



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA: INGENIERÍA EN TELECOMUNICACIONES

NRC: 9434

ASIGNATURA: CIRCUITOS DIGITALES

TRABAJO DE INVESTIGACIÓN #1

INTEGRANTES:

- MONTIEL DÍAZ DEYDER ANDRÉS
- SALAZAR DUQUE DOMENICA ALEJANDRA
- VILLACIS NARVAEZ JONATHAN STEVEN

DOCENTE: ING. DARWIN ALULEMA

TEMA: CONVERTOR DE SISTEMAS DE NUMERACIÓN DESARROLLADO EN APP INVENTOR

Junio, 2020

ÍNDICE GENERAL

PLANTEAMIENTO DEL PROBLEMA	5
1.1. CONTEXTO Y ANTECEDENTES	5
1.2. FORMULACIÓN DEL PROBLEMA	7
1.3. HIPÓTESIS DE INVESTIGACIÓN	7
OBJETIVOS	8
2.1 OBJETIVO GENERAL	8
2.2 OBJETIVOS ESPECÍFICOS	8
ESTADO DEL ARTE	8
3.1 Teaching Automation and Control with App Inventor Applications	8
3.2 An automated evaluation system for App Inventor Apps	9
MARCO TEÓRICO	10
4.1. SISTEMAS DE NUMERACIÓN	10
4.1.1. SISTEMA DE NUMERACIÓN DECIMAL.	10
4.1.2. SISTEMA DE NUMERACIÓN BINARIO.	11
4.1.3. SISTEMA DE NUMERACIÓN OCTAL	12
4.1.4. SISTEMA DE NUMERACIÓN HEXADECIMAL	13
4.1.5. CONVERSIÓN DE CUALQUIER BASE A CUALQUIER BASE	14
4.1.5.1 De decimal a cualquier base.	14
4.1.5.2 De cualquier base a decimal.	16
4.1.5.3 De binario a octal y hexadecimal	16
4.1.5.4 De octal o hexadecimal a binario	17
4.1.5.5 De octal a hexadecimal o viceversa	17
4.1.6. EQUIVALENCIA DE LOS SISTEMAS DE NUMERACIÓN	17
4.2. MIT APP INVENTOR	18
DIAGRAMAS	19

LISTA DE COMPONENTES	20
MAPA DE VARIABLES	20
7.1 COMPONENTES VISUALES	20
7.1.1 PANTALLA INTRO	20
7.1.2 PANTALLA INGRESO DATOS	21
7.1.3 PANTALLA RESULTADO	23
7.2 COMPONENTES NO VISUALES	24
7.2.1 PANTALLA INTRO	24
7.2.2 PANTALLA CONVERTOR	24
7.2.2 PANTALLA RESULTADO	25
EXPLICACIÓN DEL CÓDIGO FUENTE	26
8.1 PANTALLA DE INICIO	26
8.1.2 Iniciando desde Screen 1	26
8.2 PANTALLA DE CONVERSIÓN	26
8.2.1 Inicialización de variables:	26
8.2.2 Inicialización de parámetros:	27
8.2.3 Seleccionar el sistema de entrada	28
8.2.3.1 Función "DatoNum"	28
8.2.4 Selección del sistema de salida	28
8.2.4.1 Función validación	29
8.2.4.1.1 Función InicializarSistema	29
8.2.4.2 Función CondSystem	32
8.2.5 Botón btnConvertirNumero	32
8.2.5.1 Función ConvDecimal	33
8.2.5.1.1 Función NBaseDecimal	33
8.2.5.1.2 Función DecimalEntero	34
8.2.5.1.3 Función DecimalFraccionario	35
8.2.5.2 Función ConvGlobal	36

8.2.5.2.1 Función SepararIntermedioBO	37
8.2.5.2.2 Función BinOctEntero	38
8.2.5.2.3 Función BinOctFraccionario	38
8.2.5.2.4 Función SepararIntermedioH	39
8.2.5.2.4 Función HexEntero	39
8.2.5.2.4 Función HexFraccionario	41
8.3 PANTALLA DE RESULTADO	41
8.4 BOTON ATRÁS	42
DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN	42
Macintosh (con procesador Intel): Mac OS X 10.5, 10.6	42
CONCLUSIONES	43
RECOMENDACIONES	44
CRONOGRAMA	44
BIBLIOGRAFÍA	45
ANEXOS	46
15.1 MANUAL DE USUARIO	46
15.1.1 Requerimientos	46
15.1.2 Instalación	47
15.1.3 Ejecución del programa	50

1. PLANTEAMIENTO DEL PROBLEMA

1.1. CONTEXTO Y ANTECEDENTES

Desde 1936, con la creación del primer computador, se ha pensado la forma de administrar los recursos informáticos. Las primeras máquinas de cálculo estaban constituidas por medio de tubos al vacío y operaban en lenguaje máquina. Cubría una gran infraestructura por lo se optó por la optimización de espacio gracias al desarrollo de los circuitos integrados CI, con el propósito de realizar operaciones lógicas de computador con una mayor eficacia que un computador analógico.

