



“Pastillero electrónico para usuarios con historial cardíaco”

Caballero-Córdoba Mario Tadeo, Cabrera-Rodríguez Itzel del Carmen, Castro-Canchola Ana Cristina, Jerez-Gómez Gabriela, Pérez-Rosas Pedro Iván. Universidad Modelo. Ingeniería Biomédica. Ingeniería Biomédica. Carretera a Cholul, 200 mts. después del periférico, Mérida, Yucatán México, CP. 97305. Teléfono. 999 179 4639, 15234434@modelo.edu.mx

Palabras clave: *pastillero, medicación, Arduino.*

Resumen. En el presente artículo se propone el desarrollo de un pastillero electrónico programado en Arduino uno, para personas con historial de infarto cardiaco, así como adultos mayores que necesiten apoyo para tener presente las horas adecuadas que se le ha asignado su medicación, así como también de personas que no llevan la ingesta de sus medicamentos en el orden requerido.

Se analizó y evaluó la situación actual de las enfermedades cardiovasculares en nuestro país, se sugirió a un usuario ideal, algo igual que se analizó la demanda que existe entre los pastilleros electrónicos y la competencia que existe, la competencia se centra en la innovación tecnológica, la facilidad de uso, la durabilidad y el costo.

La decisión en el diseño de un pastillero electrónico se realizó con base en las características del usuario ideal, enfocándonos en que tenga una mejora en la adherencia a su medicación y que pueda tener una vida más saludable. El resultado de este proyecto es el diseño de pastillero electrónico, con base en el uso de nuevas tecnologías vinculadas software en Arduino uno, que además de llevar un control en el tratamiento del paciente, cuenta con un modelo ergonómico de fácil uso y cómodo para el hogar.

Introducción.

Se llevará a cabo el diseño y construcción de un pastillero electrónico controlado y programado con arduino, para mejorar la adherencia a la medicación. El prototipo para realizar contará con una capacidad de hasta 4 distintos medicamentos, según el tratamiento y la dosis que haya sido asignada a cada usuario. De igual forma, tendrá un

reloj digital integrado que señalará la hora. Cuando este reloj marque la hora en que se debe de tomar algún medicamento (previamente programada), se activará una alarma y se encenderá una luz que indique al paciente que es hora de tomar la prescripción dada por el médico. Cada uno de los compartimentos del pastillero rotará si el reloj indica que es la hora de la medicación. De esta forma, se evitarán accidentes debidos a confusiones en los horarios de los medicamentos. Primeramente, se realizará el código de programación, con base a un diseño conceptual esperado del pastillero. Una vez que se consiguió un avance funcional, se procedió a construir, de manera simultánea el prototipo de pastillero. Ya que las partes las dos partes fueron terminadas, se realizó una prueba en conjunto de todo el sistema. Basado en el resultado, se decidió si debía haber algún cambio al modelo o al código de programación, lo cual tuvimos que agregar para tener un producto óptimo. Despues de haber sido revisado y que no existiera ninguna clase de problema, se procedió al modelado del pastillero como tal, que se considera hacerse con PLA en impresión 3D, ya que es una manera óptima de construir el pastillero, lo que la hace fácil de trabajar y dar forma. Además, es moldeable lo que hace que sea fácil de modificar en caso de que fuera requerido. Es un plástico que, al impacto y al desgaste, y también es resistente a la descomposición y la putrefacción. Por último, se hizo el ensamblaje de todas nuestras plantillas cortadas y conectar los circuitos eléctricos que nos llevarán a la realización de nuestro objetivo final, que es obtener un pastillero electrónico funcional y capaz de desplegar la medicación programada. El objetivo general de nuestro



Este proyecto era construir un pastillero electrónico que mejore la adherencia a la medicación. Sus objetivos específicos contaban con realizar un código de programación funcional que se adecue a los requerimientos del pastillero, diseñar un modelo de pastillero electrónico ergonómico y funcional, construir un prototipo de pastillero electrónico sencillo, fácil de usar y distinto a los comerciales, proponer una opción de seguimiento de tratamiento médico que favorezca la adherencia a la medicación, principalmente a usuarios con historial cardíaco. Este pastillero electrónico combina la versatilidad de la plataforma Arduino con la funcionalidad específica de ayudar a las personas a seguir de manera efectiva y segura sus regímenes de medicación. Equipado con sensores, una interfaz amigable y la capacidad de conectividad, este dispositivo no solo sirve como un simple contenedor de pastillas, sino como un asistente personalizado para el cuidado de la salud. En resumen, el pastillero electrónico con Arduino representa una fusión innovadora entre la tecnología y la salud, ofreciendo una solución inteligente para el seguimiento y la administración de medicamentos. Este dispositivo no solo simplifica la vida diaria, sino que también promueve una mayor autonomía y cuidado personalizado, abriendo un nuevo capítulo en la gestión eficiente de la salud.

Materiales y métodos.

tabla #1 materiales para el circuito

Arduino Uno		El arduino es un hardware, que nos servirá como microcontrolador para nuestro proyecto y lo haremos funcionar de manera
-------------	--	---

		independiente controlado y alimentando dispositivos como la pantalla lcd.
Motor paso a paso 28byj-48 y su microcontrolador		El Motor paso a paso nos servirá para que rote el pastillero, este será alimentado por una alimentación externa de 5V
Pantalla LCD		La pantalla LCD nos servirá para despegar la hora, conjunto de la dosis que irá en cada pastillero, al igual que se podrá devolver al casillero anterior en caso de ser necesario
Botones		Los botones servirán como pulsadores que ayuden a la realización que el usuario desee
cable de arduino		Los cables de arduino nos permiten conectar el arduino al ordenador donde se ha creado un código específico para el funcionamiento de el pastillero electrónico.
módulo ds3232		El módulo ds3232 mantiene la cuenta precisa del tiempo, incluso cuando se ha ido la luz, esta nos ayudará a programar las alarmas las cuales tendrá el pastillero para avisar al usuario sobre su toma de medicamentos



módulo I2C		Ayuda con el manejo de la pantalla lcd para que sea más fácil de manejar
estaño		El estaño es un herramienta indispensable para soldar nuestra pantalla LCD y botones
cautín		El cautín es un herramienta indispensable para soldar nuestra pantalla LCD y botones

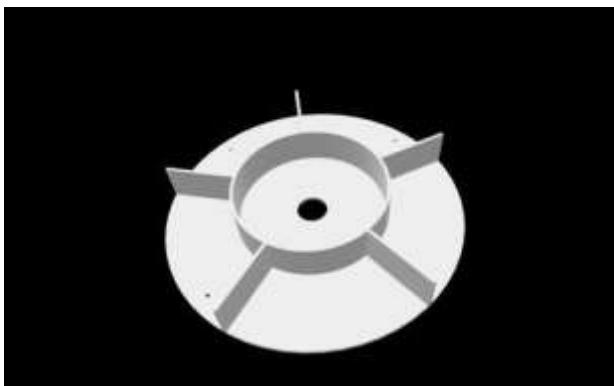


Fig.#2 Contenedor de las pastillas

La forma más sencilla de almacenar la dosis de pastillas recomendada para el usuario con historial de infarto cardiaco era en un círculo con contenedores separados para cada juego de la medicación alrededor del borde. Cada recipiente debía ser lo suficientemente grande como para contener suficientes comprimidos. Se decidió bajo el criterio del usuario ideal tener 4 contenedores

más uno de repuesto para poder colocarlo en la abertura cuando se recargara el dispensador, para un total de 5. Esto luego determinó el tamaño que debía tener.

Diseño del pastillero

Para diseño del pastillero está inspirado en un pastillero electrónico (autodesk instructable) en donde ya estaba el diseño en onshape estaba publicado en la web, el cual fue usado para el bosquejo de este proyecto, ya que al tener muy poco tiempo y falta de conocimientos del programa onshape, optamos porque era óptimo usar una plantilla, el equipo se encargó de realizar sus respectivas modificaciones para que se adecuara con nuestros objetivos propuestos.

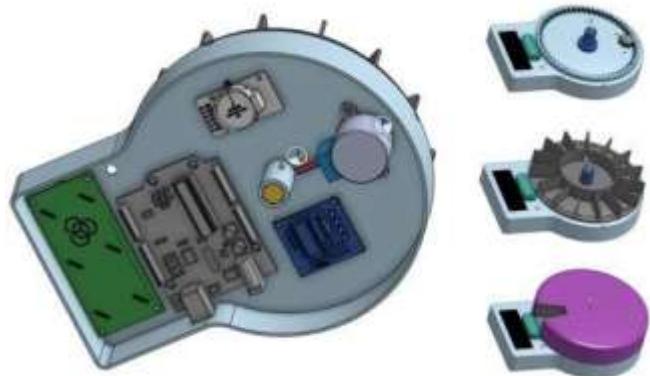


Fig.#1 plantilla inicial de nuestro proyectos (autodesk instructable)

El criterio para el dispensador fue:

- Tenía que ser compacto para que pudiera colocarse sin ocupar demasiado espacio en la encimera de la cocina.
- Ser capaz de retener y dispensar los medicamentos de un usuario con historial de infarto cardíaco, durante un día en concreto.
- El tiempo de dispensación se puede ajustar.



