



# CIRCUITOS DIGITALES

Ing. Abg. Darwin Alulema MSc.



## INTEGRANTES:

- Alvarado Bryan
  - Cruz Jorge
- Hidalgo Evelin



TEMA:

REALIZAR UN PROGRAMA EN APP INVENTOR PARA REALIZAR LA CONVERSIÓN DE UN NUMERO DECIMAL A CÓDIGO BCD APLICABLE AUN DISPLAY DE 7 SEGMENTOS. NOTA: NO SE ADMITE EL EMPLEO DE LIBRERÍAS PREESTABLECIDAS.

# 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?

# OBJETIVOS:

- 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 funciones en el diseño del código.

# DESCRIPCIÓN DE PRERREQUISITOS

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



MIT App Inventor 2

Projects - Connect - Build - Language - Help

My Projects

Create new project

Projects

Name	Date Created	Date Modified
adminAgenda	Jul 2, 2014 6:01:37 PM	Sep 12, 2014 8:50:22 AM
foto	Sep 5, 2014 7:55:57 AM	Sep 11, 2014 8:53:06 AM
rebota	Dec 12, 2013 12:05:31 PM	
destrozar	Dec 12, 2013 6:08:20 PM	
CulturaTudela	Jul 3, 2014 4:20:26 PM	
agendacultural	Jul 1, 2014 12:00:58 PM	
GPS	Jun 30, 2014 11:47:45 AM	
DiWeb	Jul 1, 2014 10:09:42 AM	
basicoa	Jul 2, 2014 10:05:30 AM	
contacto	Jul 1, 2014 11:07:55 AM	
calendar	May 26, 2014 8:20:12 AM	
multiple	Dec 16, 2013 8:35:18 PM	
elq	Dec 16, 2013 7:04:44 PM	
casas	Jun 30, 2014 10:22:50 AM	
avis	Jan 30, 2014 4:09:24 PM	
Deja	Dec 23, 2013 6:24:23 PM	
guerra	Dec 20, 2013 4:17:24 PM	
tema	Dec 20, 2013 7:50:00 AM	
Lanzad	Dec 19, 2013 8:47:26 PM	
Pantallas	Dec 13, 2013 7:10:40 PM	
sensor	Dec 13, 2013 9:38:20 AM	
dibujar_checked	Dec 13, 2013 6:45:54 PM	
dibujar	Dec 13, 2013 12:33:37 PM	
vsd	Dec 13, 2013 8:34:13 AM	
DibuCam	Dec 11, 2013 12:19:11 PM	
prueba3	Dec 10, 2013 6:27:30 PM	
Prueba	Dec 9, 2013 8:40:19 AM	
prueba		

Create new App Inventor project

Project name:

Cancel OK

Desde el botón My projects podemos acceder a los proyectos que tenemos guardados

Para crear un nuevo proyecto

## Palette

## User interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebView

## Layout

## Media

## Drawing and Animation

## Sensors

## Social

## Storage

## Connectivity

## LEGO® MINDSTORMS®

## Viewer

☐ Display hidden components in Viewer

Screen1

## Components

☐ Screen1

## Properties

Screen1

AboutScreen

AlignHorizontal

Left

AlignVertical

Top

BackgroundColor

☐ White

Backgroundimage

None...

CloseScreenAnimation

Default

Icon

None...

OpenScreenAnimation

Default

ScreenOrientation

Unspecified

Scrollable

Title

Screen1

VersionCode

1

VersionName

1.0

Para poder incluir componentes al diseño de nuestra aplicación solo hay que seleccionar los componentes en la paleta y arrástralos al visor



## Palette

## User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider

## Layout

## Media

## Drawing and Animation

## Sensors

## Social

## Storage

## Connectivity

## LEGO® MINDSTORMS®

## Viewer

☐ Display hidden components in Viewer

Screen1

Text for Button1

Text for Label1

## Components

- Screen1
  - HorizontalArrangement1
    - TextBox1
    - TextBox2
  - Button1
  - Label1

## Properties

Label1

BackgroundColor

☐ None

FontBold

☐

FontItalic

☐

FontSize

14.0

FontTypeface

default ▾

Text

Text for Label1

TextAlignment

left ▾

TextColor

☒ Black

Visible

showing ▾

Width

Automatic...

Height

Automatic...

