeno

21/02/2021 09 a. m. Aspirina 500 mg

21/02/2021 12 p. m. Biodramina 50 mg

21/02/2021 04 p. m. Ibuprofeno

21/02/2021 09 p. m. Aspirina 500 mg

22/02/2021 12 a. m. Ibuprofeno

22/02/2021 08 a. m. Ibuprofeno

22/02/2021 09 a. m. Aspirina 500 mg

22/02/2021 12 p. m. Biodramina 50 mg

22/02/2021 04 p. m. Ibuprofeno

22/02/2021 09 p. m. Aspirina 500 mg

23/02/2021 12 a. m. Ibuprofeno

23/02/2021 08 a. m. Ibuprofeno

23/02/2021 09 a. m. Aspirina 500 mg

23/02/2021 12 p. m. Biodramina 50 mg

23/02/2021 04 p. m. Ibuprofeno

23/02/2021 09 p. m. Aspirina 500 mg

24/02/2021 12 a. m. Ibuprofeno

24/02/2021 08 a. m. Ibuprofeno

24/02/2021 09 a. m. Aspirina 500 mg

24/02/2021 12 p. m. Biodramina 50 mg

24/02/2021 04 p. m. Ibuprofeno

24/02/2021 09 p. m. Aspirina 500 mg

25/02/2021 12 a. m. Ibuprofeno

25/02/2021 08 a. m. Ibuprofeno

25/02/2021 09 a. m. Aspirina 500 mg

25/02/2021 12 p. m. Biodramina 50 mg

5.1.- Captura 5

14:20 E

Screen_Pastillero

Daniel López 1960

Fármacos actuales

1.- Aspirina 500 mg

2.- Biodramina 50 mg

3.- Glutaferro

Planificación Eliminar medicamento Añadir

Volver a pantalla principal Cerrar aplicación

5.1.- Captura 4

22:36 E

Eliminar_medimento

Daniel López 1960

ELIMINACIÓN DE MEDICAMENTOS

Medicamento a eliminar

Volver Borrar

El medicamento Ibuprofeno 600 mg ha sido borrado de su lista.

Fármacos actuales

1.- Aspirina 500 mg

2.- Biodramina 50 mg

5.1.- Captura 6

5.2.- Propuestas de mejora de la app

Hay algunas mejoras para la app que me habría gustado llevar a cabo, entre las cuales estarían:

- Algo de mucho valor habría sido añadir alarmas de recordatorio para facilitar más al usuario la tarea de sus tomas, no teniendo así que estar pendiente o usar otra alarma.
- También se podría implementar la opción de guardar la planificación en un fichero e imprimirlo.
- Aumentar el máximo número de fármacos, ya que actualmente ese límite está en cinco y podría ser escaso para algunas personas.
- Sería también interesante que mostrase por pantalla, en la planificación, las enfermedades que se corresponden a cada medicamento, a modo de información aclaratoria o recordatoria.
- Como información que se pide al usuario para grabar sus datos está el tiempo de duración del tratamiento. Actualmente no ofrece la posibilidad de que éste sea crónico, de manera que debe añadir el fármaco en cuestión el usuario cada cierto tiempo y repetir el proceso cuando pase ese tiempo.
- Sería muy intuitivo para el usuario que la planificación apareciera a modo de calendario en vez de en forma de lista.

6.- BIBLIOGRAFÍA

<http://kio4.com/appinventor/index.htm>

<https://es.statista.com/grafico/8473/android-se-acerca-al-liderazgo-de-los-sistemas-operativos/>

<https://www.estrategiaynegocios.net/tecnologia/1199435-330/por-qu%C3%A9-android-lidera-el-mercado-de-los-sistemas-operativos-para-m%C3%B3viles>

<https://www.latercera.com/que-pasa/noticia/android-sistema-operativo-mas-utilizado/664877/#:~:text=Android%20actualmente%20es%20responsable%20de,aumentado%20ocho%20veces%20sus%20ingresos.>

<https://www.itreseller.es/en-cifras/2020/05/el-86-de-los-smartphones-vendidos-a-nivel-mundial-son-android#:~:text=Seg%C3%BAn%20la%20%C3%BAltima%20investigaci%C3%B3n%20de,el%20mundo%20en%20el%20trimestre.>

[http://repositorio.puce.edu.ec/bitstream/handle/22000/13113/Anexo%203%20MIT%20App%20Inventor%202.pdf?sequence=4&isAllowed=y#:~:text=App%20Inventor%202%20\(AI2\)%20es,sencilla%20aplicaciones%20para%20dispositivos%20Android.](http://repositorio.puce.edu.ec/bitstream/handle/22000/13113/Anexo%203%20MIT%20App%20Inventor%202.pdf?sequence=4&isAllowed=y#:~:text=App%20Inventor%202%20(AI2)%20es,sencilla%20aplicaciones%20para%20dispositivos%20Android.)

<https://www.webconsultas.com/tercera-edad/hogar-y-consumo/que-es-un-pastillero-y-por-que-conviene-usarlo>

<http://explore.appinventor.mit.edu/about-us>

<https://play.google.com/store/apps/details?id=eu.smartpatient.mytherapy&gl=ES>

<https://play.google.com/store/apps/details?id=com.devsoldiers.calendar.pills.limit&gl=ES>

https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/