- Proporcionar una visualización visual de la hora y el momento en que las tabletas estaban listas.

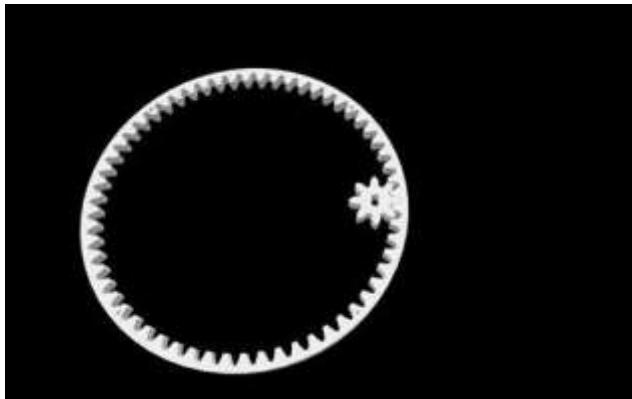


Fig.#3. anilla de engranaje interno y uno pequeño para el motor paso a paso

Lo siguiente fue cómo rotar el dispensador. Tenía un motor paso a paso 28BYJ. Se decidió usar un anillo de engranaje interno fijado en la parte inferior del dispensador que sería impulsado por un pequeño engranaje en el motor paso a paso. Lo cual tenía una proporción de engranajes de 7,5 a 1, que es divisible en 15, por lo que pude configurar fácilmente el motor paso a paso para que lograra rotar media revolución y mover cada contenedor exactamente.

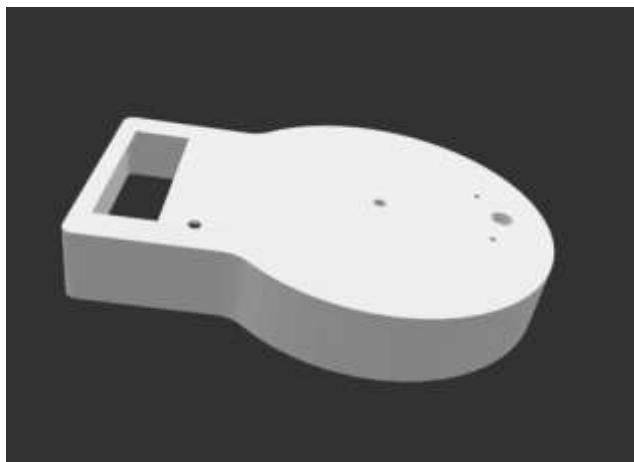


Fig.#5 Base del pastillero electrónico

La base de la unidad tenía que contener todos los componentes electrónicos y el motor, e idealmente tenía

localizadores para que las distintas piezas pudieran montarse y pegarse fácilmente en su lugar.



Fig.#4 Tapa del pastillero electrónico

La tapa de la unidad estaba pensada para poder tener una abertura lo suficientemente grande para que encajaran con los contenedores del pastillero.

Código de programación

```
1 // sección Sección electrónica
2
3 // Variables
4 #include <LiquidCrystal.h>
5 LiquidCrystal_I2C lcd(12,11,5,4,3,2);
6
7 // Sesión de inicio
8 int batteryLevelPin = 11; //Pin que indica el nivel de batería
9 int batteryLevelThreshold = 10; //Nivel de batería en el que se activa el sensor
10 int batteryWarningPin = 9; //Pin que indica si la batería es baja
11
12 // Sesión de variables
13 int mesu1; //Pin del switch principal del vaso Vesp
14 int opto1; //Pin del switch secundario que está dentro del case 1 del switch de mesu en el vaso Vesp
15 int opto2; //Pin del switch secundario que está dentro del case 2 del switch de mesu en el vaso Vesp
16 int entradaSal1; //Pin en el que se coloca el sensor
17 int gabineteDeContenedores; //Parcheado en que se coloca el sensor
18
19 // Sesión de funciones
20 void doControl(); //Para sacar las cosas de cada contenedor dentro de los casilleros - 8,11,12,9
21 void sacarMesa1(); //Para sacar la mesa en caso que se active la alerta
22 void ultimodosis(); //Para sacar la que tiene en dia la ultima dosis
23 void sacarUltimo(); //Para sacar la que tiene en dia la ultima dosis
24 void PastilleroElectronico(); //Para sacar el pausa el pastillero en la expediente
25
26 // Sesión de banderas
27 bool bandera1; //Bandera que se utiliza en el caso de sacar
28 bool bandera2; //Bandera que se utiliza en el caso de sacar
29
30 // Sección Sección declarar //////////////////////////////////////////////////////////////////
```

Inicia la “**sección declarar**”, aquí se declara los pines, es donde se conectan todos los componentes que se estén



utilizando en el circuito. De igual manera, se declaran todas las variables que se estén utilizando para el código.

```
35 void setup()
36 { // aqui inicia el Void setup
37
38 // Inicia pinsMode
39 pinMode(botonMoverse,INPUT);
40 pinMode(botonSeleccionar,INPUT);
41 pinMode(botonMoverAtras,INPUT);
42 // acaba pinsMode
43
44 lcd.begin(16,2);
45 lcd.init();
46 lcd.backlight();
47 lcd.clear();
48 // aqui acaba el void setup ////////////////////////
```

```
52 // Inicia Sección Funciones
53 void entradaMenu()
54 { // aqui inicia el Void entradaMenu-
55 // aqui acaba el Void entradaMenu ///////////////////////
56 void opciones (String mensaje, int C, int F)
57 { // aqui inicia el Void opciones
58 lcd.setCursor(C,F);
59 lcd.print(mensaje);
60 // aqui acaba el Void opciones /////////////////////
61
62 void botonMover (int boton,int operaciones)
63 { // aqui inicia el Void botonMover-
64 // aqui acaba el Void botonMover /////////////////////
65
66 void procedimientosMenu()
67 { // aqui inicia el Void procedimientosMenu-
68 // aqui acaba el Void procedimientosMenu /////////////////////
69
70 void menus()
71 { // aqui inicia el Void menus-
72 // aqui acaba el Void menus /////////////////////
73
74 void seleccionar()
75 { // aqui inicia el Void seleccionar-
76 // aqui acaba el Void seleccionar /////////////////////
77
78 void limites()
79 { // aqui inicia el Void limites-
80 // aqui acaba el Void limites /////////////////////
81
82 void opciones2 (int mensaje, int C, int F)
83 { // aqui inicia el Void opciones2-
84 // aqui acaba el Void opciones2 /////////////////////
85
86 void testeoExpotronica (String mensaje, int C, int F) // PROBAR para testeos en la expotronica
87 {-
88
89 // acaba Sección Funciones
```

En la función “**void setup**”, es donde declaramos o “ponemos” lo que queramos que aparezca siempre que arrancamos el código, en este caso declaramos que siempre usaremos los tres botones del pastillero para entrada, y en la parte de abajo declararemos que siempre se iniciara con una LCD 16x2.