Situa los componentes en el visor para establecer el aspecto de tu aplicación

A medida que incorpores componentes a tu aplicación se irán añadiendo a la lista de componentes

En el panel media añadimos imágenes, sonidos o vídeos a nuestra App

## Media

Upload File ...

# Diseñador (Designer)

Para pasar al editor de Bloques

The screenshot shows the MIT App Inventor 2 Designer interface. At the top, there's a navigation bar with tabs for 'Primer', 'Screen1', 'Add Screen ...', and 'Remove Screen'. On the right, there are buttons for 'Designer' and 'Blocks'. The 'Blocks' button is highlighted with a blue box and an arrow pointing to it from the text 'Para pasar al editor de Bloques'.

The interface is divided into four main panels:

- Palette:** Contains various UI components like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, TextBox, TimePicker, and WebViewer. It also has sections for Layout, Media, Drawing and Animation, Sensors, Social, Storage, and Connectivity.
- Viewer:** Shows a preview of the app. A red box highlights a 'Text' component in the viewer, with an arrow pointing to it from the text 'Seleccionas un componente de la lista'.
- Components:** Shows a tree view of the app's components. A red box highlights the 'Screen1' component, which contains 'HorizontalArrangement1', 'TextBox1', 'TextBox2', 'Button1', and 'Label1'. An arrow points from this box to the 'Text' component in the Viewer.
- Properties:** Shows the properties of the selected 'Label1' component. A red box highlights this panel, with an arrow pointing to it from the text 'Y las propiedades del componente seleccionado aparecen en el panel'.

Annotations in Spanish:

- 'Seleccionas un componente de la lista' (You select a component from the list) - points to the 'Text' component in the Viewer.
- 'Y las propiedades del componente seleccionado aparecen en el panel' (And the properties of the selected component appear in the panel) - points to the Properties panel.



## Blocks

## Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

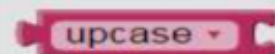
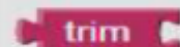
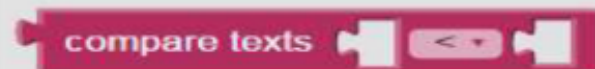
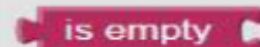
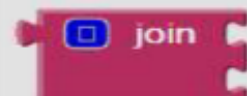
## Screen1

- HorizontalArrangement1
  - TextBox1
  - TextBox2
  - Button1
  - Label1
- Any component

Rename

Delete

## Viewer




when Button1 .Click  
do set TextBox1 .Text to Hola Mur

Arrastra los bloques para  
utilizarlos en el programa

Encaja las piezas como en un  
puzle para establecer el  
comportamiento de la aplicación





CÓDIGO FUENTE:

cuando convertir .Clic

ejecutar

si

num\_ingresado . Texto ≤ 9

y

num\_ingresado . Texto ≥ 0

entonces

si

num\_ingresado . Texto = 0

entonces

poner codigo\_BCD . Texto como " 0 0 0 0 "

poner salidas\_Display . Texto como " 1 1 1 1 1 1 0 "

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: 0 "

poner segmentos . Foto como " seg\_0.png "

sino

si

num\_ingresado . Texto = 1

entonces poner codigo\_BCD . Texto como " 0 0 0 1 "



si

num\_ingresado - . Texto - = - 1

entonces

poner codigo\_BCD - . Texto - como - " 0 0 0 1 "

poner salidas\_Display - . Texto - como - " 0 1 1 0 0 0 0 "

poner seg\_a - . Texto - como - " a: 0 "

poner seg\_b - . Texto - como - " b: 1 "

poner seg\_c - . Texto - como - " c: 1 "

poner seg\_d - . Texto - como - " d: 0 "

poner seg\_e - . Texto - como - " e: 0 "

poner seg\_f - . Texto - como - " f: 0 "

poner seg\_g - . Texto - como - " g: 0 "

poner segmentos - . Foto - como - " seg\_1.png "

sino

si

num\_ingresado - . Texto - = - 2

entonces poner codigo\_BCD - . Texto - como - " 0 0 1 0 "

poner salidas\_Display - . Texto - como - " 1 1 0 1 1 0 1 "

poner seg\_a - . Texto - como - " a: 1 "

poner seg\_b - . Texto - como - " b: 1 "

poner seg\_c - . Texto - como - " c: 0 "

poner seg\_d - . Texto - como - " d: 1 "