Su implementación dio paso al desarrollo de lenguajes de programación. con la intención de ejecutar instrucciones de forma simultánea, evitando que el usuario tenga que escribir y ejecutar una por una de forma manual. (Historia de los sistemas operativos, 2009)

Por los años 80, época donde la digitalización ya no era vista como extraña, se procedió al desarrollo de un programa constituido por un conjunto de programas, caracterizado por ejecutar largas instrucciones, emisión y recepción de información entre usuario y sistema, ejecutado en tiempo real, y susceptible a modificaciones. Por ello nace el sistema operativo, denominado el cerebro del computador, ya que se encarga de ser un intermediario entre hardware - software para la ejecución de procedimientos internos. Su paquete de datos se puede clonar haciendo que se pueda ejecutar en diversos computadores haciendo que los usuarios puedan gestionar las mismas interfaces y ventanas gráficas con diferente infraestructura. En 1982, Windows lanza su primer sistema operativo, coexistiendo con MS-DOS hasta que años más tarde su firma se independizó. A partir de ello, se han ido originado otros sistemas operativos como Linux, MAC, Unix, etc. (González, 2008)

En el siglo reciente, los sistemas operativos existentes u otros recientes han ido implementando en otros dispositivos como celulares y tablets. conservando sus mismas propiedades.

Ahora sus versiones se aplican para teléfonos móviles, denominados sistemas operativos móviles, han ido surgiendo por la necesidad de establecer una conexión constante a internet. En 2005, se originó el primer sistema operativo para móviles llamado Android, que se hizo conocer gracias a Google, naciendo el desarrollo de aplicaciones con un costo menor o nulo. Ejemplos como calendarios personalizados, sitios de noticias, correo electrónico, etc.; son desarrollados por terceros y lo pueden realizar en base a conocimientos básicos de desarrollo móvil.

(xatakandroid, 2008).

En carreras de Tecnologías de la Información y Comunicación, Electrónica y Automatización, Telecomunicaciones u otras afines, trabajan con sistemas de numeración de cualquier base donde sus aplicaciones se evidencian como la transmisión de datos, control de decisiones y seguimiento de instrucciones, diseño de plataformas, páginas web y herramientas iterativas. Las conversiones entre sistemas se los puede realizar a mano por medio de operaciones sucesivas de multiplicación, pero es propenso a cometer errores y los tiempos de respuesta de cálculo aumentan de acuerdo al tamaño del número descrito, y de acuerdo a las siguientes equivalencias entre los sistemas.

Así que por ello se necesita una herramienta en la cual permita realizar conversiones por medio de operaciones matemáticas a menor tiempo. Existen páginas web que permiten realizar conversiones de forma automática, sin embargo, muchas de ellas están susceptible a errores ya que solo se especializan en operar con su parte entera. Se puede realizar programas en cualquier lenguaje de programación, pero su entorno es muy básico, limitado a la implementación de gráficas, y para la ejecución del programa requiere su

compilador, siendo poco necesario para la ejecución de un pequeño programa. Afortunadamente existe una plataforma en la cual se puede crear aplicaciones con las características de Android, con diseño personalizado y modelo de programación basada en diagramas de bloques, siendo eficaz para estudiantes que recién comienzan a programar. Las conversiones se los realizan en cuestión de segundos y el programa se podrá usar en cualquier momento ya que por ello permanecerá guardada en el teléfono móvil, sin necesidad de que el dispositivo esté conectado a internet.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo influye la aplicación móvil, Conversor de Sistemas de Numeración, como herramienta didáctica para los estudiantes de Tercer nivel de la carrera de Ingeniería en Telecomunicaciones, de la Universidad de las Fuerzas Armadas ESPE, en la sede Sangolquí, en el periodo mayo – septiembre de 2020?

1.3. HIPÓTESIS DE INVESTIGACIÓN

Hi: Los estudiantes de Tercer Nivel de la carrera de Ingeniería en Telecomunicaciones, que utilizan App Inventor, para el desarrollo de aplicaciones, obtienen mejores conversores numéricos.

HO: Los estudiantes de Tercer Nivel de la carrera de Ingeniería en Telecomunicaciones, que no utilizan App Inventor, para el desarrollo de aplicaciones, obtienen mejores conversores numéricos.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar la aplicación móvil, Conversor de Sistemas de Numeración, a través de la plataforma App Inventor, como herramienta didáctica para los estudiantes

de Tercer nivel de la carrera de Ingeniería en Telecomunicaciones, de la Universidad de las Fuerzas Armadas ESPE, en la sede Sangolquí, en el periodo mayo – septiembre de 2020.

2.2 OBJETIVOS ESPECÍFICOS

- Investigar las diferentes formas de conversión entre sistemas de numeración.
- Desarrollar la lógica de programación por medio de algoritmos y diagramas de bloques.
- Motivar el uso de la plataforma App Inventor para el diseño de futuras aplicaciones.