Inicia toda la “**sección de funciones**”, cada una tiene donde inicia y dónde acaba para una mejor organización al momento de leer el código, aquí se utilizan todas las funciones agregadas hasta el momento para el código, cada función tiene una utilidad en el código, el “**void entrada Menú**” se utiliza para la entrada al código, es decir, lo que aparecerá cuando prendamos el pastillero, el “**void opciones**” se utiliza como unas líneas de código (mensaje) universal, ya que ahí mismo tenemos lo que aparecerá siempre que lo declaremos, solamente le cambiamos al mensaje que quiera que aparezca y su respectiva ubicación en la pantalla LCD.



```
85 void botonMover (int boton,int operaciones)
86 { // aqui inicia el void botonMover
87   if (digitalRead(boton)==1)
88   {
89     bandera=true;
90     while (bandera == true)
91     {
92       if (digitalRead(boton) == 0)
93       {
94         bandera=false;
95         opciones=opciones+operaciones;
96         lcd.clear();
97       }
98     }
99   }
100 }
101 // aqui acaba el void botonMover
```

En el “**void botón Mover**” se utiliza igualmente como una función universal, ya que declaramos que con cualquier botón que apretemos se hará una operación, dependiendo el caso, será si avanza entre las opciones o retrocede.

aparecen los 4 casilleros que se van a utilizar, con sus respectivas funciones de cada uno y su opción de salir para regresar el mensaje inicial de cuando inicias el casillero

```
176 void seleccionar()
177 { // aqui inicia el void seleccionar
178   if (digitalRead(botonSeleccionar)==1)
179   {
180     banderaBotontrue;
181     while (banderaBoton == true)
182     {
183       if (digitalRead(botonDeSeleccionar)==0)
184       {
185         banderaBoton=false;
186         salirDelMenu();
187       }
188     }
189   }
190   // inicia el switch general
191   case 0:// seleccionada estando en el menú 0 (entrada al casillero)
192     menu0; // declara que se está en el menú 0 una vez se aprieta el botón
193     break;
194   case 1 // menú 1 casillero
195     guardadoCasillero=opciones=1; // para saber que casillero elegí para el writer datosCasilleros
196     subOpciones();
197   {
198     case 1:
199       menu1;
200       break;
201     case 2:
202       menu2;
203       break;
204     case 3:
205       menu3;
206       break;
207     case 4:
208       menu4;
209       break;
210     case 5:
211       menu5;
212       break;
213   }
214   break;
215 }
```

```
318 void menus()
319 { // aqui inicia el void menus
320   switch(case1)
321   {
322     case 0:// menú 0, escogido
323       entrarMenu0();
324     break;
325     case 1:// inicia caso del Caso principal (menu1, // menú 1
326     switch (opciones)
327     {
328       case 1:// opción 1
329         opciones ("Casillero 1",0,0);
330         break;
331       case 2:// opción 2
332         opciones ("Casillero 2",0,0);
333         break;
334       case 3:// opción 3
335         opciones ("Casillero 3",0,0);
336         break;
337       case 4:// opción 4
338         opciones ("Casillero 4",0,0);
339         break;
340       case 5:// opción 5
341         opciones ("Salir",0,0);
342         break;
343     }
344   }
345   break; // acaba bucle de menú casilleros
```

El “**void menus**” es donde tiene todo lo principal del pastillero, este se mueve por un switch general (menú principal si lo llamamos de otra manera), y este tiene switch general tiene cierto número de “case”, cada case, que sería como una opción, tienen otro “sub-switch” es decir, las opciones que lleva, como se muestra en la imagen,

Inicia “**void seleccionar**” aquí es donde se le “asigna” una opción a cada uno de los casilleros, como todo se conforma de un switch general, que serían como menús generales, es decir los principales, y cada sub-switch dentro de uno general es como un sub-menú. En la imagen se aprecia como el switch 1 (switch principal, o menú principal 1) tiene otro switch y cada case de los switch tiene un menú=x, esto porque en void menus cada switch general te guía hacia un apartado, aquí lo que hacemos es que se lo asignamos, es decir, case 1 es menú=2, donde iría el casillero 2 con sus respectivas opciones, y así con cada apartado.



Inicia “**void límites**”, aquí le asignamos al apartado que queramos el “límite” para que no se pase de opciones, es decir, si casillero 1 tiene 3 apartados, y le da 4 veces al botón de avanzar, que no se pase de las opciones y le deje la pantalla en blanco, entonces se le asigna límites, por si llega a avanzar más de las opciones establecidas que le regrese a la primera opción, y de igual manera de forma contraria que si está en la opción una y le da a retroceder, que no le aparezca la pantalla en blanco, si no que se quede en la opción 1.

Aquí están 3 funciones “void opciones2”, “void testeo Expotrónica” y “void loop”,

En “void opciones 2” es como void opciones 1, solo que aquí en vez de escribir un mensaje, escribo una variable, para que se me facilite al momento de programar algo y ocupe menos líneas de código.

“void testeo Expotrónica”, solo nos sirve para cuando se muestre en la presentación, sea de manera eficaz y rápida con un tiempo de espera mucho menor, es decir, si en un casillero es cada 12 horas una pastilla, aquí para mostrarles el ejemplo lo ponemos cada 10 segundos.

El “void loop” es lo que se repetirá un ciclo indeterminado de veces, aquí es lo que queremos que se repite hasta que le digamos que ya no lo haga, aquí solo tenemos 3 funciones muy útiles, que es la de procedimientos, que solo es lo que indica si los botones avanzan o retrocede, la opción de los límites para que jamás se pase de las opciones que queramos, y la función de menús, que es donde aparecen todas las opciones del casillero.



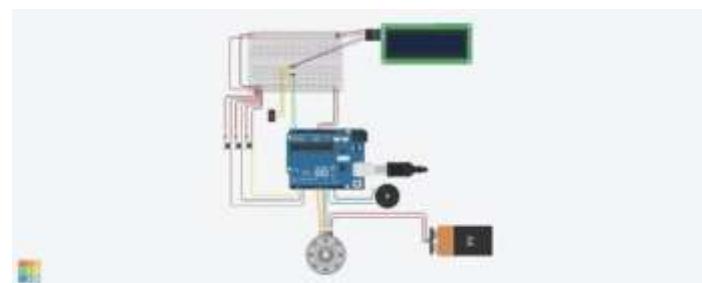
Construcción del circuito

IMAGEN	NOMBRE
	Pila de 9V (Simulando el toma corriente)
	Buzzer
	Boton
	Arduino Uno
	Pantalla LCD con modulo I2C
	Motor paso a paso con microcontrolador

	Placa de pruebas (simulando la placa PCB)
	Temporizador (Simulando el módulo de horario RTC DS1307)

Materiales sustituidos en el modelo grafico:

Imagen	Nombre
	tomacorriente (simulado con la pila de 9V)
	Modulo DS1307 con I2C (simulado con el temporizador)
	PCB (simulado por la placa de pruebas)



En la figura que se muestra podemos apreciar el boceto del circuito en el cual se distinguen sus componentes,



podemos empezar con el ARDUINO UNO, que será el cerebro de toda nuestra programación para los circuitos, al igual se presenta una PANTALLA LCD que contará con el MÓDULO I2C que permite liberar espacio tanto del Arduino como en la conexión del circuito (se refleja en la parte de atrás de la pantalla por lo que no se ve) y esta pantalla se mostrará el menú principal, la hora local, las dosis que deberá tomar el paciente, la medicina que se le administra, etc. Igual tiene 3 BOTONES que son SELECCIONAR, RETROCEDER Y AVANZAR, para que nuestro usuario pueda indagar dentro de este menú, tenemos también el BUZZER que será la alarma por la cual el propio pastillero emitirá un sonido el cual le recordará al paciente cuando es la hora de tomar la pastilla, tenemos el motor paso a paso con su microcontrolador el cual conforme a sus RPM (revoluciones por minuto) previamente ya configurados con el equipo, dará una serie de giros calculados para que quede exactamente en el casillero que se desea dentro del pastillero, este igual regresara a su punto cero que será el casillero vacío que no contendrá nada para no maltratar o humedecer las pastillas el paciente, contamos igual dentro de la figura con el MODULO DE HORARIO (en la figura representado por un contador 666) el cual nos ayuda a tener la hora local dentro del pastillero sin necesidad de configurar el Arduino para que lo tenga, este módulo modelo RTC DS1307 con I2C cuenta con una memoria interna la cual guarda la hora aunque esté apagado el pastillero para que cuando se vuelva a encender para que no se tenga que configurar cada que se apague el pastillero y finalmente en la figura tenemos una PILA DE 9V que sirve como referencia de que nosotros conectaremos a una toma de corriente nuestro circuito, ya que el Arduino uno por si solo no tiene el voltaje necesario para distribuir la energía necesaria para que todo el circuito cuente con la fuerza de voltaje sin necesidad de quemar el Arduino uno por la cantidad excesiva de voltaje que requiere este circuito, todo esto se hará con un eliminador de corriente que tendremos conectado a la toma de corriente, no obstante se tiene que

señalar que la toma de corriente se usará exclusivamente para el motor paso a paso, ya que se puede conectar mediante el eliminador y usara la energía que sea necesaria para que pueda girar sin necesidad de forzar al Arduino a llevar toda la energía al motor y así la pantalla LCD y los botones solo estarán conectado al Arduino y no sobrepasaremos el límite de este.

Hemos pasado por una serie de configuraciones dentro de nuestro circuito que conforme el paso del tiempo hemos implementado y quitado para el correcto funcionamiento de nuestro circuito ya que hemos pasado por una serie de problemas cuando quisimos conectar como queríamos, nos dimos cuenta de nuestros errores por medio de los profesores que nos instruyeron antes de hacer nuestra conexión, también tuvimos una serie de dudas cuando queríamos ver la manera de que se pueda conectar a una toma de corriente, ya que no queríamos quemar nuestros componentes, también cuando se presenta la idea de meter todo en una placa PCB, por que se tenía planeado el circuito para que esté abajo del pastillero y que este armado mediante conexiones y extensores de cable para que todo quede bien distribuido y menos junto dentro del pastillero, ya que se pensó para que cuando el técnico (en este caso nosotros) quiera arreglar cierta parte del circuito lo abra y se encuentre con un circuito mas claro y simple de nuestra conexión para que sea más eficiente al momento de reparar o hacer algún cambio dentro de este.