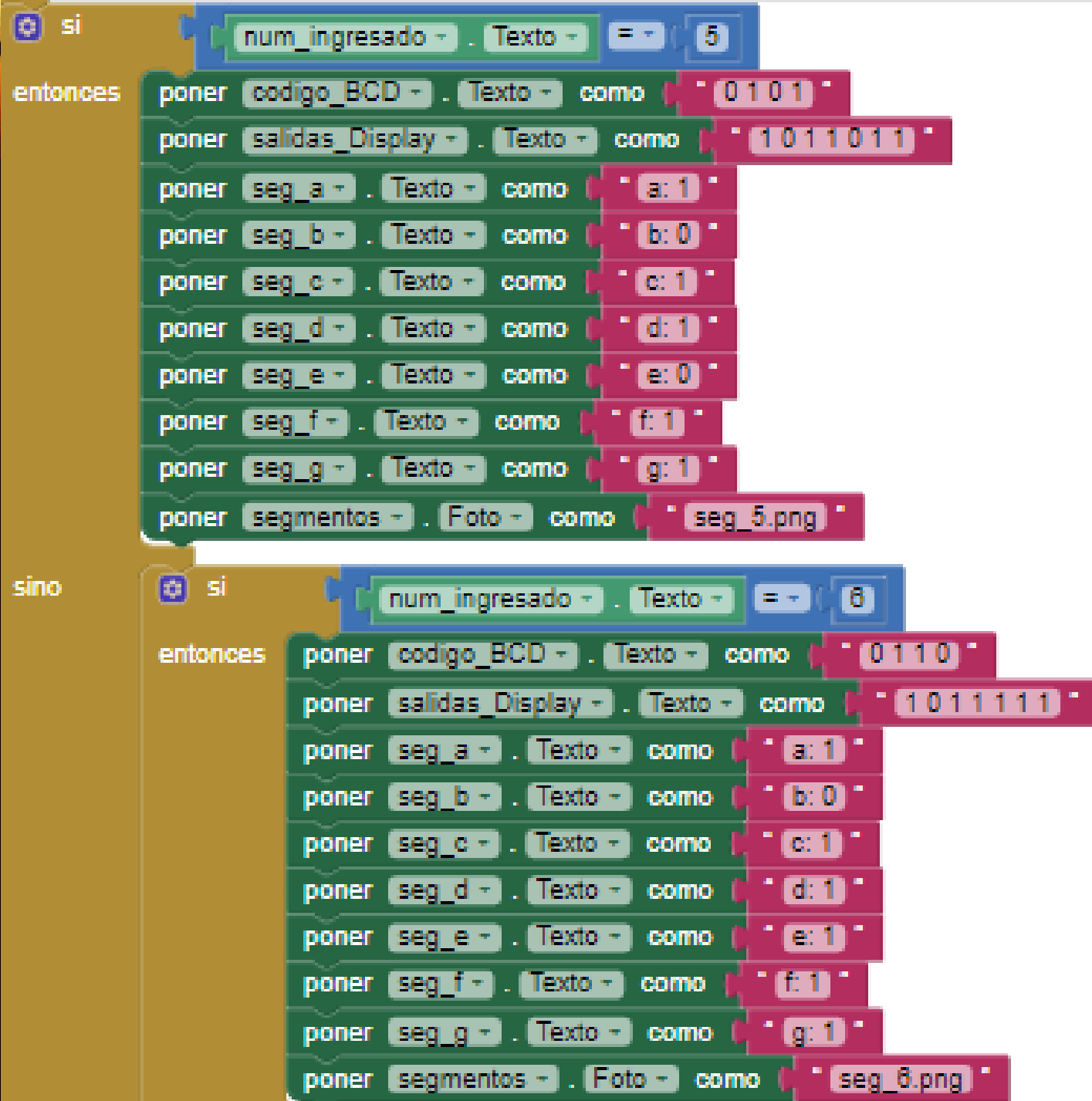
poner seg\_e - . Texto - como - " e: 1 "

poner seg\_f - . Texto - como - " f: 0 "

poner seg\_g - . Texto - como - " g: 1 "

poner segmentos - . Foto - como - " seg\_2.png "







si

num\_ingresado - Texto - = - 9

entonces

poner codigo\_BCD - Texto - como " 1 0 0 1 "

poner salidas\_Display - Texto - como " 1 1 1 0 0 1 1 "

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: 0 "

poner seg\_e - Texto - como " e: 0 "

poner seg\_f - Texto - como " f: 1 "

poner seg\_g - Texto - como " g: 1 "

poner segmentos - Foto - como " seg\_9.png "



sino

poner

ERROR ▾

.

