

PROGRAMA EN APP INVENTOR PARA REALIZAR LA CONVERSIÓN DE UN NÚMERO DECIMAL A CÓDIGO BCD APLICABLE A UN DISPLAY DE SIETE SEGMENTOS DE ÁNODO COMÚN

1. PLANTEAMIENTO DEL PROBLEMA

Se desconoce la interfaz de funcionamiento, control y conversión de un número decimal a código BCD en App Inventor para un display de siete segmentos, para lo cual se formularon las siguientes preguntas:

- ¿Cómo se convierte un número decimal a código BCD?
- ¿Cuál es el lenguaje de programación utilizado en App Inventor?
- ¿Cómo funciona un display de siete segmentos?

2. OBJETIVOS

Objetivo general

Realizar la conversión de un número decimal a BCM en App Inventor aplicable a un display de siete segmentos.

Objetivos específicos

- Comprender la estructura de control básica de App Inventor para la conversión de un número decimal a código BCD.
- Identificar los principales parámetros de funcionamiento de un display de siete segmentos.
- Implementar condicionales anidados en el diseño del código.

3. ESTADO DEL ARTE

En 2019, Thilanka Munasinghe, Evan W. Patton y Oshani Seneviratne del Instituto Politécnico Rensselaer, Departamento de Tecnologías de la Información y Ciencia Cibernética en Estados Unidos, desarrollaron aplicaciones IoT usando MIT App Inventor para recopilar y analizar datos de sensores de bajo costo, algoritmos de aprendizaje que han permitido la toma de decisiones basadas en sistemas a gran escala. La plataforma de desarrollo de MIT App Inventor permite crear aplicaciones utilizando varios sensores y plataformas de IoT como Raspberry Pi 2 y Android Things fáciles de programar con un enfoque en la recopilación de datos de sensores conectados en un solo ambiente (Munasinghe, 2019, p.1) [1].

Ruanqianqian Huang y Franklyn Turbak del Wellesley College, Department of Computer Science en 2019 implementaron un diseño para la conversión bidireccional entre bloques y texto para App Inventor mediante fragmentos de código visual que evitan errores semánticos sintácticos y estáticos y reducen la carga cognitiva para este fin diseñaron un sistema de modo dual para MIT App Inventor que admite representaciones textuales para bloques, espacios de trabajo, pantallas y proyectos completos que permiten la conversión bidireccional entre bloques isomórficos y representaciones de texto, permitiendo que individuos de varios niveles de

experiencia en programación se relacionen con la interfaz (R. Huang,F. Turbak,2019)[2].

Lance A. Allison y Mohammad Muztaba Fuad del Departamento de Ciencias de la Computación de la Universidad Estatal de Salem en 2016, basaron su estudio en la comunicación entre aplicaciones de Android desarrolladas en App-Inventor y Android

Studio para proporcionar diferentes funcionalidades específicas de cada plataforma al dispositivo, en Android pueden desarrollarse usando App Inventor 2 (AI) del MIT o usando IDEs como Android Studio (AS) con la ayuda del SDK de Android, la versión del App Inventor original de Google, que es un entorno de desarrollo basado en un navegador web para una forma más sencilla de desarrollar aplicaciones de Android. Con poco o ningún conocimiento de programación, uno puede desarrollar e implementar una aplicación de Android usando AI, la mayoría de las aplicaciones desarrolladas en AI tienen que seguir una plantilla de diseño específica (Lance,2006,p.1) [3].

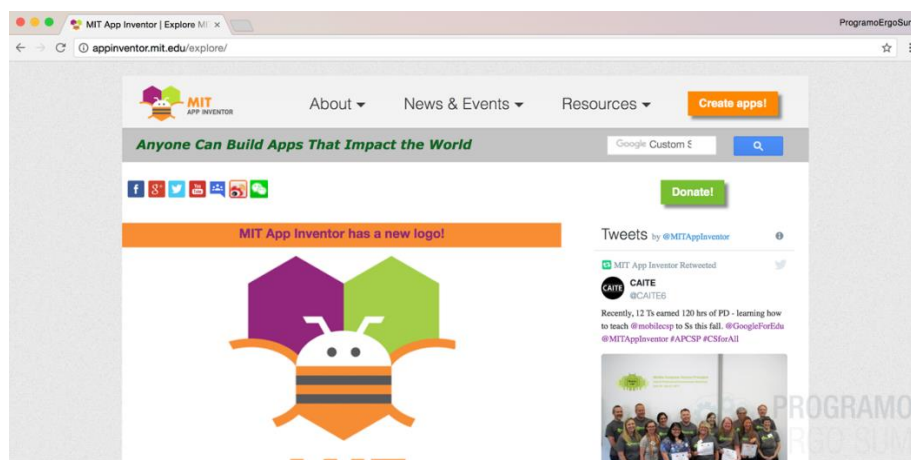
Para el trabajo de investigación presente, se utilizó el entorno de desarrollo de software MIT App Inventor dirigido a la Conversión de un "NÚMERO DECIMAL A CÓDIGO BCD" (Munasinghe,2019,p.1), el entorno de programación para el aplicativo se hizo sobre la base de MIT App Inventor (Lance,2006,p.1). El sistema tiene la capacidad de proyectar un número decimal de salida en la pantalla del dispositivo android simulando la proyección de un display de 7 segmentos (R. Huang,F. Turbak,2019).

4.MARCO TEÓRICO

APP INVENTOR

App Inventor es un entorno de desarrollo de software creado por Google para la elaboración de aplicaciones destinadas al sistema operativo de Android. El lenguaje es gratuito y se puede acceder 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.



Interfaz de App Inventor

En esta nueva entrada os mostramos cómo está diseñada la interfaz de la plataforma App Inventor que

nos permitirá crear nuestras propias aplicaciones para dispositivos móviles.

Los principales pasos que hemos de dar antes de arrancar con nuestra creación son:

pensar cuál es el objetivo que queremos alcanzar con nuestra aplicación.

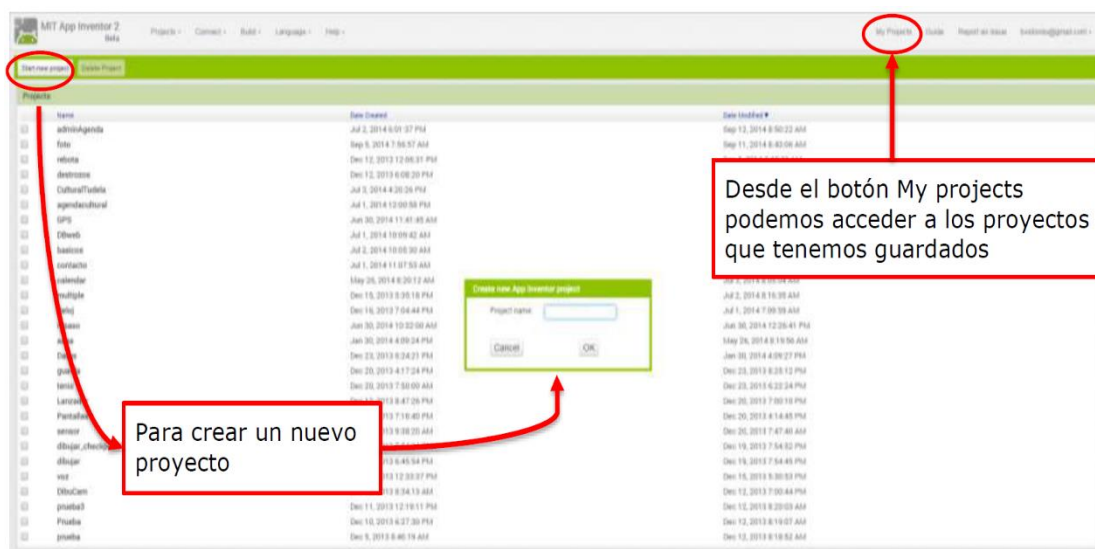
tener claro cuál es el diseño que queremos darle a esta nueva App, para lo que podemos usar, por ejemplo, un boceto en papel del diseño deseado. Para poder acceder al servicio se necesita una cuenta de Google.

Para conseguir esto, App Inventor divide el desarrollo en tres fases:

- Gestor de proyectos: Donde se guardan los proyectos.
- Diseñador: Donde se diseña la interfaz de las aplicaciones.
- Editor de bloques: Donde se programa las acciones que se realizarán con la interfaz anteriormente diseñada.

Mis Proyectos (My Projects)

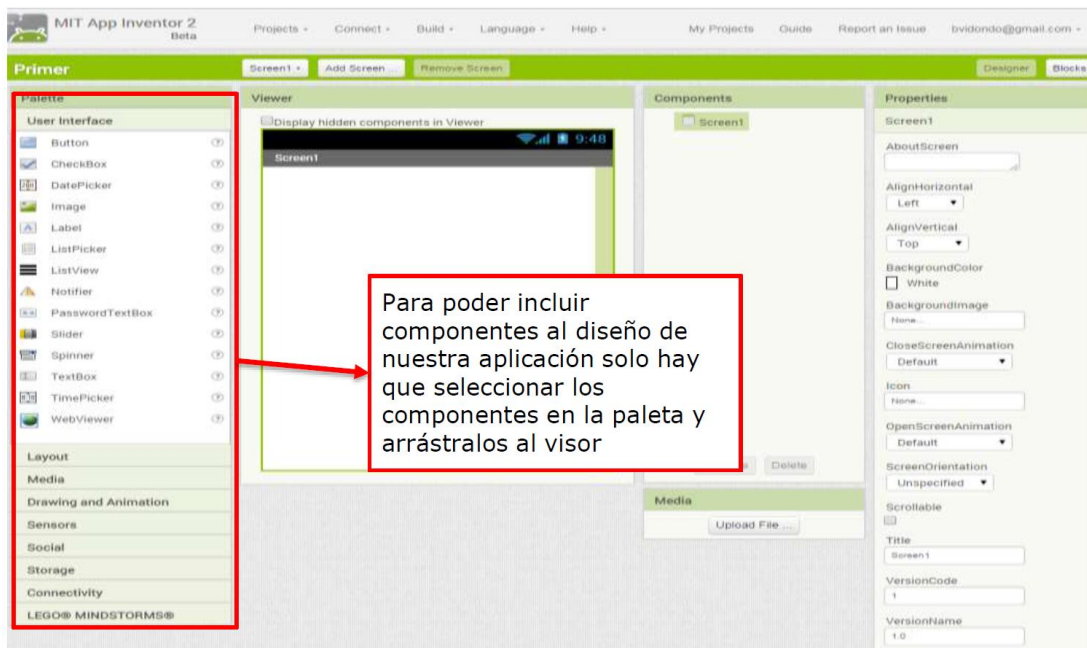
La vista de los proyectos es la primera que se carga cuando se accede a App Inventor, en esta herramienta se puede realizar un seguimiento de todos nuestros proyectos



Diseñador

(Designer)

La herramienta de diseño permite seleccionar los componentes de la App y definir el interfaz de usuario de la misma (botones, cuadros de texto, etiquetas, etc.)