La última observación que tuvimos por parte de nuestro profesor es que el módulo de horario y nuestra pantalla LCD compartían los mismos pines análogos dentro de nuestro Arduino y tuvimos que ver la forma de que se pueda compartir los pines sin necesidad de implementar un segundo Arduino solo por dos pines, lo que hicimos es que dentro de nuestra placa de pruebas conectamos esos pines y en la misma línea de este conectamos tanto el módulo como la pantalla y así resolvimos el último problema que



tuvimos dentro de nuestra conexión (como se observa dentro el modelo gráfico)

Resultados.

Para visualizar nuestros resultados podemos comenzar viendo nuestro objetivo general el cual es construir un pastillero electrónico que mejore la adherencia a la medicación, tomando en cuenta esto podemos decir que si cumplimos con nuestro objetivo, aunque este objetivo es bastante general por lo que si queremos visualizarlos a mayor detalle usaremos nuestros objetivos específicos.

En cuanto a la construcción en general podemos observar que nuestro ensamblaje llevó algunas modificaciones que se realizaron a mano en la universidad, algunas con ayuda de las máquinas disponibles en el área de mecatrónica, también se pintó el pastillero con el fin de darle un mejor aspecto del que tenía, escogimos los colores negro y azul.

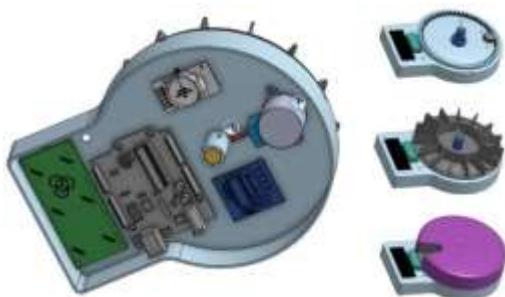


Fig.#Diseño del pastillero:se muestra el diseño sin cambio de color.

Evidencia del diseño ya impreso:

<https://drive.google.com/file/d/1s8k4D0Rq5qiPDKmr5uk3Z5OSu4kAp9h6/view?usp=drivesdk>

Nuestro primer objetivo es realizar un código de programación funcional que se adecue a los requerimientos del pastillero, este se cumple en su totalidad ya que es de vital importancia para el funcionamiento óptimo de nuestro pastillero, sin este no hay proyecto, este código tuvo diferentes versiones y se fue actualizando conforme al asesoramiento de nuestro profesor de programación el cual

nos recomendó varias herramientas para hacer nuestro código más eficiente y corto.

Evidencia del código menú en funcionamiento:

https://drive.google.com/file/d/181vtrbgZ9q_IUO8-3D9UuYaT9-Hx2N_1/view?usp=drivesdk

Para nuestro segundo objetivo el cual es diseñar un modelo de pastillero electrónico ergonómico y funcional podemos encontrar que este si se cumplió ya que nuestro diseño es de fácil uso, pequeño y solo requiere de conectarlo a la corriente para que comience con su función.

El proceso de decisión de diseño para nuestro proyecto no llevó mucha dificultad ya que encontramos herramientas como el software Onshape que nos permite modelar a 3D diversos proyectos, para nuestro diseño encontramos plantillas que fueron modificadas a conveniencia del diseño ya plasmado. Este se mandó a imprimir cada pieza por separado con el profesor del taller de robótica de la universidad, se obtuvieron aproximadamente 6 piezas independientes que conforman nuestro diseño, las cuales son:

- · Engranaje pequeño
- · Engranaje grande
- · Tapa
- · Estructura de casilleros
- · Cobertura del tornillo
- · Base

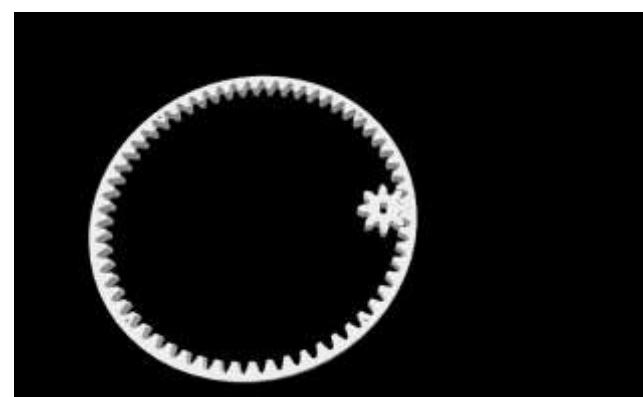
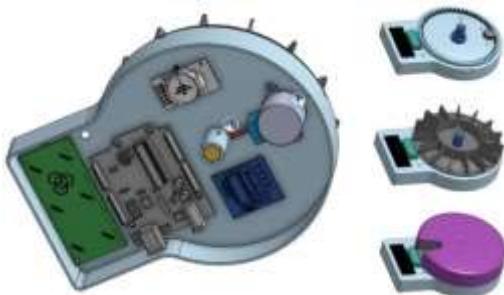




Fig.# Engranaje pequeño y grande: se muestra la forma en la que se representó y diseño nuestros engranajes.

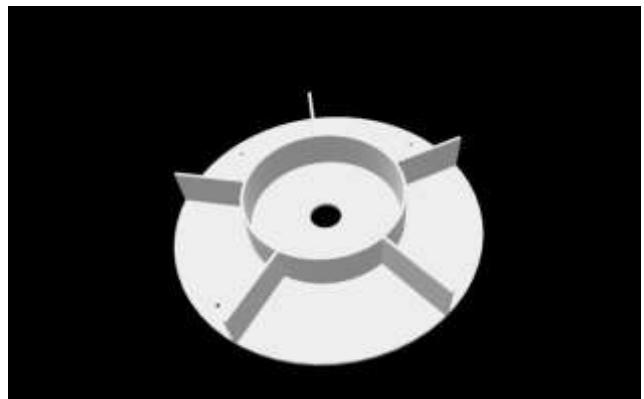
Diseño del pastillero

Para diseño del pastillero está inspirado en un pastillero electrónico (autodesk instructable) en donde ya estaba el diseño en onshape estaba publicado en la web, el cual fue usado para el bosquejo de este proyecto, ya que al tener muy poco tiempo y falta de conocimientos del programa onshape, optamos porque era óptimo usar una plantilla, el equipo se encargó de realizar sus respectivas modificaciones para que se adecuara con nuestros objetivos propuestos.

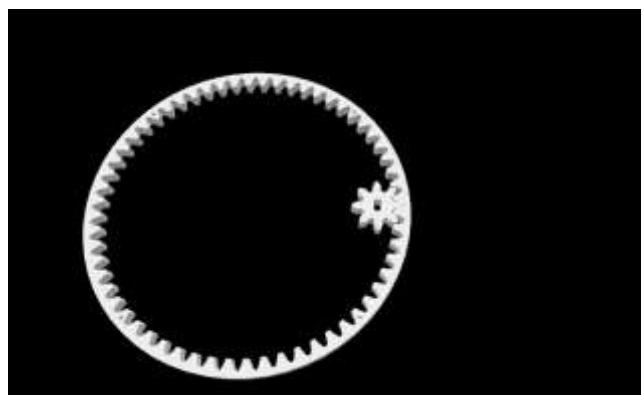


El criterio para el dispensador fue:

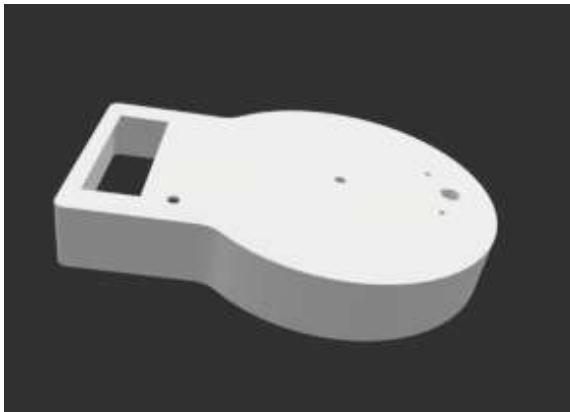
- Tenía que ser compacto para que pudiera colocarse sin ocupar demasiado espacio en la encimera de la cocina.
- Ser capaz de retener y dispensar los medicamentos de un usuario con historial de infarto cardíaco, durante un día en concreto.
- El tiempo de dispensación se puede ajustar.
- Proporcionar una visualización visual de la hora y el momento en que las tabletas estaban listas.