Texto ▾

como

" El valor ingresado no es válido o se encuentra f... "

cuando nuevo\_cálculo ▾ .Presionar

ejecutar

poner

codigo\_BCD ▾

.

Texto ▾

como

" "

" "

poner

salidas\_Display ▾

.

Texto ▾

como

" "

" "

poner

seg\_a ▾

.

Texto ▾

como

" "

" "

poner

seg\_b ▾

.

Texto ▾

como

" "

" "

poner

seg\_c ▾

.

Texto ▾

como

" "

" "

poner

seg\_d ▾

.

Texto ▾

como

" "

" "

poner

seg\_e ▾

.

Texto ▾

como

" "

" "

poner

seg\_f ▾

.

Texto ▾

como

" "

" "

poner

seg\_g ▾

.

Texto ▾

como

" "

" "

poner

ERROR ▾

.

Texto ▾

como

" "

" "

poner

num\_ingresado ▾

.

Texto ▾

como

" "

" "

poner

segmentos ▾

.

Foto ▾

como

" "

" "

cuando salir ▾ .Clic

ejecutar

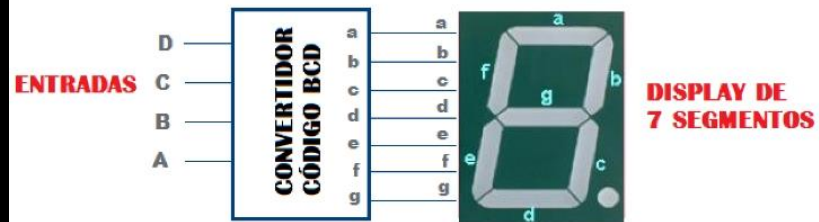
cerrar la aplicación



# INTERFAZ GRÁFICA

# TRABAJO DE INVESTIGACIÓN I

## CONVERTIDOR DE DECIMAL A CÓDIGO BCD APLICABLE A UN DISPLAY DE 7 SEGMENTOS



INGRESE UN NÚMERO DECIMAL DEL " 0 al 9 "

5

CONVERTIR

NUEVO CÁLCULO

# TRABAJO DE INVESTIGACIÓN I

CÓDIGO BCD  
( D C B A )

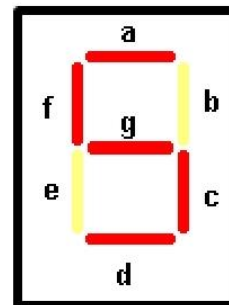
0 1 0 1

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

1 0 1 1 0 1 1

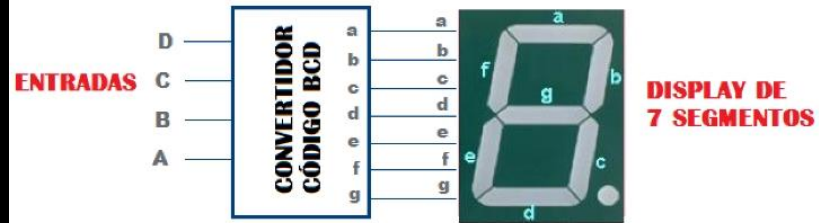
"1" ENCENDIDO ..... "0" APAGADO

a: 1  
b: 0  
c: 1  
d: 1  
e: 0  
f: 1  
g: 1



SALIR

# TRABAJO DE INVESTIGACIÓN I



INGRESE UN NÚMERO DECIMAL DEL " 0 al 9 "

15

El valor ingresado no es válido o se encuentra fuera de rango

CONVERTIR

NUEVO CÁLCULO

NÚMERO EN BINARIO  
( D C B A )



## TRABAJO DE INVESTIGACIÓN I

INGRESE UN NÚMERO DECIMAL DEL " 0 al 9 "