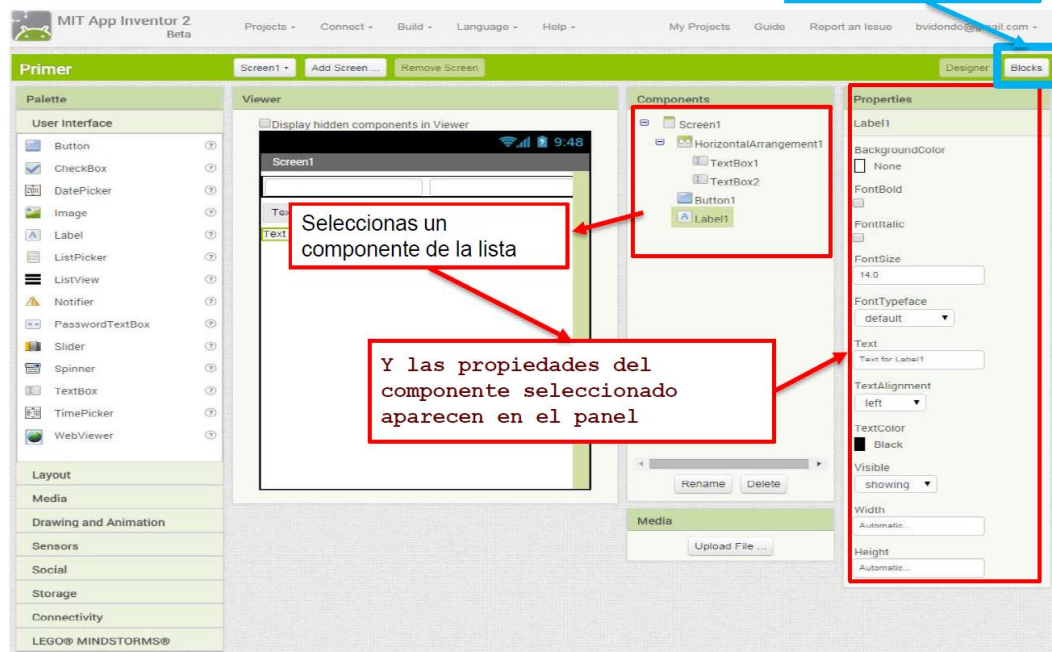


Diseñador (Designer)

La herramienta de diseño permite seleccionar los componentes de una aplicación y definir el entorno de usuario de la misma.

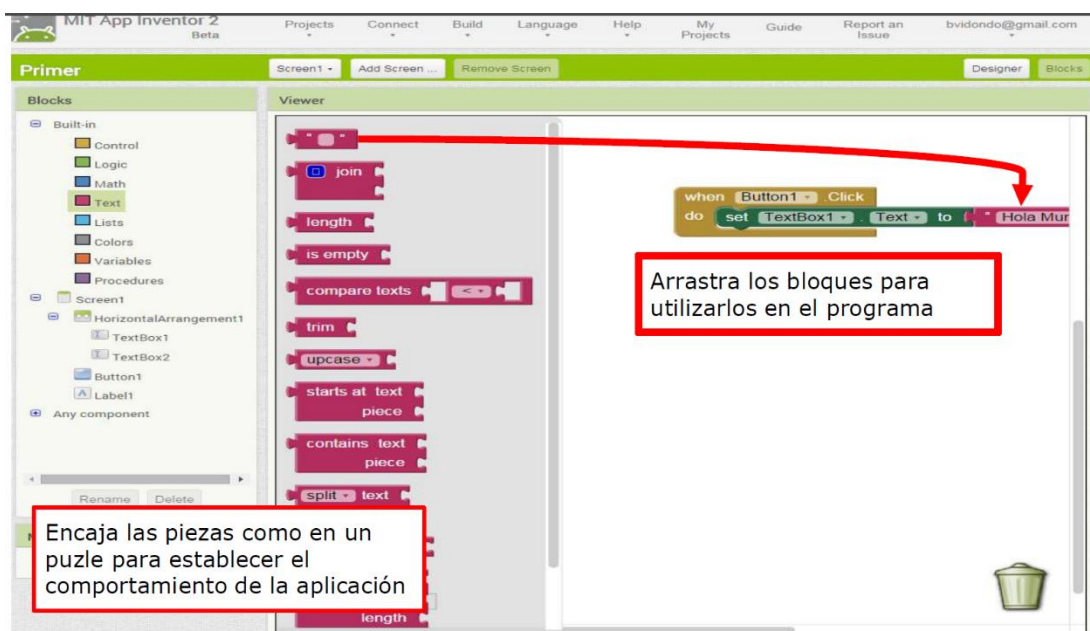


Diseñador (Designer)



Editor de Bloques (Blocks)

El comportamiento de la aplicación se programa mediante bloques en el editor de bloques.





Características y funciones

- El editor de bloques de la plataforma App Inventor, utilizaba anteriormente la librería Open Blocks. Basado en [http://Blockly] de JavaScript para crear un lenguaje visual. Estas librerías están distribuidas por el 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.
- 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.
- 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.

Ventajas

- Se pueden crear aplicaciones por medio de bloques de manera intuitiva y gráfica, sin necesidad de saber código de programación.
- Se puede acceder en cualquier momento y cualquier lugar siempre que estemos conectados a internet.
- Nos ofrece varias formas de conectividad: directa, o wi fi o por medio del emulador.
- Nos permite descargar la aplicación mediante el .apk a nuestro pc.

Inconvenientes

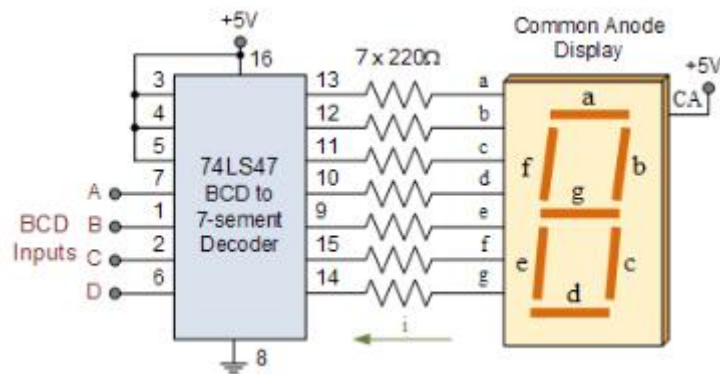
- No genera código Java para desarrollos más profundos.
- Solo se puede desarrollar para Android.

SISTEMA DE NUMERACIÓN DECIMAL CODIFICADO EN BINARIO O BCD

El decimal codificado en binario, o BCD, es otro proceso para convertir números decimales en sus equivalentes binarios, hay muchos códigos binarios diferentes utilizados en circuitos digitales y electrónicos, cada uno con su propio uso específico.

Naturalmente se vive en un mundo decimal (base-10) y de alguna forma se necesita convertir estos números decimales en un entorno binario (base-2) que las computadoras y los dispositivos electrónicos digitales entiendan, y el código decimal codificado en binario permite hacerlo. Un código binario de n bits es un grupo de " n " bits que asumen hasta 2^n combinaciones distintas de 1 y 0. La ventaja del sistema decimal codificado en binario es que cada dígito decimal está representado por un grupo de 4 dígitos binarios o bits de forma muy similar a la de Hexadecimal. Así que para los 10 dígitos decimales (0 a 9) se necesita un código binario de 4 bits.

Los números decimales codificados en binario se detienen en 9 binario 1001₂. Esto significa que, aunque se pueden representar 16 números (2^4) con cuatro dígitos binarios, en el sistema de numeración BCD las seis combinaciones de códigos binarios de: 1010 (decimal 10), 1011 (decimal 11), 1100 (decimal 12), 1101 (decimal 13), 1110 (decimal 14) y 1111 (decimal 15) se clasifican como números prohibidos y no se pueden utilizar.



La principal ventaja del código decimal codificado en binario es que permite la conversión fácil entre el formato decimal (base-10) y binario (base-2). Sin embargo, la desventaja es que el código BCD es inútil ya que los estados entre 1010 (decimal 10) y 1111 (decimal 15) no se utilizan. Sin embargo, el decimal codificado en binario tiene muchas aplicaciones importantes, especialmente el uso de pantallas digitales.

En el sistema de numeración BCD, un número decimal se separa en cuatro bits por cada dígito decimal dentro del número. Cada dígito decimal está representado por su valor binario ponderado que realiza una traducción directa del número. Por lo tanto, un grupo de 4 bits representa cada dígito decimal visualizado desde 0000 para un cero hasta 1001 para un nueve, por ejemplo, 357_{10} (trescientos cincuenta y siete) en decimal se presentarían en decimal codificado en binario como:

$357_{10} = 0011\ 0101\ 0111$ (BCD)

Se puede ver que BCD usa codificación ponderada, porque el bit binario de cada grupo de 4 bits representa un peso dado del valor final. En otras palabras, el BCD es un código ponderado y los pesos utilizados en el código decimal codificado en binario son 8 , 4 , 2 , 1 , comúnmente llamado el código 8421, ya que forma la representación binaria de 4 bits del dígito decimal relevante.

Representación decimal codificada en binario de un número decimal

Poder binario	2^3	2^2	2^1	2^0
Peso binario:	8	4	2	1

El peso decimal de cada dígito decimal a la izquierda aumenta en un factor de 10. En el sistema numérico BCD, el peso binario de cada dígito aumenta en un factor de 2 como se muestra. Luego, el primer dígito tiene un peso de 1 (2⁰), el segundo dígito tiene un peso de 2 (2¹), el tercero un peso de 4 (2²), el cuarto un peso de 8 (2³).

El código binario puro se basa en una regla (potencias de 2), mientras que los códigos BCD se basan en una tabla en la que salen los números decimales del 0 al 9 y las correspondientes “traducciones” en BCD. Hay varios tipos de BCD:

- BCD puro: Binary Code Digit, decimal o decimal codificado en binario es un estándar para representar números decimales en el sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits.
- BCDS XS3: La conversión se produce sumando 3 unidades al decimal que queramos transformar en binario.
- Aiken: Código similar al código BCD natural con los "pesos" o "valores" distribuidos de manera diferente. En el código BCD natural, los pesos son: 8-4-2-1, en el código Aiken la distribución es: 2-4-2-1.

Si los anteriores códigos se basaban en una regla, éstos se basan en una tabla. El método consiste en sustituir cada dígito decimal por los cuatro bits correspondientes.

decimal	BCD puro	BCD XS3	Aiken	BCD 5421
0	0000	0011	0000	0000
1	0001	0100	0001	0001
2	0010	0101	0010	0010
3	0011	0110	0011	0011
4	0100	0111	0100	0100
5	0101	1000	1011	1000
6	0110	1001	1100	1001
7	0111	1010	1101	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100

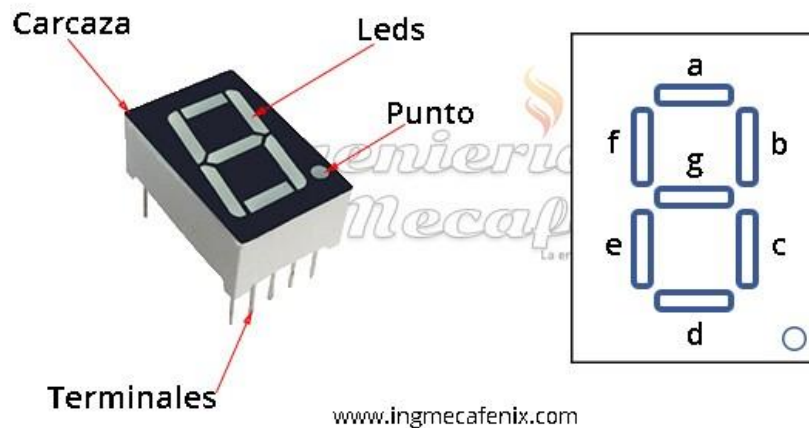
DISPLAY DE 7 SEGMENTOS

El display de 7 segmentos es un dispositivo electrónico que se utiliza para representar visualmente números y algunos caracteres. Este display es muy popular debido a su gran efectividad y simplicidad al momento de utilizarlo.