3. ESTADO DEL ARTE

3.1 Teaching Automation and Control with App Inventor Applications

Autora: Moura Olivera

Año: 2015

Se enfoca en el desarrollo de dos aplicaciones para sistema Android, enfocado para los cursos de Automatización Industrial y control, particularmente en control de procesos, lógica binaria y control programado. Inicialmente se describió que se basó en conocimientos básicos de programación y se los plasmó en circuitos eléctricos con interruptores y describió acerca de los operadores booleanos como las operaciones AND, OR y NOT.

Esta investigación se desarrolló en Tallinn University of Technology, Tallinn, Estonia.

3.2 An automated evaluation system for App Inventor Apps

Autores: Yue Li, Yiping Pan, Wensheng Liu, Xingming Zhang

Año: 2018

El presente artículo tiene como finalidad diseñar un método de puntuación que permita a los maestros de las escuelas K-12 evaluar de forma rápida y sencilla las aplicaciones de App Inventor que les mandan a desarrollar a sus estudiantes. Este método de puntuación es completamente automatizado y está basado en TF-IDF. La aplicación y el desarrollo del método fueron hechos en el país de China y estuvo a cargo de los autores de artículo.

3.3 Punto de vista de los estudiantes sobre el uso de MIT App Inventor en educación. 2019 42a Convención Internacional sobre Tecnología de la Información y la Comunicación, Electrónica.

Autor: St. Georgiev, T. (2019).

Año: 2019

El presente artículo tiene la finalidad de dar una apertura a la visión respecto a la plataforma App Inventor viendo las ventajas de su uso, junto con las desventajas o cosas en contra que tiene la aplicación, este artículo es hecho por 73 estudiantes de la universidad de Rusia acerca de la aplicación APP Inventor y su uso para los diferentes niveles existentes de educación , también se describe la diferencia con otros lenguajes de programación y se realiza el análisis correspondiente

4. MARCO TEÓRICO

4.1. SISTEMAS DE NUMERACIÓN

4.1.1. SISTEMA DE NUMERACIÓN DECIMAL.

Es aquel sistema que se conforma por los siguientes diez dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, y 9. Este sistema de numeración es el que más se utiliza desde un punto de vista cotidiano y ordinario, inclusive la suma de los dedos de las manos posee esa cantidad. Este sistema también se le conoce como de base diez.



FIGURA 1. Ejemplo del empleo de números decimales

Fuente: <https://www.creerenti.com/2017/06/23/las-manos/>

Para contar un número decimal, se comienza aumentando una unidad desde el 0 hasta el 9, por siguiente se añade a la izquierda un 1 y el dígito anterior regresa a 0, y de ahí nuevamente se cuenta desde la primera posición. De la misma forma, después de llegar al 99 se añade un 1 y los demás dígitos se reinician a 0. Así seguirá el conteo hasta llegar a un número deseado.

0	10	20	...	90	100
1	11	21	...	91	101
2	12	22	...	92	102
3	13	23	...	93	103
4	14	24	...	94	104
5	15	25	...	95	...

6	16	26	...	96	
7	17	27	...	97	
8	18	28	...	98	
9	19	29	...	99	

TABLA 1. Conteo de números decimales

4.1.2. SISTEMA DE NUMERACIÓN BINARIO.

Es aquel sistema que se conforma por los siguientes dos dígitos o bits: 0 y 1. Es muy útil para trabajar en sistemas digitales debido a que se puede representar dos estados distintos y constituye los principales elementos de lenguaje máquina. Este sistema se conoce de base 2.

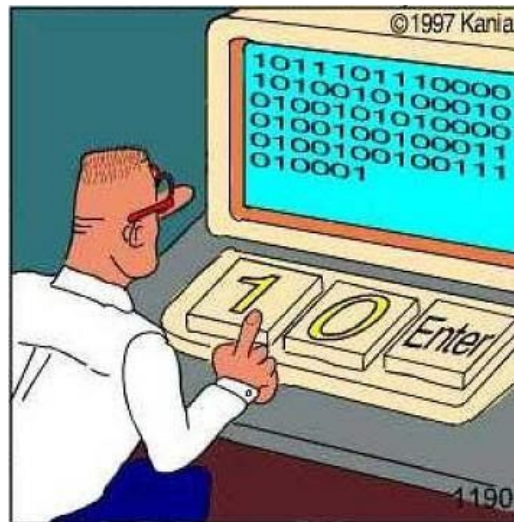


FIGURA 2. Ejemplo del empleo de números binarios

Fuente: <https://lignux.com/lenguaje-maquinal/>

Para su conteo, los bits comienzan desde 0 y el bit menos significativo aumenta a 1, posteriormente se añade un 1 a la izquierda del bit contado y

ese mismo se reinicia a 0. Así mismo el conteo comienza desde el bit inferior hacia el superior.

0	10	100	1000	...
1	11	101	1001	...
		110	1010	..
		111	1011	...
			1100	...
			1101	...
			1111	..