15

El valor ingresado no es válido o se encuentra fuera de rango

CONVERTIR

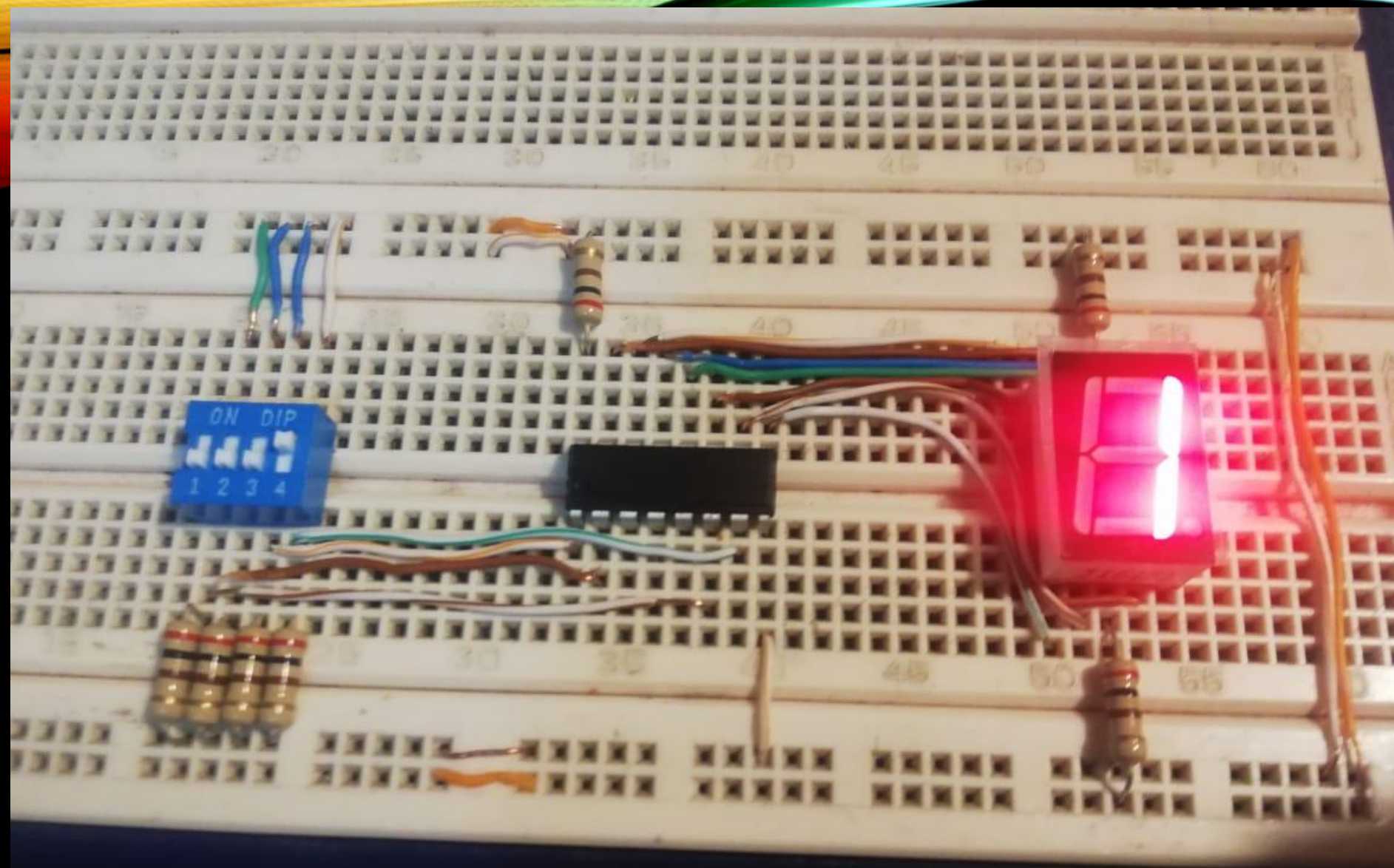