Partes de un display de 7 segmentos

Se le conoce como 7 segmentos por que cuenta con siete diodos led principales y uno extra para representar un punto. También cuenta con una carcasa para cubrirlos y 10 terminales: 2 son de alimentación (2 de Vcd o 2 de Gnd), 1 es para visualizar un punto y 7 son para representar cada uno de

los números según la combinación que se le ponga, estos están representados por una letra del abecedario desde la "A" hasta la letra "G".

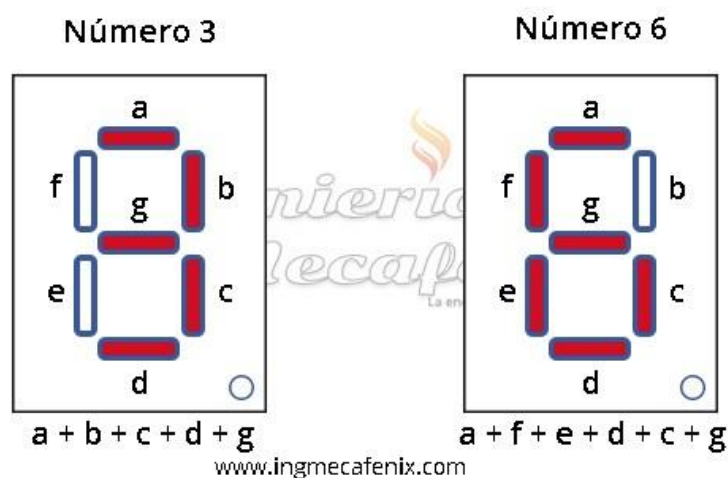


Funcionamiento de un display

Para poder representar los números o caracteres con este dispositivo solo basta con saber la configuración de cada una de sus leds y combinarlos.

Ejemplos:

Para representar el número 3 debemos encender las letras A, B, G, C y D, para representar el número 6 tenemos que combinar las letras A, F, E, D, C y G.



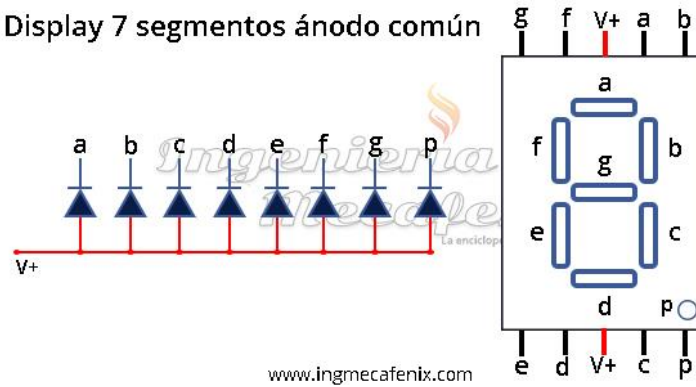
Tipos de display de 7 segmentos

Al ser un dispositivo muy simple no existe mucha variedad, solo podemos encontrar de:

Ánodo común

Se llama así porque todos los leds están unidos en su terminal positiva (ánodo), para encenderlos tenemos que poner tierra en la terminal de la letra que se desee.

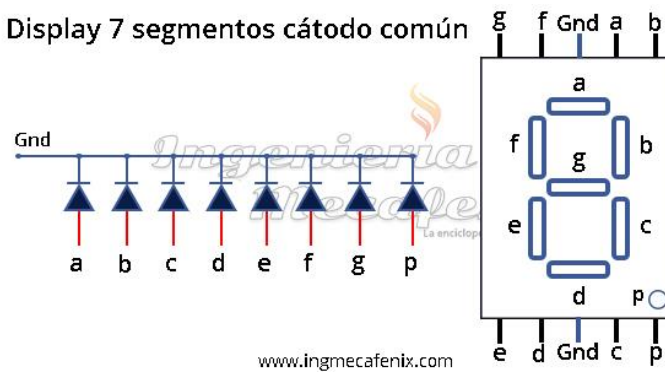
Display 7 segmentos ánodo común



Cátodo común

Este display es el opuesto del ánodo común ya que los leds están unidos en la terminal negativa (cátodo). Para encender los leds tenemos que poner voltaje en las terminales de las letras.

Display 7 segmentos cátodo común



Display múltiple

Dentro de este tipo podemos encontrar de ánodo o cátodo común, la única diferencia es que son 2, 4 y hasta 6 displays unidos. Estos son ideales para cuando se necesitan representar cifras de más de un dígito.



Por tamaño

En este tipo también podemos encontrar de ánodo o cátodo común, la única variación es el tamaño del display, ya que los podemos encontrar de 2.3 y 4 pulgadas.



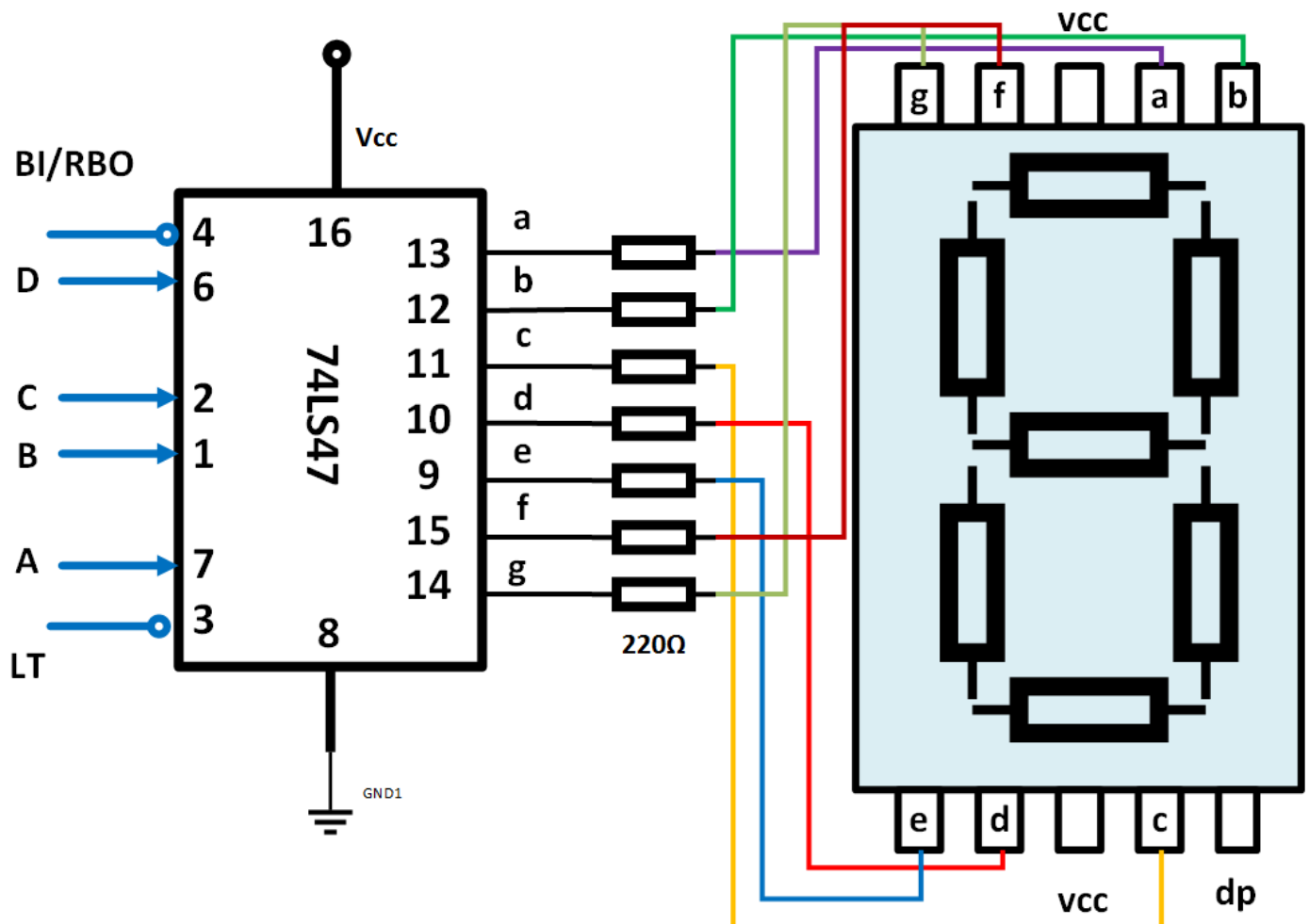
www.ingmecafenix.com

Aplicaciones del display 7 segmentos

Las principales aplicaciones de los displays 7 segmentos son como contadores, relojes de tiempo real, para desplegar marcadores o algún tipo de cuenta regresiva o incremental.