TABLA 2. Conteo de números binarios

4.1.3. SISTEMA DE NUMERACIÓN OCTAL

Es aquel sistema que se conforma por los siguientes ocho dígitos: 0, 1, 2, 3, 4, 5, 6 y 7. La posición de los dígitos se expresa en potencias de 8. Se deriva del sistema binario ya que representa un caso específico de su base, y también forma parte del sistema hexadecimal ya que sus dígitos complementan a mitad del mismo, usándose como alternativa en códigos binarios. Este sistema se conoce de base 8.

Para su conteo, los dígitos comienzan desde 0 y el dígito menos significativo aumenta de 1 a 7 con paso 1, posteriormente se añade un 1 a la izquierda del bit contado y ese mismo se reinicia a 0. Así mismo el conteo comienza desde el dígito inferior hacia el superior.

0	10	...	70	80
1	11	...	71	81
2	12	..	72	82
3	13	...	73	83
4	14	...	74	...
5	15	...	75	...
6	16	...	76	...
7	17	...	77	...

TABLA 3. Conteo de números octales.

4.1.4. SISTEMA DE NUMERACIÓN HEXADECIMAL

Es aquel sistema que se conforma de 16 dígitos como: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. La posición de los dígitos se expresa en potencias de 16. Este sistema es utilizado comprimir grandes cadenas binarias por medio de sus equivalencias, pudiendo representar elementos visuales como el caso de la generación de los códigos de colores para diseñar una página web.

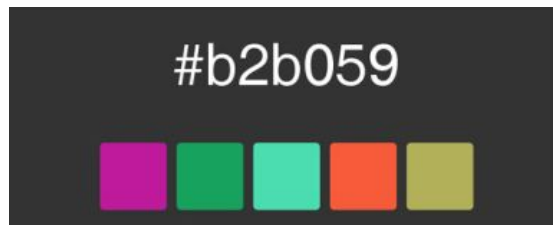


FIGURA 3. Ejemplo de empleo de los sistemas hexadecimales.

Fuente:

<https://arquitectura07.webnode.com.co/bases-numericas/aplicacion-en-la-informatica>
L

De la misma forma, el conteo de esos números sigue la misma metodología como los demás sistemas de numeración, solo que con más elementos.

0	10	20	...	90	100
1	11	21	...	91	101
2	12	22	...	92	102
3	13	23	...	93	103
4	14	24	...	94	104
5	15	25	...	95	...
6	16	26	...	96	...
7	17	27	...	97	...
8	18	28	...	98	...
9	19	29	...	99	...

A	1A	2A	...	9A	...
B	1B	2B	...	9B	...
C	1C	2C	...	9C	...
D	1D	2D	...	9D	...
E	1E	2E	...	9E	...
F	1F	2F	...	9F	...

TABLA 4. Ejemplo de empleo de los sistemas hexadecimales.

4.1.5. CONVERSIÓN DE CUALQUIER BASE A CUALQUIER BASE

4.1.5.1 De decimal a cualquier base.

Para ello, se requiere dividir cualquier número decimal para la base a convertir y anotar los residuos de esa división de derecha a izquierda hasta que su cociente sea 0.

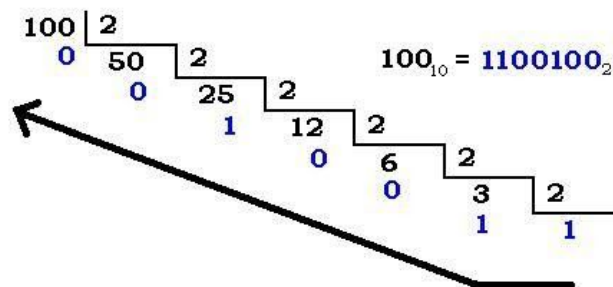


FIGURA 4. Ejemplo de conversión de un número decimal a otro de base 2 (binario).

Fuente: https://es.wikipedia.org/wiki/Sistema_binario

Esto se puede explicar en el siguiente diagrama de flujo;

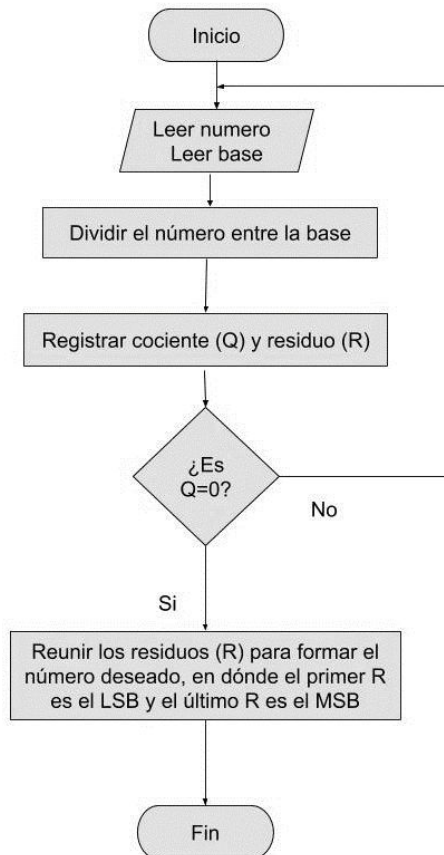


FIGURA 5. Diagrama de flujo que describe la conversión de un número entero o fraccionario del sistema decimal a otro.