NUEVO CÁLCULO

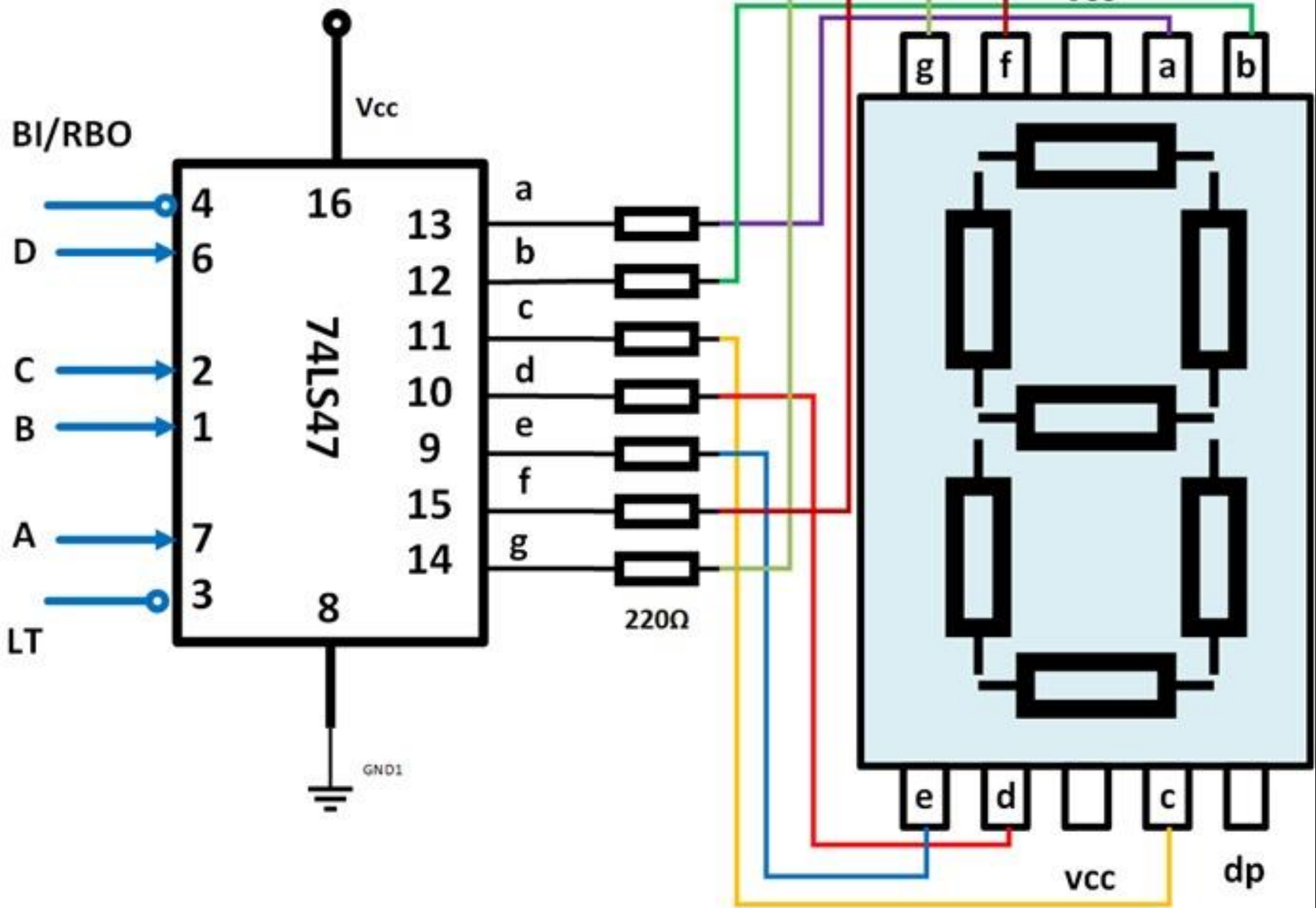
NÚMERO EN BINARIO  
( D C B A )

NÚMERO EN CÓDIGO BCD  
( a b c d e f g )