5.DIAGRAMAS

Diagrama eléctrico



SN74LS47

LOGIC DIAGRAM

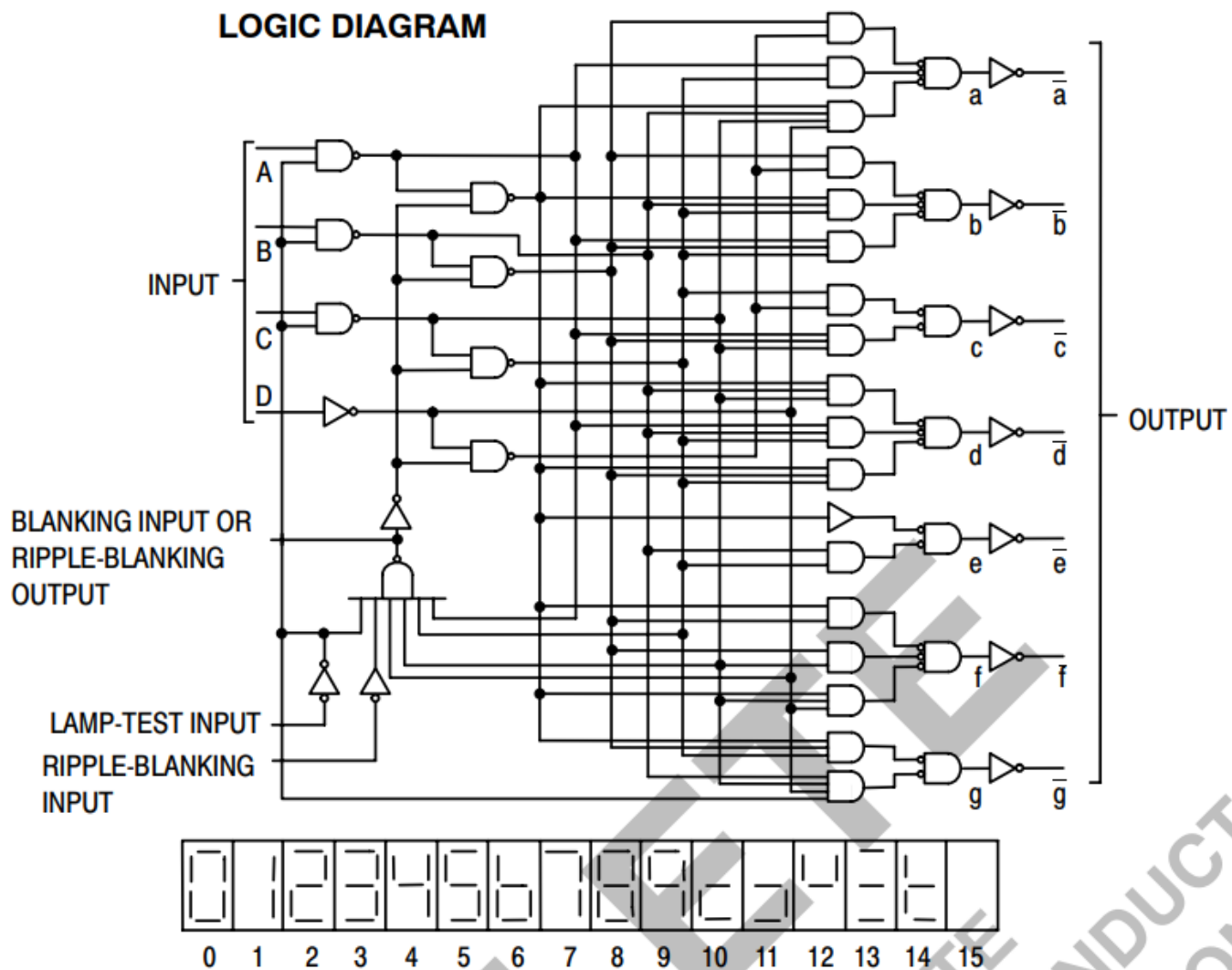
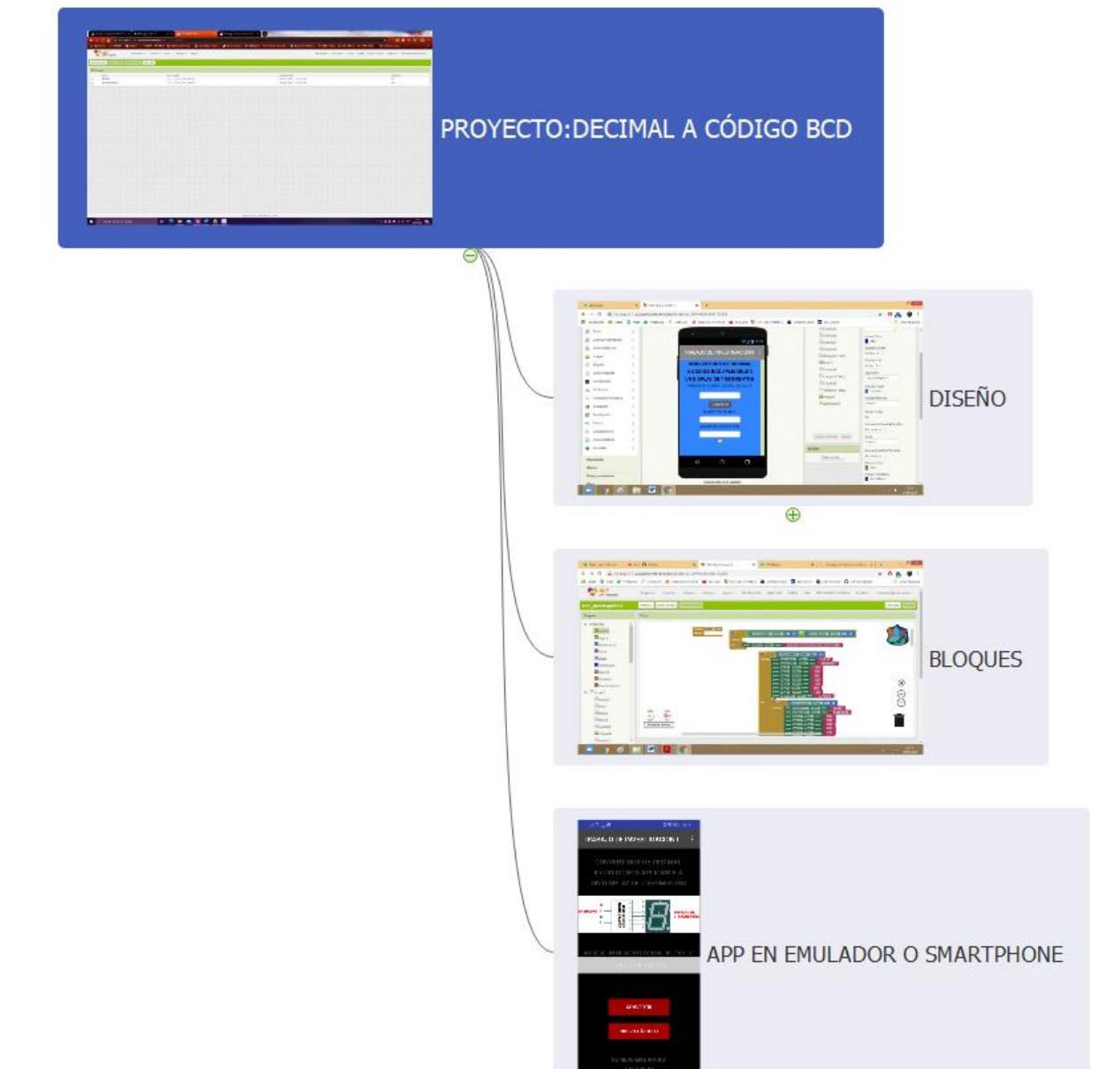


Diagrama de bloques

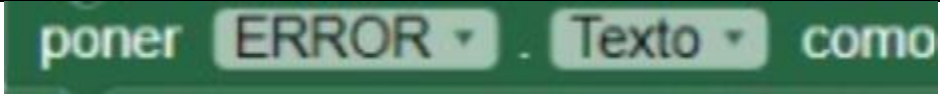
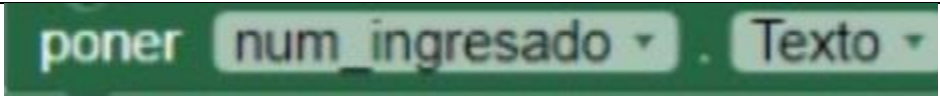
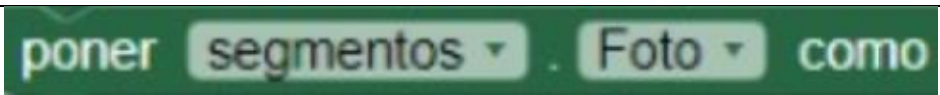

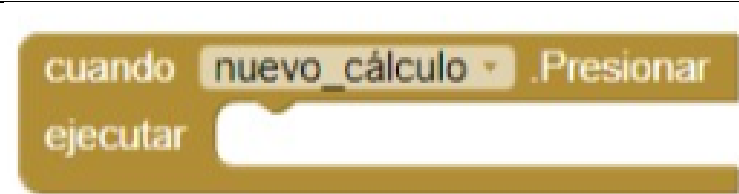




6.LISTA DE COMPONENTES

- Laptop
Procesador: Intel Core i5-7200U CPU @ 2.50Ghz
Memoria instalada (Ram): 8 Gb
Tipo de sistema: 64 bits.
Sistema operativo: Windows 8.0 en adelante.
- Smartphone.
Sistema operativo: Android
- App Inventor
Link: http://ai2.appinventor.mit.edu/?locale=es_ES
- MIT A

7.MAPA DE VARIABLES

VARIABLES	FUNCIÓN OPERATIVA
	<p>Esta variable de ejecución, nos inicializara el programa principal. Ejecutando la aplicación.</p>
	<p>Esta variable nos almacenara el numero binario obtenido, para luego ser impreso en la pantalla principal.</p>
	<p>Esta variable nos almacenara el número del código BCD.</p>
	<p>En estas variables se almacenará un “1” o un “0”, dependiendo del código BCD e indicara el segmento que se va a prender en el Display.</p>

	<p>En esta variable se almacenará el mensaje “El valor ingresado no es válido o se encuentra fuera de rango”. El cual se mostrará en la pantalla bajo algunas condiciones.</p>
	<p>Esta la variable que ingresa el usuario mediante el teclado para luego realizar la validación y ejecución del programa.</p>
	<p>Esta variable nos almacenara la imagen del estado del Display, teniendo nueve opciones de composición.</p>
	<p>Es la variable que guarda la imagen del integrado y del Display, que se muestra al inicio de la aplicación y es una variable fija.</p>
	<p>Es una variable de ejecución, que guarda como valor nulo o valor vacío las variables antes inicializadas.</p>
	<p>Esta variable de ejecución nos desencadenara la finalización de la aplicación.</p> <p></p>

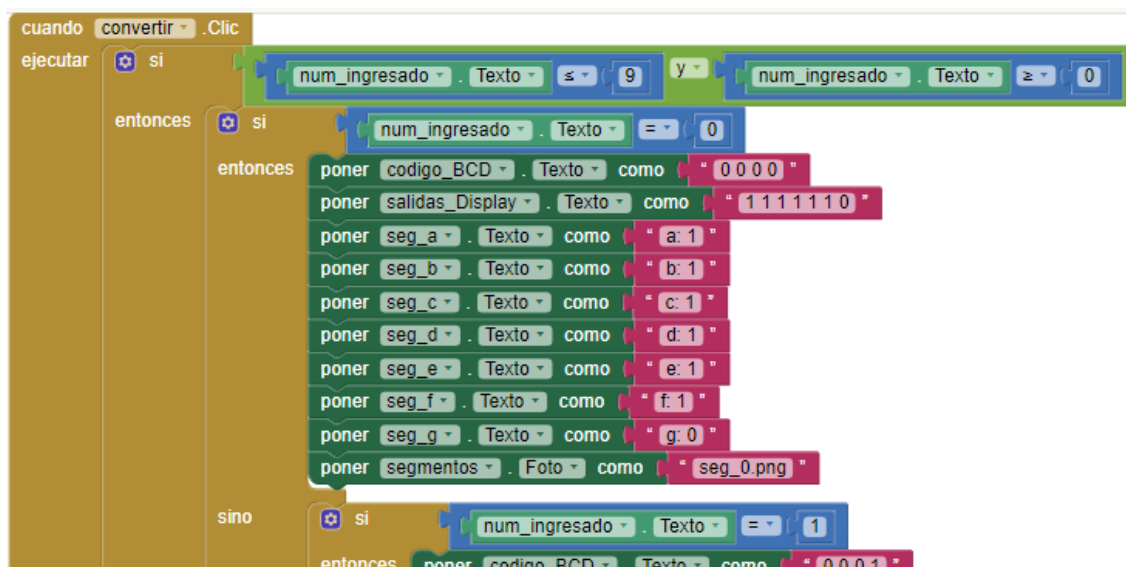
8.EXPLICACIÓN DEL CÓDIGO FUENTE

En la pestaña Diseñador colocamos todas las variables que van a intervenir en la ejecución del programa al mismo tiempo que se crea la interfaz gráfica de la aplicación, una vez inicializadas todas las variables en la pestaña Diseñador pasamos a la pestaña Bloques para diseñar el código de ejecución



Pestaña Bloques

Código 1



1. cuando convertir. Clic

ejecutar

Este botón guarda una serie de instrucciones para realizar la conversión de decimal a binario, las mismas que se ejecutarán siempre y cuando le demos un clic.