En caso de existir parte fraccionaria de ese número, se separa de la parte entera y se procede a realizar lo siguiente. La parte fraccionaria se multiplica por la base y se va agrupando la parte entera de su resultado. Se debe realizar este proceso hasta que la parte fraccionaria sea cero.

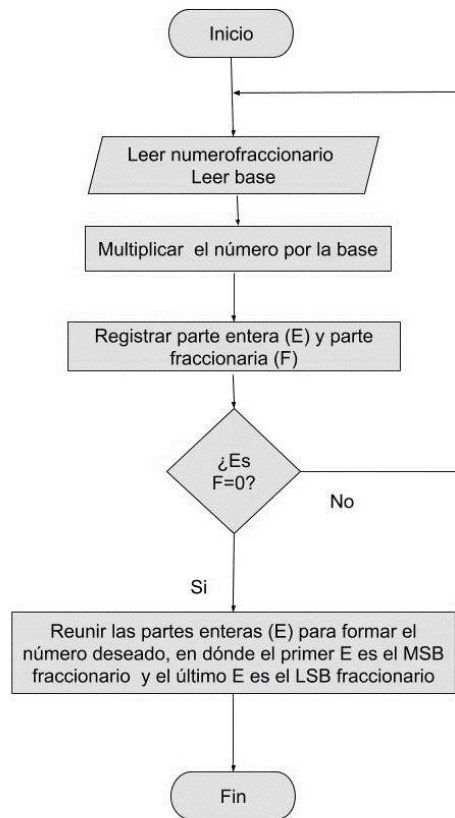


FIGURA 6. Diagrama de flujo que describe la conversión de un número fraccionario del sistema decimal a otro.

4.1.5.2 De cualquier base a decimal.

Su proceso consiste en una sumatoria de cada dígito multiplicado por su base elevado a la posición en la que se encuentra. Si existen dígitos fraccionarios, se toman con sus posiciones negativas comenzando por -1 de la izquierda del punto decimal derecha.

4.1.5.3 De binario a octal y hexadecimal

Este proceso es sencillo debido a que toca seleccionar grupos de bits. Para convertir de binario a octal, se debe seleccionar grupos de 3 bits de derecha

a izquierda en caso de la parte entera y de izquierda a derecha en caso de existir parte fraccionaria, estos grupos transformar a su dígito octal equivalente. El mismo proceso se lo realiza para convertir de binario a hexadecimal solo que los grupos formados son de 4 bits. En caso de faltar dígitos, se los rellena con 0.

4.1.5.4 De octal o hexadecimal a binario

El procedimiento para esta conversión es muy simple y consiste en convertir cada dígito hexadecimal u octal a su equivalente en binario, y al final juntarlos en un solo número binario.

Ejemplo:

$$1E_{16} = 0001\ 1110_2$$

4.1.5.5 De octal a hexadecimal o viceversa

No existe un método directo por lo que es más recomendable pasar ese número al sistema binario y después transformar al sistema deseado por medio de la agrupación de bits.

4.1.6. EQUIVALENCIA DE LOS SISTEMAS DE NUMERACIÓN

El resumen visto en los sistemas binarios se describe en la siguiente tabla:

Decimal	Binario	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2

3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

TABLA 5. Equivalencia de números en el sistema decimal, hexadecimal y binario.

4.2. MIT APP INVENTOR

La aplicación App Inventor es un software creado y desarrollado por Google, que permite a las personas crear sus propias aplicaciones para el sistema Android. El lenguaje que posee este software se lo puede encontrar de forma gratuita, además que su comprensión es bastante sencilla. aunque las aplicaciones son simples permiten cubrir varias necesidades básicas en los dispositivos móviles.

Utilizar este tipo de aplicaciones posee varias ventajas como lo son el poder programar de forma sencilla, en el caso de App Inventor el usuario puede programar de forma intuitiva utilizando bloques de programación. Además, la accesibilidad es bastante amplia ya que se puede acceder a la aplicación en cualquier momento y lugar, siempre y cuando se tenga conexión a internet.

En conclusión, programar con App Inventor facilita el proceso de programación permitiendo crear aplicaciones simples pero funcionales sin la necesidad de saber código de programación.

El método de programación que utiliza App Inventor es un lenguaje de programación llamado 'lenguaje de programación en bloques'. La programación en bloques consiste en incrustar piezas prediseñadas de tal forma que juntas generan una lista de acciones que se debe seguir para solucionar un problema. Este tipo de programación es una manera práctica y visual que se utiliza para programar en entornos gráficos.