La forma más sencilla de almacenar la dosis de pastillas recomendada para el usuario con historial de infarto cardiaco era en un círculo con contenedores separados para cada juego de la medicación alrededor del borde. Cada recipiente debía ser lo suficientemente grande como para contener suficientes comprimidos. Se decidió bajo el criterio del usuario ideal tener 4 contenedores más uno de repuesto para poder colocarlo en la abertura cuando se recargara el dispensador, para un total de 5. Esto luego determinó el tamaño que debía tener.



Lo siguiente fue cómo rotar el dispensador. Tenía un motor paso a paso 28BYJ. Se decidió usar un anillo de engranaje interno fijado en la parte inferior del dispensador que sería impulsado por un pequeño engranaje en el motor paso a paso. Lo cual tenía una proporción de engranajes de 7,5 a 1, que es divisible en 15, por lo que puedo configurar fácilmente el motor paso a paso para que lograra rotar media revolución y mover cada contenedor exactamente.



La base de la unidad tenía que contener todos los componentes electrónicos y el motor, e idealmente tenía localizadores para que las distintas piezas pudieran montarse y pegarse fácilmente en su lugar.



La tapa de la unidad estaba pensada para poder tener una abertura lo suficientemente grande para que encajaran con los contenedores del pastillero.

MATERIAL PARA PROYECTOS EXPLICACIÓN CÓDIGO

```
// incluye funcións electronicas
1 // Declaración de librerías
2 #include <avr-libc.h>
3 #include <avr/io.h>
4 #include <LiquidCrystal.h>
5 #include<util/delay.h>
6
7 // Sección setup
8 void setup() {
9     // Configuración de los pines de salida
10    DDRB = 0x00; // Pines de salida
11    PORTB = 0x00; // Pines de salida
12
13    // Sección variables
14    int numero; // el switch principal del vicio
15    int numero2; // el switch secundario que está dentro del caso I del switch de inicio en cada tipo
16    int entrada; // el pin que indica si el botón se ha pulsado
17    int guardando; // para saber en que pantalla estoy
18
19    // Sección variables (matriz)
20    int desordenamiento[10][10]; // para ordenar los datos de cada matriz dentro de los casilleros 0-10x10
21    int desordenamiento2[10][10]; // para ordenar los datos de cada matriz dentro de los casilleros 0-10x10
22    int desordenamiento3[10][10]; // para ordenar los datos de cada matriz dentro de los casilleros 0-10x10
23
24    // Variables para la retroalimentación de los resultados
25    int resultado; // para saber si el resultado es correcto o no
26    int resultado2; // para saber si el resultado es correcto o no
27
28    // Variables para la retroalimentación de los resultados
29    int resultado3; // para saber si el resultado es correcto o no
30    int resultado4; // para saber si el resultado es correcto o no
31
32    // Sección variables
33    int bandera; // bandera que se utiliza en el void seleccionar
34    int bandera2; // bandera que se utiliza en el void seleccionar
35
36    // Sección declaraciones
37}
```

Inicia la “**sección declarar**”, aquí se declara los pines, es donde se conectan todos los componentes que se estén utilizando en el circuito. De igual manera, se declaran todas las variables que se estén utilizando para el código.

```
35 void setup()
36 { // aquí inicia el void setup
37
38    // Inicia pinsMode
39    pinMode(buttonReverse, INPUT);
40    pinMode(buttonSeleccionar, INPUT);
41    pinMode(buttonMoveAtras, INPUT);
42    // Inicia pinsMode
43
44    lcd.begin(16,2);
45    lcd.clear();
46    lcd.backlight();
47    lcd.setCursor(0,0);
48 } // aquí acaba el void setup ////////////////////////
```

En la función “**void setup**”, es donde declaramos o “ponemos” lo que queramos que aparezca siempre que arrancamos el código, en este caso declaramos que siempre usaremos los tres botones del pastillero para entrada, y en la parte de abajo declararemos que siempre se iniciara con una LCD 16x2.



```
52 // Inicia Sección Funciones
53 void entradaMenu()
54 {
55     // Aquí inicia el Void entradaMenu
56     void opciones (String mensaje, int C, int F)
57     {
58         lcd.setCursor(C,F);
59         lcd.print(mensaje);
60     } // Aquí acaba el Void opciones
61
62 void botonMover (int boton,int operaciones)
63 {
64     // Aquí inicia el Void botonMover
65     // Aquí acaba el Void botonMover
66
67 void procedimientosMenu()
68 {
69     // Aquí inicia el Void procedimientosMenu
70     // Aquí acaba el Void procedimientosMenu
71
72 void menus()
73 {
74     // Aquí inicia el Void menus
75     // Aquí acaba el Void menus
76
77 void seleccionar()
78 {
79     // Aquí inicia el Void seleccionar
80     // Aquí acaba el Void seleccionar
81
82 void limites()
83 {
84     // Aquí inicia el Void limites
85     // Aquí acaba el Void limites
86
87 void opciones2 (int mensaje, int C, int F)
88 {
89     // Aquí inicia el Void dosis
90     // Aquí acaba el Void dosis
91
92 void testeoExpotronica (String mensaje, int C, int F) // PROBAR para testeó en la expotronica
93 {
94     // Aquí acaba Sección Funciones
95 }
```

Inicia toda la “**sección de funciones**”, cada una tiene donde inicia y dónde acaba para una mejor organización al momento de leer el código, aquí se utilizan todas las funciones agregadas hasta el momento para el código, cada función tiene una utilidad en el código, el “**void entrada Menú**” se utiliza para la entrada al código, es decir, lo que aparecerá cuando prendamos el pastillero, el “**void opciones**” se utiliza como unas líneas de código (mensaje) universal, ya que ahí mismo tenemos lo que aparecerá siempre que lo declaremos, solamente le cambiamos al mensaje que quiera que aparezca y su respectiva ubicación en la pantalla LCD.

```
95 void botonMover (int boton,int operaciones)
96 {
97     // Aquí inicia el Void botonMover
98     if (digitalRead(boton)==1)
99     {
100         bandera=true;
101         while (bandera == true)
102         {
103             if (digitalRead(boton) == 0)
104             {
105                 bandera=false;
106                 opciones1=opciones1+operaciones;
107                 lcd.clear();
108             }
109         }
110     }
111 } // Aquí acaba el Void botonMover
```

En el “**void botón Mover**” se utiliza igualmente como una función universal, ya que declaramos que con cualquier botón que apretemos se hará una operación, dependiendo el caso, será si avanza entre las opciones o retrocede.

```
138 void menus()
139 {
140     // Aquí inicia el Void menus
141     switch (caso)
142     {
143         case 0: // menú P. Basico
144             entradaMenu();
145             break;
146         case 1: // Inicia cada del Caso principal caso 1 // menú 3
147             switch (opciones)
148             {
149                 case 1: // opción 1
150                     opciones ("Casillero 1",3,0);
151                     break;
152                 case 2: // opción 2
153                     opciones ("Casillero 2",3,0);
154                     break;
155                 case 3: // opción 3
156                     opciones ("Casillero 3",3,0);
157                     break;
158                 case 4: // opción 4
159                     opciones ("Casillero 4",3,0);
160                     break;
161                 case 5: // opción 5
162                     opciones ("Salir",3,0);
163                     break;
164             }
165         break; // Acaba break de menú casilleros
166     }
167 }
```

El “**void menus**” es donde tiene todo lo principal del pastillero, este se mueve por un switch general (menú principal si lo llamamos de otra manera), y este tiene switch general tiene cierto número de “case”, cada case, que sería como una opción, tienen otro “sub-switch” es decir, las opciones que lleva, como se muestra en la imagen, aparecen los 4 casilleros que se van a utilizar, con sus respectivas funciones de cada uno y su opción de salir para regresar el mensaje inicial de cuando inicias el casillero

```
170 void seleccionar()
171 {
172     // Aquí inicia el Void seleccionar
173     if (digitalRead(botonSeleccionar)==1)
174     {
175         banderaBotonSelec=true;
176         while (banderaBoton == true)
177         {
178             if (digitalRead(boton)==1)
179             {
180                 banderaBoton=false;
181             }
182             else
183             {
184                 caso = digitalRead(boton); // Selecciona el menú que se está usando
185                 menu1=0; // Declara que se estrena en el menú 1 una vez se aprieta el botón
186                 break;
187             }
188         }
189     }
190 }
```

Inicia “**void seleccionar**” aquí es donde se le “asigna” una opción a cada uno de los casilleros, como todo se conforma de un switch general, que serían como menús generales, es decir los principales, y cada sub-switch dentro de uno general es como un sub-menú. En la imagen se aprecia



como el switch 1 (switch principal, o menú principal 1) tiene otro switch y cada case de los switch tiene un menú=x, esto porque en void menus cada switch general te guía hacia un apartado, aquí lo que hacemos es que se lo asignamos, es decir, case 1 es menú=2, donde iría el casillero 2 con sus respectivas opciones, y así con cada apartado.