si

entonces

sino

Este ciclo realiza la validación de las condiciones, mientras se cumpla la condición Inicial se ejecuta el proceso de conversión de decimal a código BCD caso contrario arroja el mensaje ***“El valor ingresado no es válido o se***

encuentra fuera de rango.”

num_ingresado Texto (menor = 9) y num_ingresado Texto (mayor = 0)

Es la condición inicial que rige el ciclo, nos indica que siempre que se ingrese un dato de tipo “Texto” siendo un número entre el “0” y el “9”, el programa ingresa a un nuevo ciclo “si entonces sino” donde después de una validación final se procede a definir las variables y a configurar los mensajes que se arrojarán como resultados.

num_ingresado TEXTO = 0

Esta es la última validación que se realiza, aquí se controla si el número ingresado es igual a “0” entonces se definen las variables y se imprimen los resultados, caso contrario pasamos al siguiente número y realizamos el mismo proceso de validación siempre partiendo de un ciclo “si entonces sino”

poner codigo_BCD Texto como “0 0 0 0”

Esta variable va a imprimir el número “0 0 0 0” que es el equivalente en CÓDIGO BCD del “0”, según el valor que guarde num_ingresado se imprime un valor distinto, este proceso se realiza para los demás valores.

poner salidas_Display Texto como “1 1 1 1 1 1 0”

Esta variable imprime como resultado las SALIDAS que entrega el DISPLAY de 7 SEGMENTOS de ÁNODO COMÚN tomando como lógica de programación al “1” como ENCENDIDO y al “0” como APAGADO

poner seg_a Texto como “a: 1”

poner seg_b Texto como “b: 1”

poner seg_c Texto como “c: 1”

poner seg_d Texto como “d: 1”

poner seg_e Texto como “e: 1”

poner seg_f Texto como “f: 1”

poner seg_g Texto como “g: 1”

poner segmentos Foto como “seg_0.png”

En este bloque de variables
imprime verticalmente los
segmentos del display indicando
“1 encendido” y “0 apagado”

Esta variable permite imprimir la imagen que se va a proyectar según el código BCD

Este proceso se realiza para todos los valores del “0” al “9”

Imágenes Código 2 al código 6

poner ERROR Texto como “El valor ingresado no es válido o se encuentra fuera de rango”

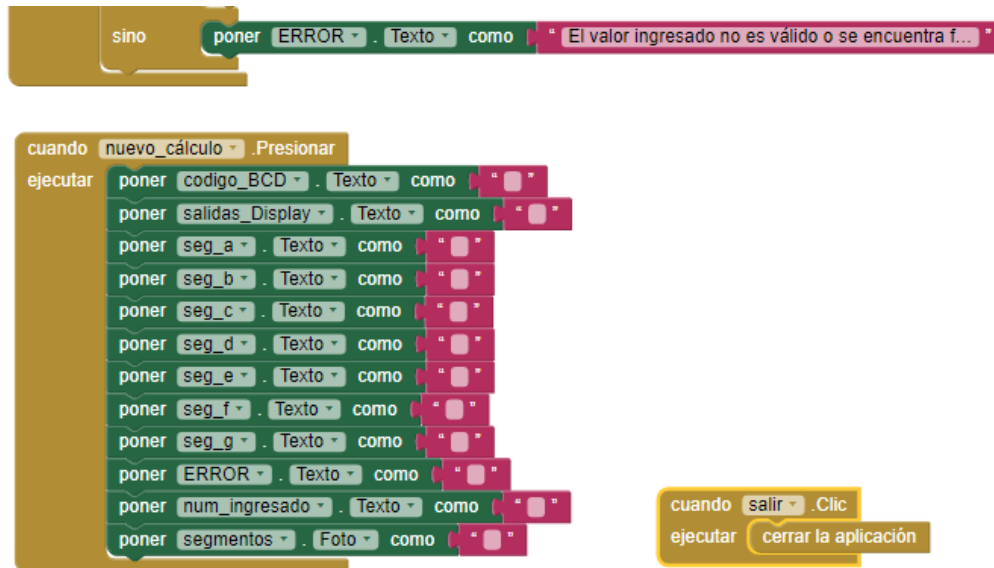
Con esta variable se manda a imprimir un mensaje de advertencia en el caso que los valores ingresados por el usuario no sean los correctos.

**2. cuando nuevo_cálculo .Presionar
ejecutar**

Este proceso reinicia todas las variables, limpiando la pantalla para un nuevo cálculo.

**3. cuando salir. Clic
ejecutar cerrar la aplicación**

Este proceso termina el programa y cierra la aplicación.



9.DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Tener una cuenta de Google, debido a que la verificación y creación de la cuenta en app inventor será a partir de esta.

Es esencial tener una conexión estable a internet, debido que será necesario para la creación del programa en app inventor.

Los sistemas operativos de los computadores en los cuales funciona App Inventor son:

- Macintosh (con procesador Intel): Mac OS X 10.5, 10.6
- Windows: Windows XP, Windows Vista, Windows 7
- GNU / Linux: Ubuntu 8 +, 5 + Debian

Tener un computador con todas las actualizaciones necesarias, como también nuestro navegador deberá tener las siguientes especificaciones:

- Mozilla Firefox 3.6 o superior
- Apple Safari 5.0 o superior
- Google Chrome 4.0 o superior
- Microsoft Internet Explorer 7 o superior

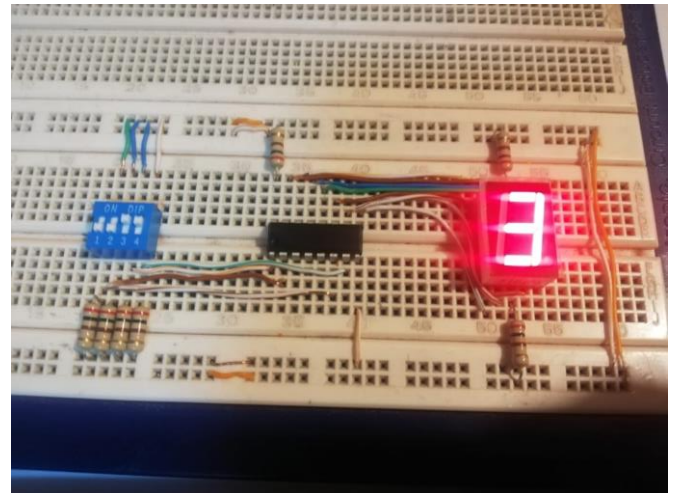
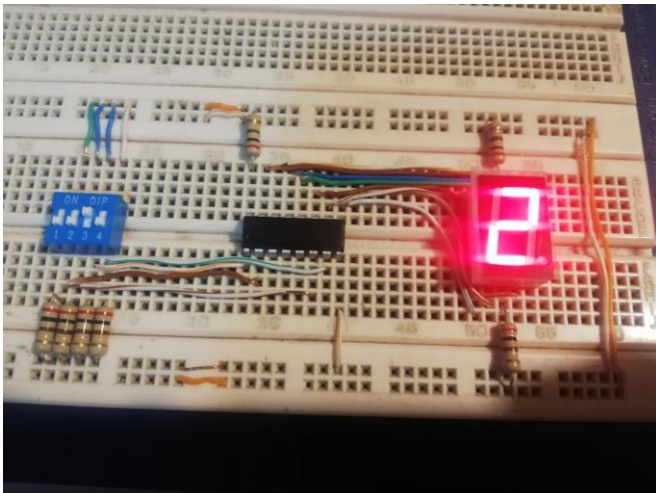
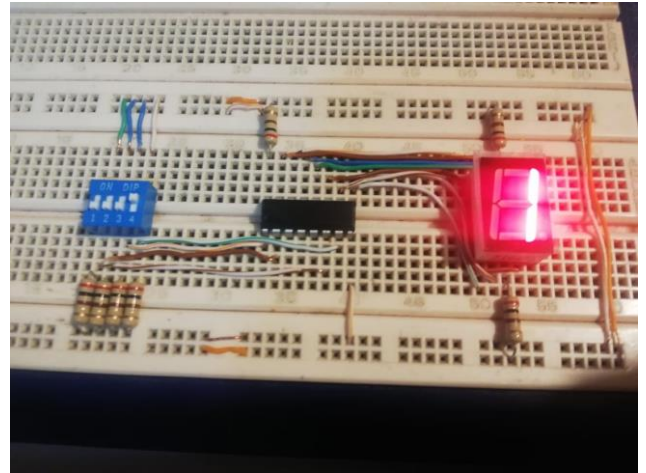
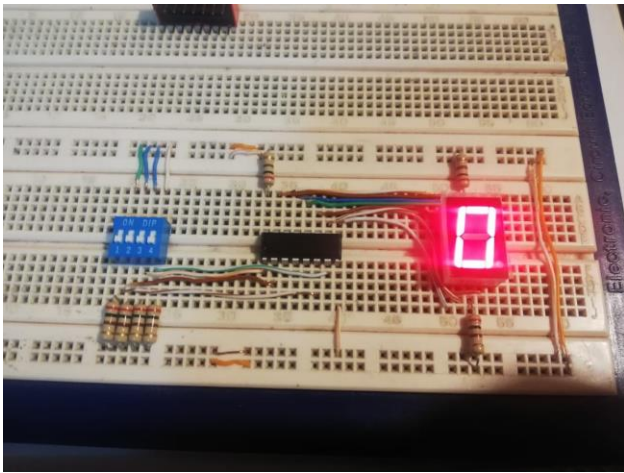
La aplicación creada con App Inventor puede funcionar en cualquier teléfono Android. El entorno de desarrollo