Esta plataforma maneja posee un controlador de eventos. En los lenguajes de programación un evento es considerado como un suceso que pasa y activa el código desarrollado, para que este comience a correr. La manera en que el computador maneja un evento será de la forma en que el programador la desarrollado para que ese evento se llevará a cabo.

Además de los eventos, App inventor posee varias funcionalidades como son: Control, Logic, Math, Text, etc. Las funciones en App Inventor son bloques de código cuyo propósito es de ejecutar lo que programador desea realizar. Las funciones integradas son funcionalidades básicas que aceptan un input y output para la ejecución del programa.

Realizar el desarrollo de una aplicación utilizando el lenguaje de programación de App Inventor, no presenta mayores complicaciones, ya que las funciones básicas que se disponen en la plataforma son sencillas de implementar. Lo que sí se debe tener muy en cuenta es la lógica con la que se las utiliza y el manejo de los eventos para evitar errores de tipo lógicos.

5. DIAGRAMAS

La explicación del funcionamiento de la aplicación está dado por el siguiente diagrama de bloques

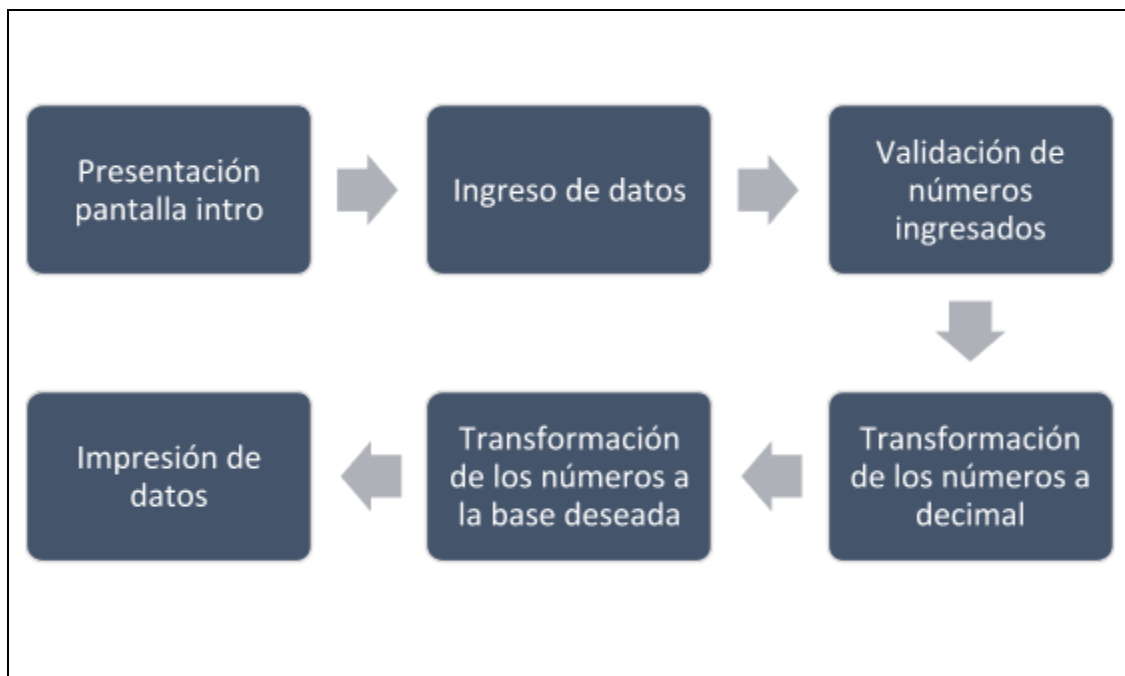


FIGURA 7. Diagrama de bloques que describe el manejo de datos en la aplicación.

6. LISTA DE COMPONENTES

Componente	Descripción
Teléfono móvil	Ejecución de la aplicación
Computadora	Desarrollo de la aplicación por medio de un navegador web

7. MAPA DE VARIABLES

7.1 COMPONENTES VISUALES

7.1.1 PANTALLA INTRO

Variable	Tipo	Funcionalidad
ArregloPrincipal	Arreglo	Contiene todos los elementos

		visuales de la pantalla ordenado de forma vertical
LogoApp	Imagen	Imagen diseñada para la aplicación
Separador1	Arreglo	Espacio entre líneas
TituloApp	Etiqueta	Describe el nombre de la App
Separador2	Arreglo	Espacio entre líneas
Versión	Etiqueta	Describe la versión de la App