Inicia “**void límites**”, aquí le asignamos al apartado que queramos el “límite” para que no se pase de opciones, es decir, si casillero 1 tiene 3 apartados, y le da 4 veces al botón de avanzar, que no se pase de las opciones y le deje la pantalla en blanco, entonces se le asigna limites, por si llega a avanzar más de las opciones establecidas que le regrese a la primera opción, y de igual manera de forma contraria que si esta en la opción una y le da a retroceder, que no le aparezca la pantalla en blanco, si no que se quede en la opción 1.

Aquí están 3 funciones “void opciones2”, “void testeo Expotrónica” y “void loop”,
En “void opciones 2” es como void opciones 1, solo
que aquí en vez de escribir un mensaje, escribo una

variable, para que se me facilite al momento de programar algo y ocupe menos líneas de código.

“void testeo Expotrónica”, solo nos sirve para cuando se muestre en la presentación, sea de manera eficaz y rápida con un tiempo de espera mucho menor, es decir, si en un casillero es cada 12 horas una pastilla, aquí para mostrarles el ejemplo lo ponemos cada 10 segundos.

El “void loop” es lo que se repetirá un ciclo indeterminado de veces, aquí es lo que queremos que se repite hasta que le digamos que ya no lo haga, aquí solo tenemos 3 funciones muy útiles, que es la de procedimientos, que solo es lo que indica si los botones avanza o retrocede, la opción de los límites para que jamás se pase de las opciones que queramos, y la función de menús, que es donde aparecen todas las opciones del casillero.

La idea principal desde la cual partió la realización de este proyecto fue construir un pastillero electrónico que supusiera una mejora una adherencia a la medicación, con el fin de lograr una mejora significativa en el seguimiento reglamentario de la prescripción médica, en un paciente con historial de problemas cardíaco, A lo largo del desarrollo del prototipo y del código de programación del mismo, se presentaron distintas situaciones que obligaron cambios en el diseño del pastillero y al público al cual estaba dirigido. Estos cambios supusieron un retraso en el plan de trabajo del proyecto y un gasto superior al propuesto puesto a que debimos conseguir materiales que no estaban contemplados en el presupuesto. Cabe mencionar que a pesar de los contratiempos generados, las actualizaciones realizadas al proyecto permitieron que este se lograra terminar de manera más oportuna, adaptándose a las capacidades y conocimientos



adquiridos durante el semestre en curso. Después de discutir como equipo y con nuestros profesores las implicaciones de estos cambios, tomamos la decisión de dirigir nuestro pastillero a una persona en específico.

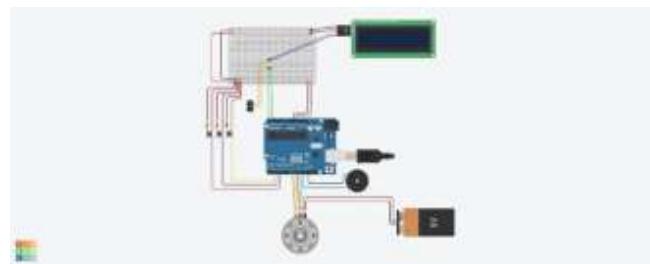
El pastillero electrónico cumplió con los objetivos establecidos. Se destaca que se logró diseñar un código de programación que controla al pastillero electrónico con todas las especificaciones establecidas y se logró diseñar y construir un pastillero electrónico funcional y fácil de manejar. No obstante, hubieron problemas que se pudieron haber evitado de haber previsto variables de tiempo y especificaciones que no se tomaron en cuenta. Para proyectos futuros, se espera poder contemplar todas las partes y solicitudes del proyecto antes de comenzar con el diseño y construcción del prototipo.

IMAGEN	NOMBRE
	Pila de 9V (Simulando el toma corriente)
	Buzzer
	Botón
	Arduino Uno

	Pantalla LCD con modulo I2C
	Motor paso a paso con microcontrolador
	Placa de pruebas (simulando la placa PCB)
	Temporizador (Simulando el módulo de horario RTC DS1307)

Materiales sustituidos en el modelo grafico:

Imagen	Nombre
	tomacorriente (simulado con la pila de 9V)
	Modulo DS1307 con I2C (simulado con el temporizador)
	PCB (simulado por la placa de pruebas)





En la figura que se muestra podemos apreciar el boceto del circuito en el cual se distinguen sus componentes, podemos empezar con el ARDUINO UNO, que será el cerebro de toda nuestra programación para los circuitos, al igual se presenta una PANTALLA LCD que contará con el MÓDULO I2C que permite liberar espacio tanto del Arduino como en la conexión del circuito (se refleja en la parte de atrás de la pantalla por lo que no se ve) y esta pantalla se mostrará el menú principal, la hora local, las dosis que deberá tomar el paciente, la medicina que se le administra, etc. Igual tiene 3 BOTONES que son SELECCIONAR, RETROCEDER Y AVANZAR, para que nuestro usuario pueda indagar dentro de este menú, tenemos también el BUZZER que será la alarma por la cual el propio pastillero emitirá un sonido el cual le recordará al paciente cuando es la hora de tomar la pastilla, tenemos el motor paso a paso con su microcontrolador el cual conforme a sus RPM (revoluciones por minuto) previamente ya configurados con el equipo, dará una serie de giros calculados para que quede exactamente en el casillero que se desea dentro del pastillero, este igual regresara a su punto cero que será el casillero vacío que no contendrá nada para no maltratar o humedecer las pastillas el paciente, contamos igual dentro de la figura con el MODULO DE HORARIO (en la figura representado por un contador 666) el cual nos ayuda a tener la hora local dentro del pastillero sin necesidad de configurar el Arduino para que lo tenga, este módulo modelo RTC DS1307 con I2C cuenta con una memoria interna la cual guarda la hora aunque esté apagado el pastillero para que cuando se vuelva a encender para que no se tenga que configurar cada que se apague el pastillero y finalmente en la figura tenemos una PILA DE 9V que sirve como referencia de que nosotros conectaremos a una toma de corriente nuestro circuito, ya que el Arduino uno por si solo no tiene el voltaje necesario para distribuir la energía necesaria para que todo el circuito cuente con la fuerza

de voltaje sin necesidad de quemar el Arduino uno por la cantidad excesiva de voltaje que requiere este circuito, todo esto se hará con un eliminador de corriente que tendremos conectado a la toma de corriente, no obstante se tiene que señalar que la toma de corriente se usará exclusivamente para el motor paso a paso, ya que se puede conectar mediante el eliminador y usara la energía que sea necesaria para que pueda girar sin necesidad de forzar al Arduino a llevar toda la energía al motor y así la pantalla LCD y los botones solo estarán conectado al Arduino y no sobrepasaremos el límite de este.

Hemos pasado por una serie de configuraciones dentro de nuestro circuito que conforme el paso del tiempo hemos implementado y quitado para el correcto funcionamiento de nuestro circuito ya que hemos pasado por una serie de problemas cuando quisimos conectar como queríamos, nos dimos cuenta de nuestros errores por medio de los profesores que nos instruyeron antes de hacer nuestra conexión, también tuvimos una serie de dudas cuando queríamos ver la manera de que se pueda conectar a una toma de corriente, ya que no queríamos quemar nuestros componentes, también cuando se presenta la idea de meter todo en una placa PCB, porque se tenía planeado el circuito para que esté abajo del pastillero y que este armado mediante conexiones y extensores de cable para que todo quede bien distribuido y menos junto dentro del pastillero, ya que se pensó para que cuando el técnico (en este caso nosotros) quiera arreglar cierta parte del circuito lo abra y se encuentre con un circuito mas claro y simple de nuestra conexión para que sea más eficiente al momento de reparar o hacer algún cambio dentro de este.

La última observación que tuvimos por parte de nuestro profesor es que el módulo de horario y nuestra pantalla LCD compartían los mismos pines análogos dentro de nuestro Arduino y tuvimos que ver la forma de que se pueda compartir los pines sin necesidad de implementar un segundo Arduino solo por dos pines, lo que hicimos es



que dentro de nuestra placa de pruebas conectamos esos pines y en la misma línea de este conectamos tanto el módulo como la pantalla y así resolvimos el último problema que tuvimos dentro de nuestra conexión (como se observa dentro el modelo gráfico)

Resultados.