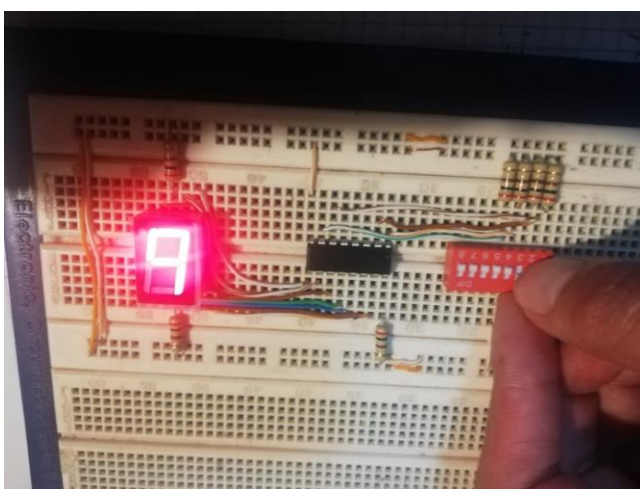
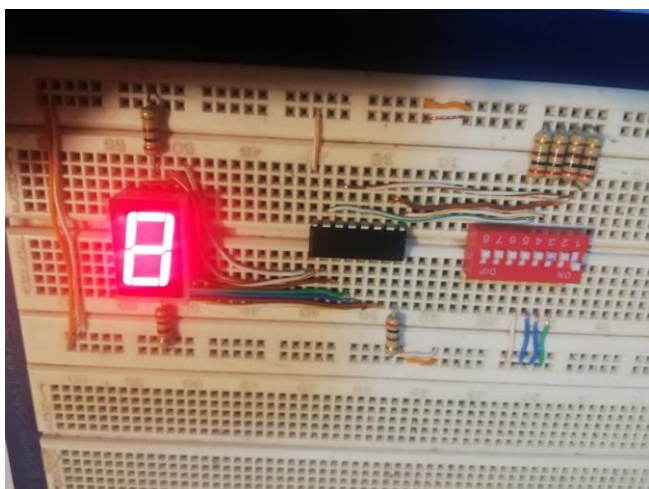
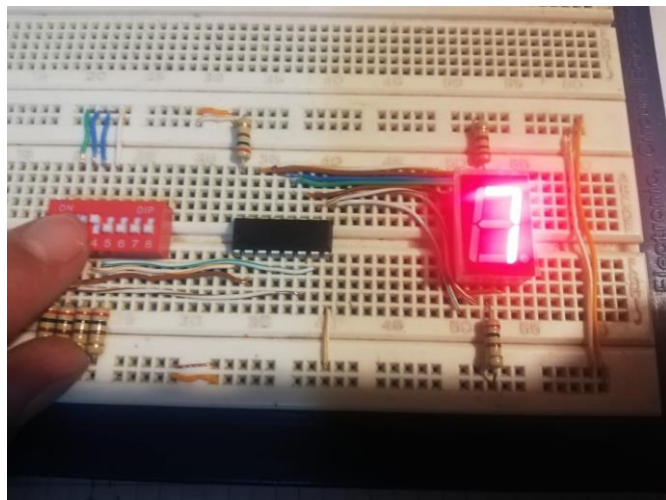
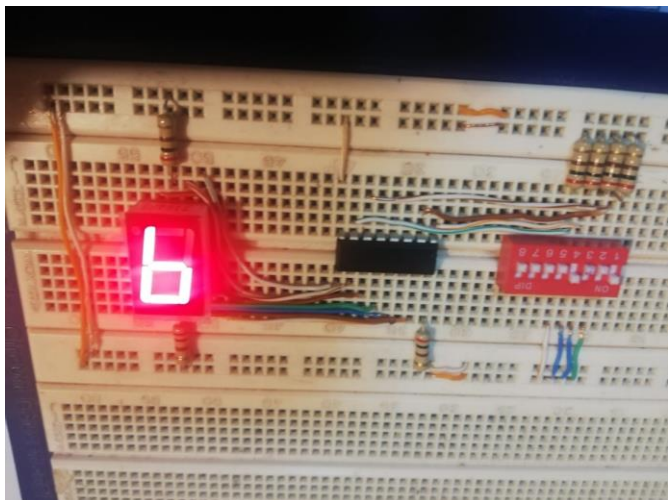
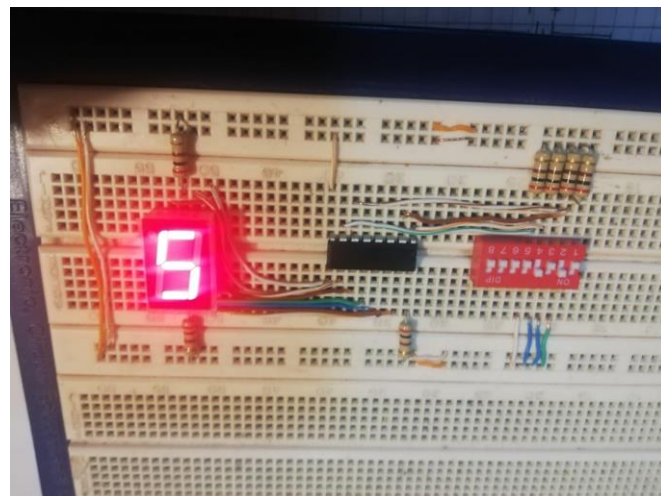
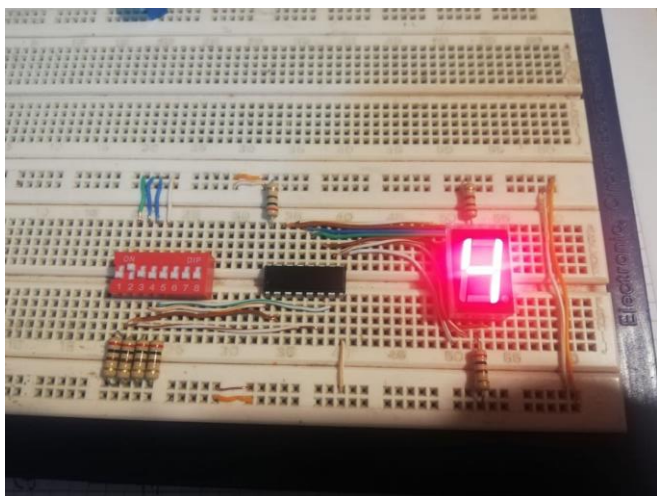
y software de instalación se apoya directamente a los teléfonos siguientes: Los dispositivos adicionales requieren los controladores de Windows proporcionados por el fabricante del hardware.

Para realizar los test de nuestro programa en el móvil, será necesario descargarnos una app desde play store para Android. El nombre de la app es “MIT AI2 Companion”

10.APORTACIONES:

Como parte de la aportación se realizó la implementación física a un display de 7 segmentos, a partir del código BCD. Este circuito se realizó a partir del decodificador “SN74LS47N”. En este circuito se requirió el uso de 7 resistencias de 200 Ohms como medida de protección. Se uso también dip switch de 4 puertos. El display de 7 segmentos con ánodo común. Este circuito se lo realizo en una proboard, usando como fuente de voltaje un transformador con las siguientes características:
Input: 120V – Output: 5V -1.2 A





11.CONCLUSIONES

En conclusión:

- Crear una aplicación para convertir un número decimal a BCD en App Inventor se torna relativamente sencillo ya que tiene una interfaz muy intuitiva sobre todo en el diseño de la app donde simplemente se escoge bloques a modo de programación y se empaqueta la aplicación en un archivo APK para instalar en cualquier smartphone.
- El display de siete segmentos funciona a partir de activar o desactivar cada led ("0" o "1") para formar el

número requerido. A partir de esta lógica, mediante el código BCD, el cual nos entrega una combinación diferente para las 10 opciones de números que se pueden generar en el display. Es importante mencionar que en la parte física existen diferentes tipos de display como el de ánodo común y cátodo común.

- La lógica utilizada en la aplicación para convertir un número decimal a BCD es muy semejante a la aplicada en Java o C++, programas conocidos por la mayoría de estudiantes de ingeniería lo cual facilitó en gran medida la ejecución de la app.

12.RECOMENDACIONES

- Se recomienda realizar la idea del programa en un diagrama de flujo, antes de empezar a programarlo en app inventor.
- No se debe asignar más de un proceso a un mismo botón, debido a que solo puede cumplir una función a la vez, si se incumple esto el botón quedara obsoleto.
- Antes de realizar los bloques de programación, primero se deberá colocar en la interfaz gráfica los botones, campos de texto e imágenes que sea necesarias para la ejecución de la aplicación. Ya que al momento de invocar las variables no se generen errores en el bucle establecido.
- Definir las variables de acuerdo a la función que van a desempeñar dentro del programa, para tener una mejor percepción del mismo.
- Al momento de realizar nuestro circuito de aportación, se recomienda colocar resistencias de protección tanto al Display de 7 segmentos como al integrado "SN74LS47". Como también es importante mantener un orden en los cables de conexión.

13.CRONOGRAMA

Trabajo de investigación	Persona	Estado	Fecha	Cronograma	
Investigación del proyecto	+2	Listo	may-27	may-27 - jun-1	
Pruebas de funcionamiento con App Inventor	+2	Listo	may-28	may-28 - 30	
Crear la cuenta en Github		Listo	may-28	may-28	
Desarrollo del trabajo de investigación	+2	Listo	may-29	may-29 - jun-12	
Desarrollo del programa en app inventor	+2	Listo	jun-1	jun-1 - 13	
Desarrollo del informe	+2	Listo	jun-5	jun-5 - 14	
Primera prueba del programa desarrollado en app inventor.	+2	Listo	jun-8	jun-8	
Desarrollo del paper		Listo	jun-7	jun-7 - 9	
Desarrollo del circuito como aportación		Listo	jun-11	jun-11 - 12	
Creación y edición del video multimedia		Listo	jun-11	jun-11 - 13	
Creación de las diapositivas		Listo	jun-12	jun-12 - 13	
+ Agregar					
				may-27 - jun-14	

14.BIBLIOGRAFÍA

[1] Munasinghe, T., Patton, EW y Seneviratne, O. (2019). Desarrollo de aplicaciones IoT usando MIT App Inventor para recopilar y analizar datos de sensores. Conferencia internacional IEEE 2019 sobre Big Data (Big Data). doi:

10.1109 / bigdata47090.2019.9006203

[2] Huang, R. y Turbak, F. (2019). Un diseño para la conversión bidireccional entre bloques y texto para App Inventor. 2019 IEEE Blocks and Beyond Workshop (B&B). doi: 10.1109 / bb48857.2019.8941197

[3] Lance A. Allison, Mohammad Muztaba Fuad. (2016). Comunicación entre aplicaciones entre aplicaciones de Android desarrolladas en App-Inventor y Android Studio. Conferencia 2016 IEEE / ACM International Conference on Mobile Software

Engineering and Systems (MOBILESoft). doi: 10.1109 / MobileSoft.2016.018

[4] Gonzales,Erick,(2018). Display 7 Segmentos ánodo y cátodo común.Recuperado de:
<https://hetpro-store.com/TUTORIALES/display-7-segmentos-anodo-catodo-comun/>

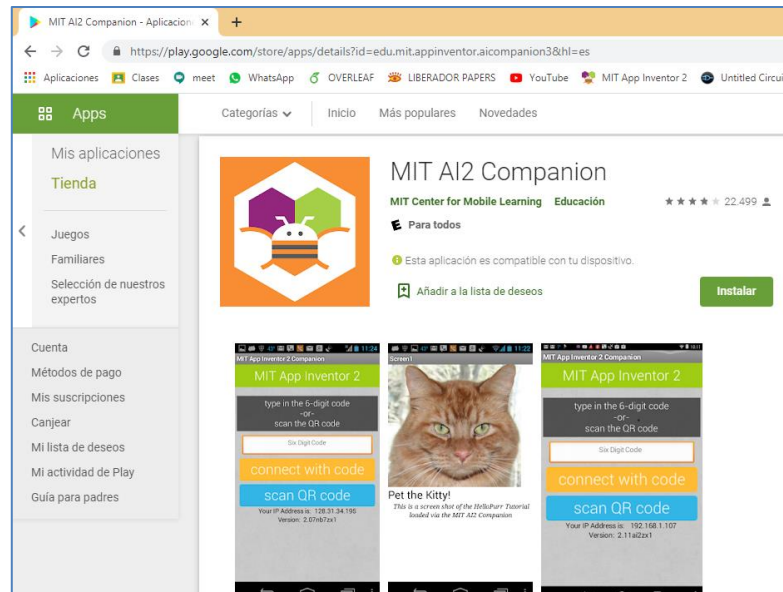
[5]Posada.Fernando,(2019). Creando aplicaciones para móviles Android con MIT App Inventor 2. DOI (web) 104438/2695-4176_OTTE_2019_847-19-121-5. <https://intef.es/wp-content/uploads/2019/03/MIT-App-Inventor-2.pdf>