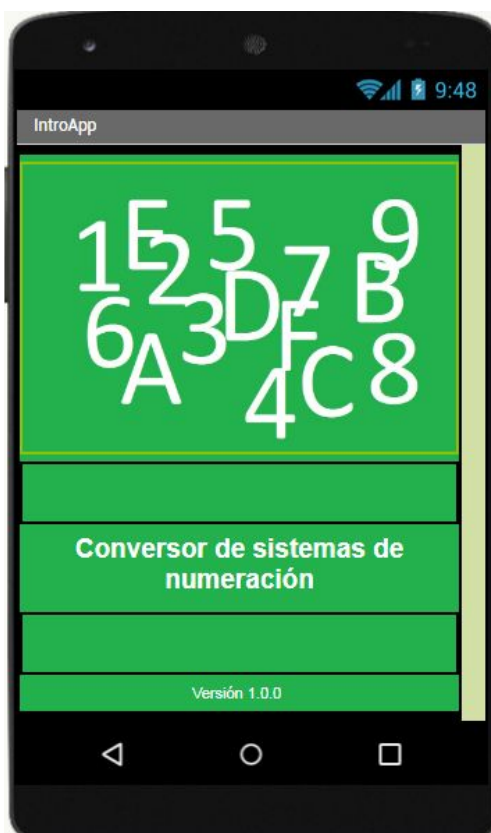


FIGURA 8. Pantalla intro de la aplicación

7.1.2 PANTALLA INGRESO DATOS

Variable	Tipo	Funcionalidad
lblTitulo	Etiqueta	Muestra el título del conversor

Espacio1	Arreglo	Espacio entre líneas
lblSeleccion1	Etiqueta	Texto que indica seleccionar el sistema de numeración del número ingresado
spnNum1	Lista desplegable	Desplegable donde contiene los sistemas de numeración a ingresar.
Espacio 2	Arreglo	Espacio entre líneas
lblIngreso	Etiqueta	Texto que indica que debe ingresar el número
txtNúmero1	Campo de texto	Cajón de texto donde se debe ingresar el número
lblError	Etiqueta	Texto que señala si el número ingresado es incorrecto de acuerdo al sistema de numeración original
Espacio3	Arreglo	Espacio entre líneas
lblSeleccion2	Etiqueta	Texto que indica seleccionar el sistema de numeración del número ingresado
spnNumO	Lista desplegable	Desplegable donde contiene los sistemas de numeración a convertir.
btnConvertir	Botón	Botón que ejecuta la acción de convertir los datos ingresados.
lblAvisoMismoNumero	Etiqueta	Texto que muestra si el sistema de numeración ingresado y a convertir son los mismos

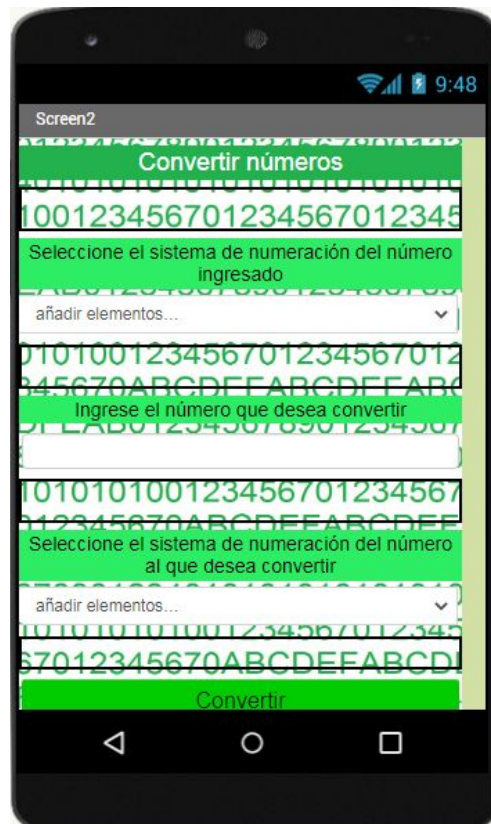


FIGURA 9. Pantalla de conversión de la aplicación

7.1.3 PANTALLA RESULTADO

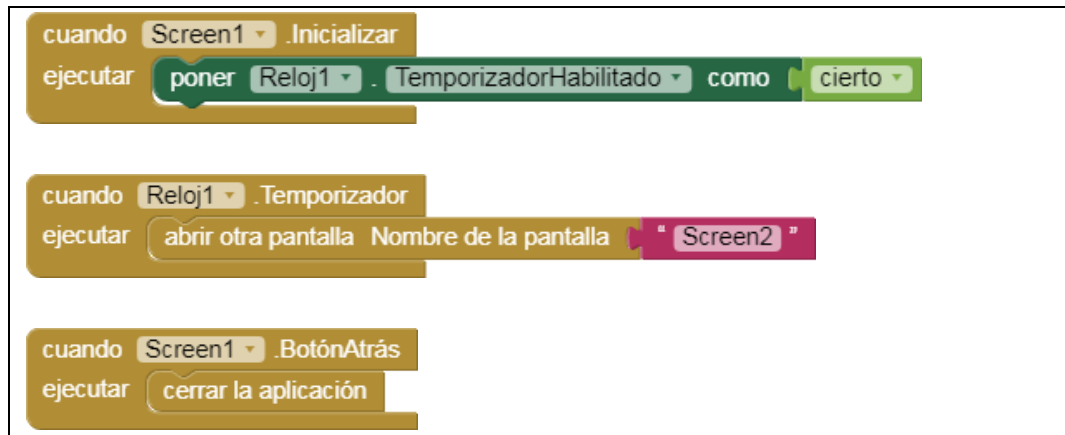
Variable	Tipo	Funcionalidad
ArregloPrincipal	Arreglo	Contiene todos los elementos visuales de la pantalla ordenado de forma vertical
lblResultado	Etiqueta	Texto que indica el resultado
txtNumeroR	Campo de texto	Cajón de texto que muestra el resultado
Separador1	Arreglo	Espacio entre líneas
btnRegresar	Botón	Botón que lleva a la pantalla de conversión
Separador2	Arreglo	Espacio entre líneas