Para visualizar nuestros resultados podemos comenzar viendo nuestro objetivo general el cual es construir un pastillero electrónico que mejore la adherencia a la medicación., tomando en cuenta esto podemos decir que si cumplimos con nuestro objetivo, aunque este objetivo es bastante general por lo que si queremos visualizarlos a mayor detalle usaremos nuestros objetivos específicos.

En cuanto a la construcción en general podemos observar que nuestro ensamblaje llevó algunas modificaciones que se realizaron a mano en la universidad, algunas con ayuda de las máquinas disponibles en el área de mecatrónica, también se pintó el pastillero con el fin de darle un mejor aspecto del que tenía, escogimos los colores negro y azul.

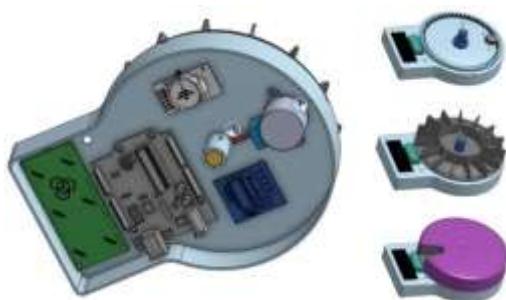


Fig.#Diseño del pastillero:se muestra el diseño sin cambio de color.

Evidencia del diseño ya impreso:

<https://drive.google.com/file/d/1s8k4D0Rq5qiPDKmr5uk3Z5OSu4kAp9h6/view?usp=drivesdk>

Nuestro primer objetivo es realizar un código de programación funcional que se adecue a los requerimientos del pastillero, este se cumple en su totalidad ya que es de vital importancia para el funcionamiento óptimo de nuestro pastillero, sin este no hay proyecto, este código tuvo diferentes versiones y se fue actualizando conforme al asesoramiento de nuestro profesor de programación el cual nos recomendó varias herramientas para hacer nuestro código más eficiente y corto.

Evidencia del código menú en funcionamiento:

https://drive.google.com/file/d/181vtrbgZ9q_IUO8-3D9UuYaT9-Hx2N_1/view?usp=drivesdk

Para nuestro segundo objetivo el cual es diseñar un modelo de pastillero electrónico ergonómico y funcional podemos encontrar que este si se cumplió ya que nuestro diseño es de fácil uso, pequeño y solo requiere de conectarlo a la corriente para que comience con su función.

El proceso de decisión de diseño para nuestro proyecto no llevó mucha dificultad ya que encontramos herramientas como el software Onshape que nos permite modelar a 3D diversos proyectos, para nuestro diseño encontramos plantillas que fueron modificadas a conveniencia del diseño ya plasmado. Este se mandó a imprimir cada pieza por separado con el profesor del taller de robótica de la universidad, se obtuvieron aproximadamente 6 piezas independientes que conforman nuestro diseño, las cuales son:

- Engranaje pequeño
- Engranaje grande
- Tapa



- Estructura de casilleros
- Cobertura del tornillo
- Base

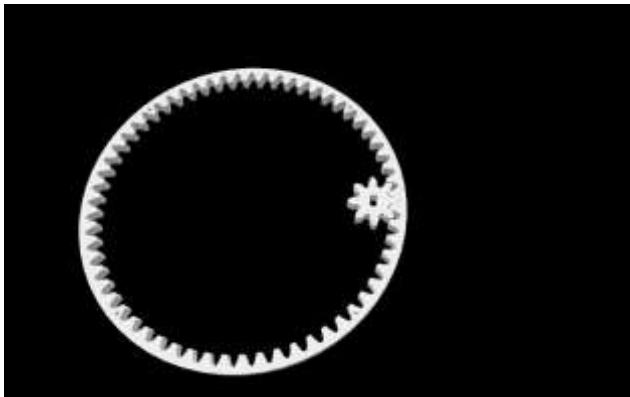


Fig.# Engranaje pequeño y grande: se muestra la forma en la que se representó el diseño nuestros engranajes.



Fig.# Tapa: diseño 3D de la tapa con abertura por la que se obtienen las pastillas.

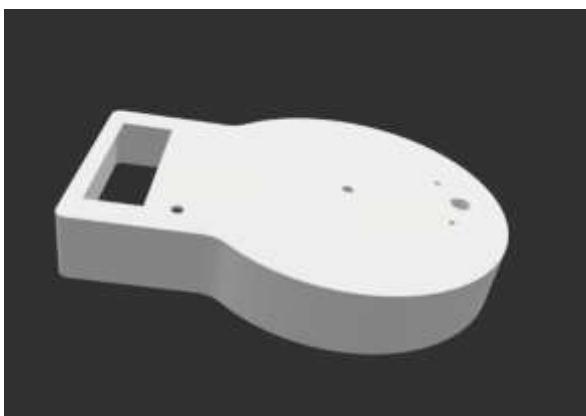


Fig.# Base: diseño 3D de la base con abertura para la lcd y pequeños huecos para el motor y tornillo.

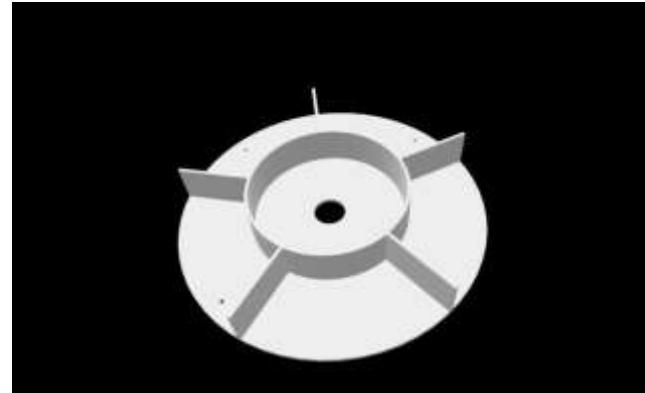


Fig.# Estructura con casilleros: diseño 3D de la estructura en donde se pondrán las pastillas.

Nuestro tercer objetivo es construir un prototipo de pastillero electrónico sencillo, fácil de usar y distinto a los comerciales, este objetivo también se cumple ya que como mencionamos arriba nuestro diseño es de fácil uso, solo se debe conectar a la corriente y manipular a su conveniencia a través de botones que despliegan el menú correspondiente al pastillero.

Es sencillo ya que por sus dimensiones se puede transportar con facilidad, es de un material sólido que le da soporte a toda la estructura y a su vez es distinto ya que no es un diseño muy común como puede verse reflejado en el mercado que los comercializa.

Este pastillero funciona de manera que a la hora que se deba tomar sus pastillas la cuál ya viene preconfigurada nuestro pastillero rotará los grados suficientes como para revelar o dejar libre el casillero que contiene las pastillas para que el usuario solo las agarre y se las pueda tomar.

Evidencia del funcionamiento de nuestro pastillero:

<https://drive.google.com/file/d/1s8k4D0Rq5qiPDKmr5uk3Z5OSu4kAp9h6/view?usp=drivesdk>

Para nuestro cuarto objetivo podemos encontrar proponer una opción de seguimiento de satisfacción en su tratamiento médico impulsado por el pastillero, principalmente a usuarios con historial cardíaco. Este



objetivo se plasmó con el fin de conocer el desempeño de nuestro pastillero, lo podemos visualizar a través de nuestra encuesta realizada en un forms que se entrega o proporciona al usuario, por el momento no contamos con repuestas ya que este pastillero no se comercializará, sin embargo creemos que es importante recibir feedback de nuestro prototipo para enmarcar las áreas de mejora, podemos encontrar que si cumplimos con este objetivo ya que le brindamos a nuestro usuario una manera de calificar su satisfacción con el prototipo.

Este formulario solo cuenta con muy pocas preguntas que en su mayoría son abiertas para que el usuario pueda expresarse libremente y hacer que su experiencia con el pastillero electrónico.

Formulario de satisfacción:

<https://forms.gle/ZWghsZ9tYhMcMo9eA>

Dejando un poco de lado el recuento de nuestros objetivos también podemos encontrar como resultado el correcto funcionamiento de nuestro circuito el cuál se fue ajustó según las necesidades del pastillero, este fue diseñado en tinkercad. Cuenta con un buzzer, nuestros botones y los componentes necesarios para la correcta funcionalidad del circuito.

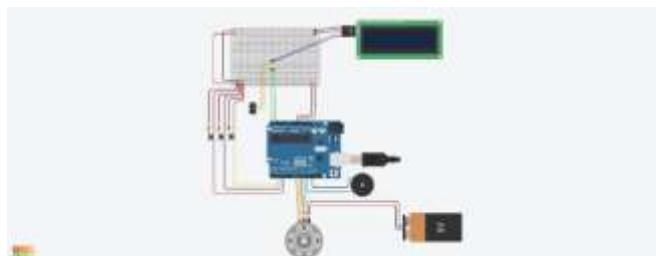


Fig.# Circuito en tinkercad: se muestran los componentes y sus debidas conexiones al protoboard y arduino.