15.ANEXOS

15.1MANUAL DE USUARIO

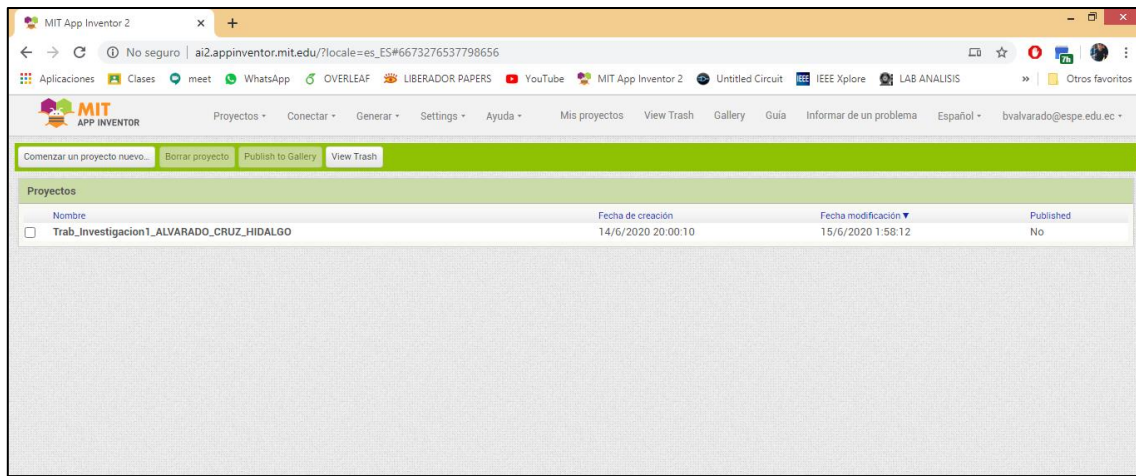
1. Para ejecutar la aplicación Convertidor de Decimal a Código BCD aplicable a un Display de 7 Segmentos, primero se debe instalar el apk MIT AI2 Companion en nuestro Smartphone con sistema ANDROID 6.0 en adelante, la descarga del apk se la realiza directamente del Play Store o en el enlace:

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3&hl=es>

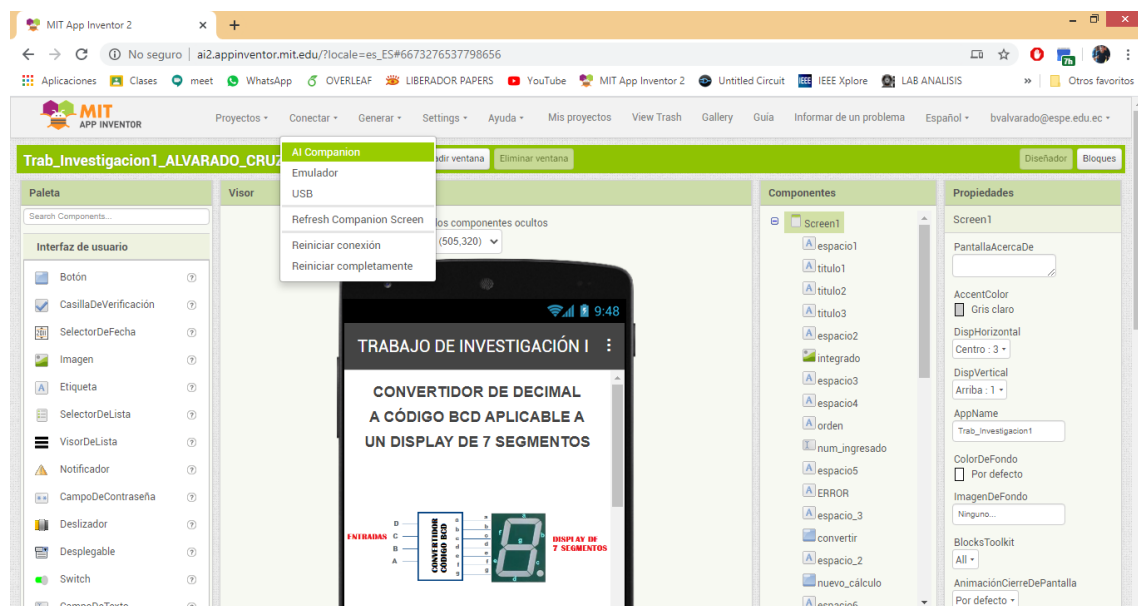


2. Desde un ordenador dirigirse al sitio web de App Inventor seleccionamos el proyecto Trab_Investigacion 1_ALVARADO_CRUZ_HIDALGO en el enlace:

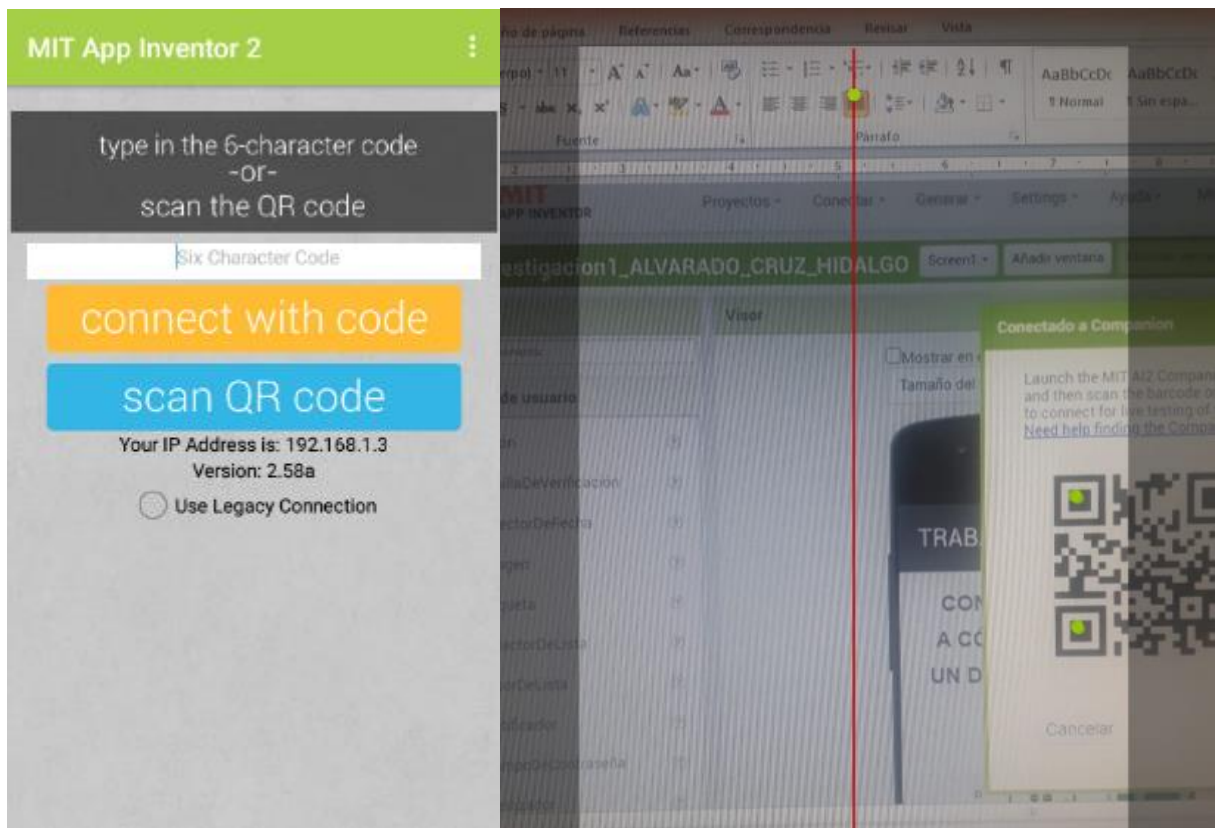
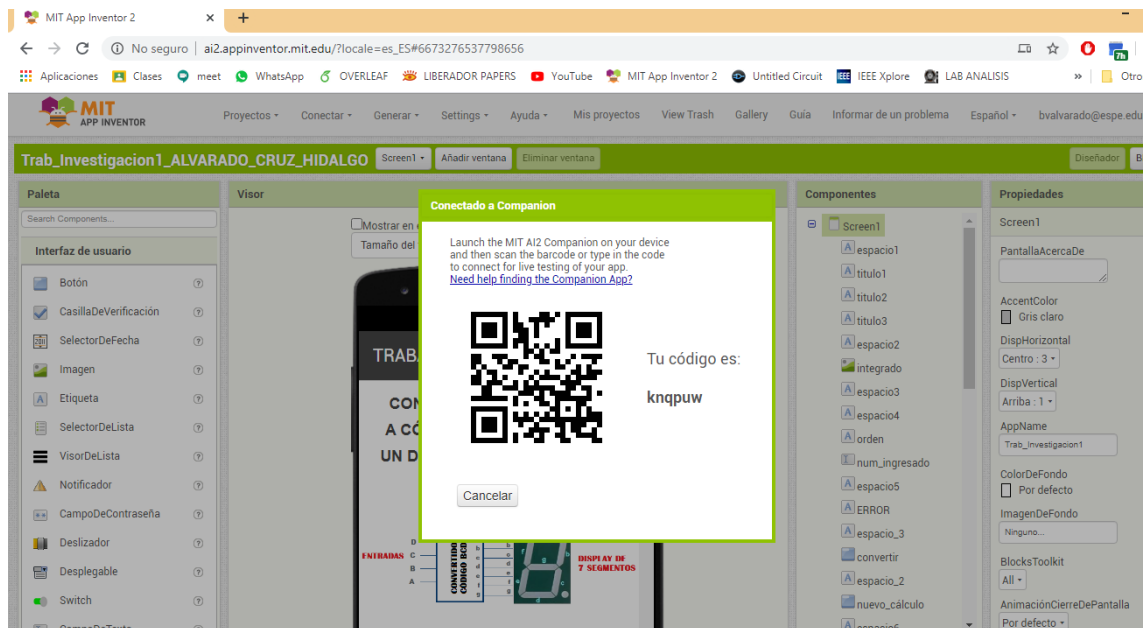
http://ai2.appinventor.mit.edu/?locale=es_ES#6673276537798656



3. Se puede visualizar la interfaz gráfica del proyecto, dar clic sobre la pestaña Conectar / Al Companion



4. Aparece un código QR como paso siguiente se debe abrir la aplicación MIT App Inventor 2 en el celular y escanear dicho código dando clic en la opción “scan QR code”



5. Automáticamente la aplicación se abre en el celular, en el campo “ingrese el número” se puede escribir cualquier número decimal de “0” a “9”



6. Una vez ingresado el valor deseado le damos clic en CONVERTIR, se puede observar que la aplicación muestra tanto el CÓDIGO BCD como las SALIDAS del Display de 7 segmentos.



7. Finalmente, para ingresar un nuevo número, se debe dar clic en el botón "NUEVO CÁLCULO" caso contrario dar clic en la opción "SALIR" para cerrar la aplicación.