# APORTACIÓN:

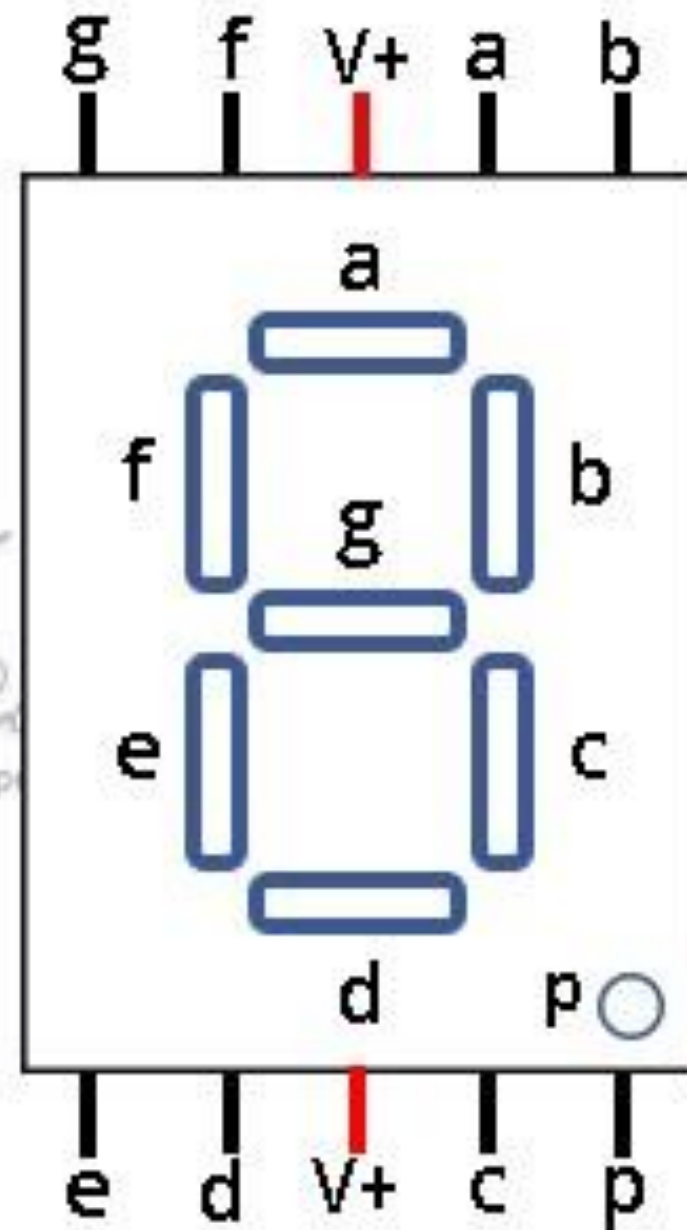
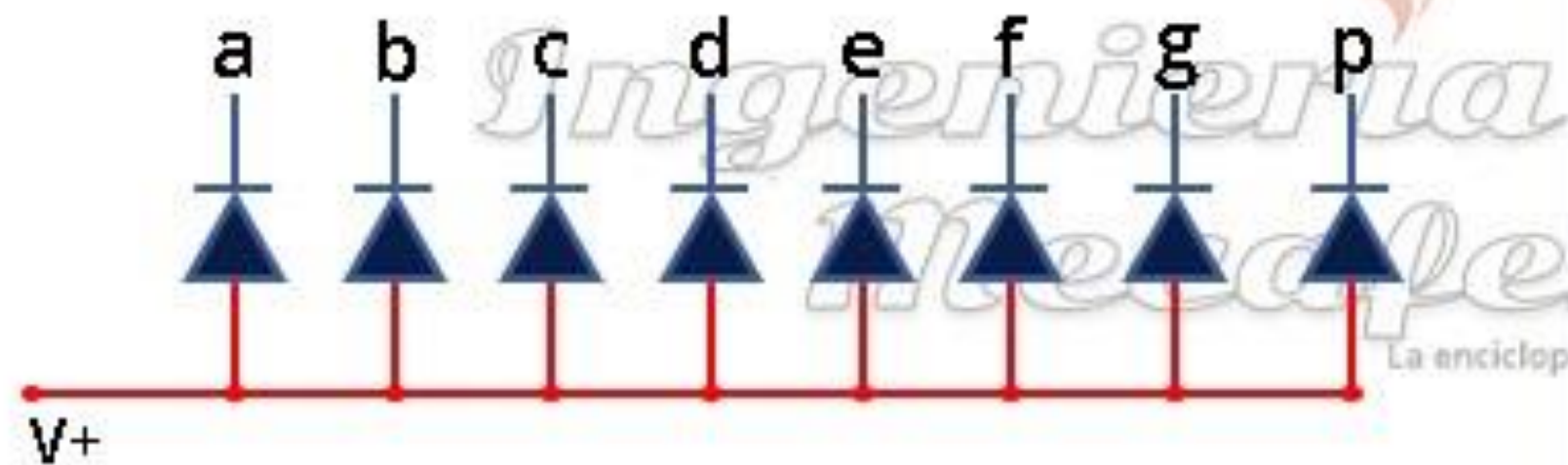
- Como parte de la aportación se realizó la implementación física a un display de 7 segmentos, a partir del código BCD. Este circuito se realizó a partir del decodificador "SN74LS47N". En este circuito se requirió el uso de 7 resistencias de 200 Ohms como medida de protección. Se uso también dip switch 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







# Display 7 segmentos ánodo común





# CONCLUSIONES:

- 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.

# 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\_OTE\_2019\_847-19-121-5. <https://intef.es/wp-content/uploads/2019/03/MIT-App-Inventor-2.pdf>