Conclusión

La idea principal desde la cual partió la realización de este proyecto fue construir un pastillero electrónico que supusiera una mejora una adherencia a la medicación, con el fin de lograr una mejora significativa en el seguimiento reglamentario de la prescripción médica, en un paciente con historial de problemas cardíaco, A lo largo del desarrollo del prototipo y del código de programación del mismo, se presentaron distintas situaciones que obligaron cambios en el diseño del pastillero y al público al cual estaba dirigido. Estos cambios supusieron un retraso en el plan de trabajo del proyecto y un gasto superior al propuesto puesto a que debimos conseguir materiales que no estaban contemplados en el presupuesto. Cabe mencionar que, a pesar de los contratiempos generados, las actualizaciones realizadas al proyecto permitieron que este se lograra terminar de manera más oportuna, adaptándose a las capacidades y conocimientos adquiridos durante el semestre en curso. Después de discutir como equipo y con nuestros profesores las implicaciones de estos cambios, tomamos la decisión de dirigir nuestro pastillero a una persona en específico.

El pastillero electrónico cumplió con los objetivos establecidos. Se destaca que se logró diseñar un código de programación que controla al pastillero electrónico con todas las especificaciones establecidas y se logró diseñar y construir un pastillero electrónico funcional y fácil de manejar. No obstante, hubo problemas que se pudieron



haber evitado de haber previsto variables de tiempo y especificaciones que no se tomaron en cuenta. Para proyectos futuros, se espera poder contemplar todas las partes y solicitudes del proyecto antes de comenzar con el diseño y construcción del prototipo.

Perspectivas.

Se espera que, a futuro, se puedan implementar mejoras al diseño y regresar a las características iniciales desacertadas del modelo inicial. De igual forma, partiendo de nuestro prototipo y expandiendo el código de programación, lograr mejorar el pastillero e implementar mejoras estéticas y de sistema. Las mejoras propuestas se enlistan a continuación:

- Practicidad. Presentar un diseño más pequeño y fácil de llevar, con una batería portable para que no tenga la necesidad de ser conectado a una toma de corriente.
- Bluetooth. Se propone agregar un módulo bluetooth que permita recibir todos los datos de dosis del usuario a un dispositivo móvil y del mismo modo, programar los horarios de toma de las pastillas de manera remota. También, permitirá llevar un registro de cuantas veces el usuario ha tomado sus prescripciones, cuando las ha tomado fuera del horario establecido y cuando no las ha tomado.
- Sensores. Agregar sensores en cada casillero del pastillero que permitan detectar cuando ya no quedan pastillas y que envíe una notificación al teléfono del usuario para recordarle que debe rellenar el pastillero.
- Capacidad. Agregar más casilleros al pastillero para que pueda ser utilizado por personas con



prescripciones médicas más grandes o que toman una mayor cantidad de fármacos al día.

Conclusiones individuales.

Ana Cristina Castro Canchola:

En conclusión, este proyecto mejoró muchas de mis habilidades como la comunicación, gestión de tiempos, así como el trabajo de equipo y delegación de tareas. Así como mejoré varias de mis habilidades también adquirí nuevos conocimientos que en conjunto de algunos profesores se volvieron fundamentales para el éxito y realización del proyecto, me encantó descubrir y aprender nuevas cosas.

Creo que es fundamental que empleemos estas herramientas en nuestra carrera académica ya que nos aportan una gran base y preparación para desafíos que se pueden presentar más adelante, también viéndolo de lado más social es posible ayudar o generar cierto impacto en la salud de la sociedad, con este pastillero ayudamos hasta cierto punto a la adherencia de un paciente en específico, tomando en cuenta nuestros objetivos creo que logramos el fin de nuestro proyecto de una manera más calmada y organizada, sin embargo como en cualquier caso hay cosas que podemos mejorar teniendo en cuenta nuestras experiencias.

Mario Tadeo Caballero Córdova

Este proyecto en lo personal siento que me ayudó demasiado en varias maneras, ya que tuve que aprender varias cosas sobre programación para poder elaborarlo, ya que en lo personal, yo inicie a trabajar este proyecto con un conocimiento casi nulo de programación, y a medida que íbamos avanzando en este proyecto aprendí a mejorar mi pensamiento lógico y a comprender mejor este área de la programación, al igual que sacar medidas exactas y utilizar

aplicaciones nuevas que me ayudaran en la carrera, como sería tinkercad, que es una aplicación demasiado útil donde puedo ver diagramas de circuitos y comprenderlos mejor, o kidcad, que esto sirve para hacer PCBs, que es lo que más se usa en el día a día que son demasiado útiles, en resumen, me gustó mucho el proyecto aunque también sufrí estrés y desesperación por la necesidad de llegar a soluciones en momento donde no tienes cabeza para pensar, pero eso te ayuda a practicar y ser mejor para un futuro.

Itzel del Carmen Cabrera Rodríguez:

Considero que el desarrollo de este proyecto fue un proceso de grandes aprendizajes en cuanto a programación, diseño de circuitos, diseño de prototipos, impresión 3D, diseño y elaboración de placas de circuito impreso (PCB), metodología de un proyecto y múltiples temas más. Como equipo, tuvimos algunas dificultades para organizarnos, dividir las actividades y seguir el plan del trabajo. Sin embargo, logramos resolverlos y coordinar las partes del proyecto. En cuanto a la metodología Modelo D+, pudimos desempeñarnos de manera satisfactoria en cada uno de los puntos de control, aunque de haber destinado más tiempo para la elaboración del prototipo, hubiéramos tenido mayor margen de error y espacio para más pruebas y para implementar mejoras. El objetivo general del proyecto se alcanzó, pero el resultado pudo haber sido mejor.

Gabriela Jerez Gómez:

Se ha concluido este proyecto con la realización de este pastillero electrónico en lo cual tuvimos que aprender mucho, en la cual ha sido cansado, sin embargo, fue demasiado enriquecedor poder construir algo lo cual en mi experiencia nunca había realizado, me siento muy feliz de poder haber concluido este semestre con este proyecto en el que aprendí a trabajar en equipo, en el que di un poco



más de mi para adaptarme al trabajo en equipo, en el que investigue demasiado, estoy agradecida con todos los De S maestros que nos apoyaron con el proceso de este proyecto, tales como el maestro de electrónica, Rodrigo Solís, que nos ayudó con la construcción de nuestra pcb, con el maestro Rutilio Nava que nos ayudó con el código de programación, nos dio tips de cómo mejorar nuestro diseño y donde conseguir contactos para la impresión 3D, al igual de donde podíamos conseguir todos los componentes necesarios que no sabíamos que nos facilitaban el trabajo, a la maestra Patricia Yolanda por guiarnos en todo momento y estar abierta a ayudarnos en todo momento, este fue un trabajo duro de todo el equipos por el compromiso que tenemos con poder seguir adelante como profesionales, de apoyo, de regaños y de andar detrás de los compañeros, espero en el próximo proyecto, se pueda ver con más tiempo y tener una mejor organización, porque en este no se pudo terminar todo con anticipación y eso era lo óptimo para todos.

Pedro Iván Pérez Rosas:

Para mí el pastillero ha sido un recorrido bastante amplio y con muchos tropiezos ya que no se contaban con todos los conocimientos necesarios para poder recrear el pastillero, pero de los errores se aprenden bastante y nosotros recorrimos demasiado camino viendo como hacer que nuestro pastillero. completamente funcional, implementamos cosas que nosotros no sabíamos que existían para poder meter en nuestro pastillero, para mi el documentar todo nuestro proyecto fue lo más difícil ya que no soy fan de documentar cada paso que hacíamos con el pastillero, con mi equipo no tuve ningún roce y creo que congeniamos bastante bien al momento de trabajar, supimos como comunicarnos y cuando actuar, hubo demasiada conexión como equipo y creo que por eso nuestro pastillero quedó exitosamente funcional, como cualquier proyecto tuvo fallas y mejoras, pero para mi es un logro completamente decir que fui parte de este proyecto.

Referencias.

alud, S. (s. f.). 490. *Cada año, 220 mil personas fallecen debido a enfermedades del corazón.* gob.mx.
<https://www.gob.mx/salud/prensa/490-cada-ano-220-mil-personas-fallecen-debido-a-enfermedades-del-corazon#:~:text=En%20M%C3%A9xico%2C%20cerca%20de%202020,elevado%20y%20diabetes%20no%20controlada>.