NUEVO CÁLCULO

CÓDIGO BCD
(D C B A)

SALIDAS DEL DISPLAY DE 7 SEGMENTOS
(a b c d e f g)

"1" ENCENDIDO "0" APAGADO

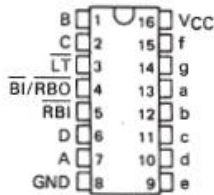
SALIR

15.2HOJAS TÉCNICAS

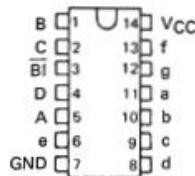
SN5446A, '47A, '48, SN54LS47, 'LS48, 'LS49
 SN7446A, '47A, '48, SN74LS47, 'LS48, 'LS49
 BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS
 SDSL111 - MARCH 1974 - REVISED MARCH 1988

'46A, '47A, 'LS47 feature	'48, 'LS48 feature	'LS49 feature
<ul style="list-style-type: none"> • Open-Collector Outputs Drive Indicators Directly • Lamp-Test Provision • Leading/Trailing Zero Suppression 	<ul style="list-style-type: none"> • Internal Pull-Ups Eliminate Need for External Resistors • Lamp-Test Provision • Leading/Trailing Zero Suppression 	<ul style="list-style-type: none"> • Open-Collector Outputs • Blanking Input

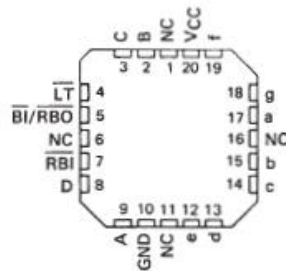
SN5446A, SN5447A, SN54LS47, SN5448,
 SN54LS48 . . . J PACKAGE
 SN7446A, SN7447A,
 SN7448 . . . N PACKAGE
 SN74LS47, SN74LS48 . . . D OR N PACKAGE
 (TOP VIEW)



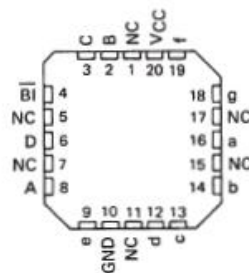
SN54LS49 . . . J OR W PACKAGE
 SN74LS49 . . . D OR N PACKAGE
 (TOP VIEW)



SN54LS47, SN54LS48 . . . FK PACKAGE
 (TOP VIEW)



SN54LS49 . . . FK PACKAGE
 (TOP VIEW)



NC - No internal connection

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
 INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1988, Texas Instruments Incorporated

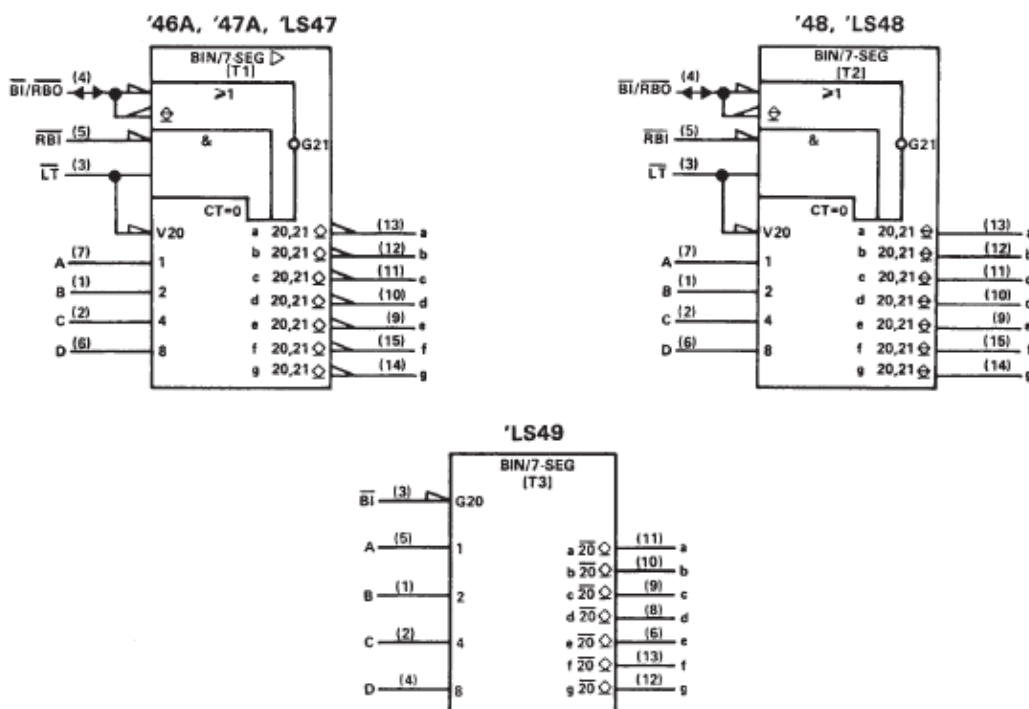
1

**SN5446A, '47A, '48, SN54LS47, 'LS48, 'LS49
SN7446A, '47A, '48, SN74LS47, 'LS48, 'LS49
BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS**
SDLS111 - MARCH 1974 - REVISED MARCH 1988

- All Circuit Types Feature Lamp Intensity Modulation Capability

TYPE	DRIVER OUTPUTS				TYPICAL POWER DISSIPATION	PACKAGES
	ACTIVE LEVEL	OUTPUT CONFIGURATION	SINK CURRENT	MAX VOLTAGE		
SN5446A	low	open-collector	40 mA	30 V	320 mW	J, W
SN5447A	low	open-collector	40 mA	15 V	320 mW	J, W
SN5448	high	2-k Ω pull-up	6.4 mA	5.5 V	265 mW	J, W
SN54LS47	low	open-collector	12 mA	15 V	35 mW	J, W
SN54LS48	high	2-k Ω pull-up	2 mA	5.5 V	125 mW	J, W
SN54LS49	high	open-collector	4 mA	5.5 V	40 mW	J, W
SN7446A	low	open-collector	40 mA	30 V	320 mW	J, N
SN7447A	low	open-collector	40 mA	15 V	320 mW	J, N
SN7448	high	2-k Ω pull-up	6.4 mA	5.5 V	265 mW	J, N
SN74LS47	low	open-collector	24 mA	15 V	35 mW	J, N
SN74LS48	high	2-k Ω pull-up	6 mA	5.5 V	125 mW	J, N
SN74LS49	high	open-collector	8 mA	5.5 V	40 mW	J, N

logic symbols†



†These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12. Pin numbers shown are for D, J, N, and W packages.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

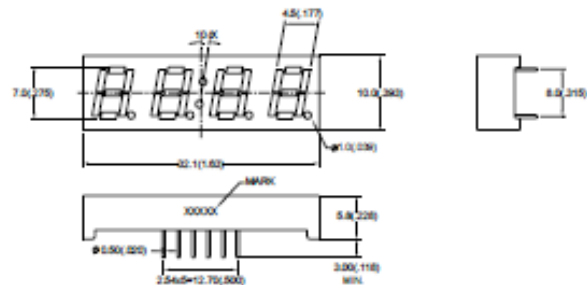
Features :

1. 0.28 inch (7.00mm) Digit Height.
2. Continuous uniform segments.
3. Low power requirement.
4. Excellent characters appearance.
5. Solid state reliability.
6. Categorized for luminous intensity.
7. Multiplex drive common anode.

Description :

1. The BQ-M282RD is a 7.00mm (0.28") high four digit seven segments display.
2. This product use green chips, which are made from GaP on GaP substrate.
3. This product have a black face and white segments.
4. This product doesn't contain restriction substance, comply ROHS standard.

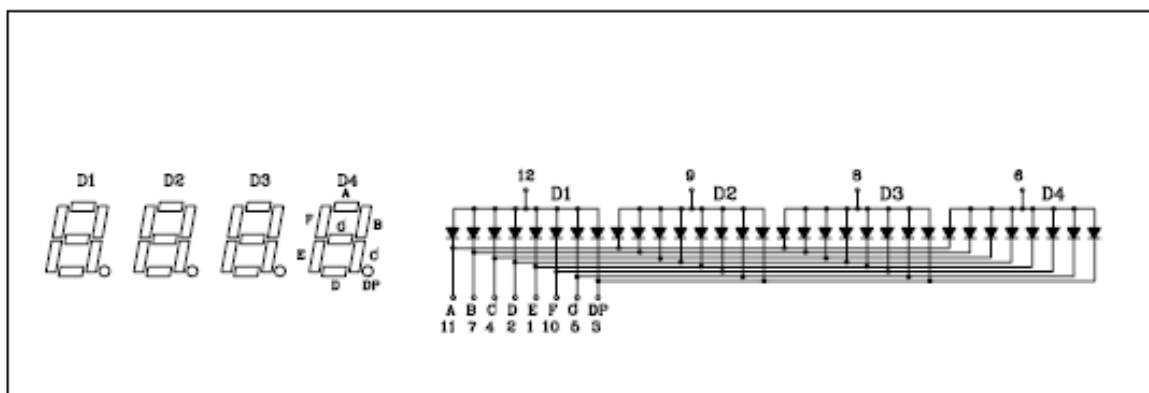
Package Dimensions :



Notes:

1. All dimensions are in millimeters(inches).
2. Tolerance is $\pm 0.25\text{mm}(.01")$ unless otherwise specified.
3. Specifications are subject to change without notice.

Internal Circuit Diagram :



● Absolute Maximum Ratings(Ta=25°C)

Parameter	Symbol	Rating	Unit
Power Dissipation Per Segment	Pd	80	mW
Forward Current Per Segment	I _F	30	mA
Peak Forward Current Per Segment	I _{FP} (Duty 1/10, 1KHZ)	150	mA
Reverse Voltage Per Segment	V _R	5	V
Operating Temperature	Topr	-40°C~80°C	-
Storage Temperature	Tstg	-40°C~85°C	-
Soldering Temperature (1/16" From Body)	Tsol	260°C For 5 Seconds	-

● Electrical And Optical Characteristics(Ta=25°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage Per Segment	V _f	I _F =10mA	-	2.1	2.5	V
Luminous Intensity Per Segment	I _v	I _F =10mA	-	2.5	-	mcd
Reverse Current Per Segment	I _R	V _R =5V	-	-	100	μA
Peak Wave Length	λ _p	I _F =10mA	-	568	-	nm
Dominant Wave Length	λ _d	I _F =10mA	569	-	574	nm
Spectral Line Half-width	Δλ	I _F =10mA	-	30	-	nm

DIP Switches - GD Series, High Feature Line

GDS Series

- Low profile
- End stackable
- Auto insertable
- Gold contacts



GDP Series

- Piano actuator
- Classic profile
- Gold contacts



GDR Series

- Recessed rocker actuator
- Classic profile
- Gold contacts



Dips

Standard number of positions	2, 4, 6, 8, 10
Mounting style	THT, SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	6.2 x 2.7 mm
Packaging	Tube, Tape and Reel

Standard number of positions	2, 4, 6, 8, 10
Mounting style	THT, SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	9.6 x 8.7 mm
Packaging	Tube, Tape and Reel

Standard number of positions	2, 4, 6, 8, 10
Mounting style	THT, SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	9.6 x 8.7 mm
Packaging	Tube, Tape and Reel

Standard number of positions	2, 4, 6, 8, 10
Mounting style	THT, SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	9.6 x 6.9 mm
Packaging	Tube, Tape and Reel

GDH Series

- Low profile
- Half pitch
- Gold contacts



GDHL Series

- Ultra low profile
- Half pitch
- Gold contacts



Dips

Standard number of positions	2, 4, 6, 8, 10
Mounting style	SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	6.2 x 3.2 mm
Packaging	Tape and Reel

Standard number of positions	2, 4, 6, 8, 10
Mounting style	SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	4.5 x 1.45 mm
Packaging	Tube, Tape and Reel

Standard number of positions	2, 4, 6, 8, 10
Mounting style	SMT
Tape seal	Yes
Contact rating	25 mA @ 24 Vdc
Electrical endurance	1000 ops.
Operating temperature range	-30...+85°C
Dimensions (WxH)	4.5 x 1.45 mm
Packaging	Tube, Tape and Reel