8. EXPLICACIÓN DEL CÓDIGO FUENTE

8.1 PANTALLA DE INICIO

8.1.2 Iniciando desde Screen 1

Por defecto, la pantalla que se muestra primero cuando se inicia la App es “Screen 1”. Así que se escribió una instrucción para que la presentación de esa pantalla dure 3 segundos y luego se proceda a abrir la siguiente pantalla “Screen 2”.



8.2 PANTALLA DE CONVERSIÓN

8.2.1 Inicialización de variables:

Las variables presentadas para la ejecución del programa se encuentran en “Screen 2”. Las variables presentadas globales, por lo tanto, ante cualquier procedimiento o acción externa, se los debe inicializar primero a ellas. Estas variables son las entradas, la intermedia y las salidas del programa.

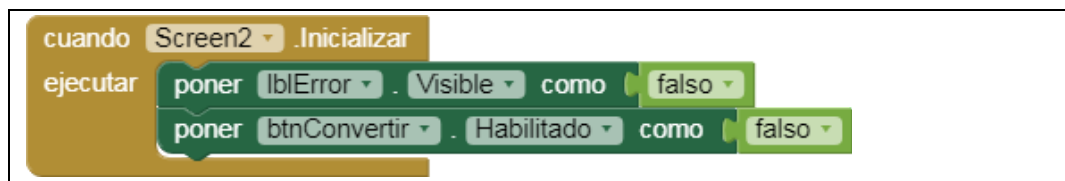
Como las variables de entrada y de salida también pueden contener letras, en el caso de números hexadecimales, se las inicializa con un

carácter, en este caso, un carácter de “0”. El valor de las bases y el intermedio son numéricos, por lo tanto, se las inicializará con valor de 0. Además, se necesitará verificar si el número ingresado posee parte fraccionaria, por lo que se inicializará con “cierto”.



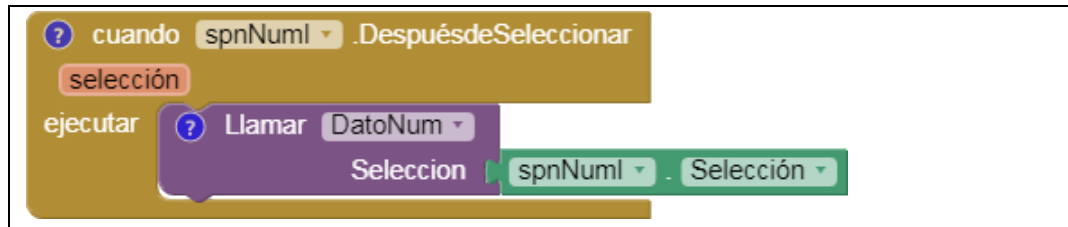
8.2.2 Inicialización de parámetros:

Cuando se inicializa la pantalla “Screen 2”, la etiqueta de error debe estar desactivada debido a que aún no se ha ingresado un dato. Con la misma razón, el botón “Convertir” debe estar inhabilitado.



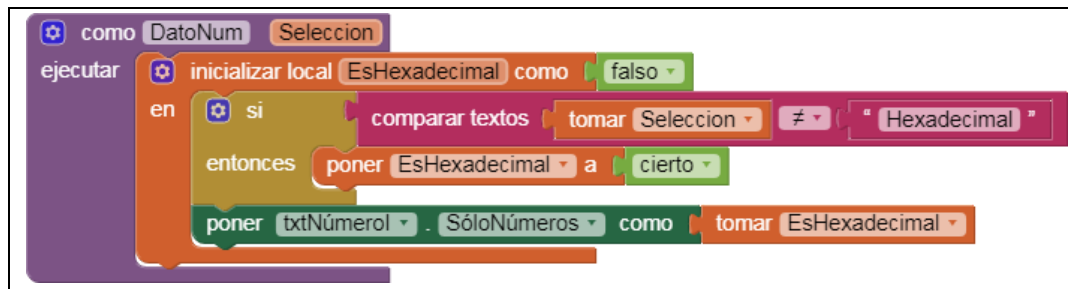
8.2.3 Seleccionar el sistema de entrada

Se procede a configurar el modo del teclado. Para ello, se llama a la siguiente función “DatoNum”. Esta recibe como argumento el ítem seleccionado de la lista desplegable



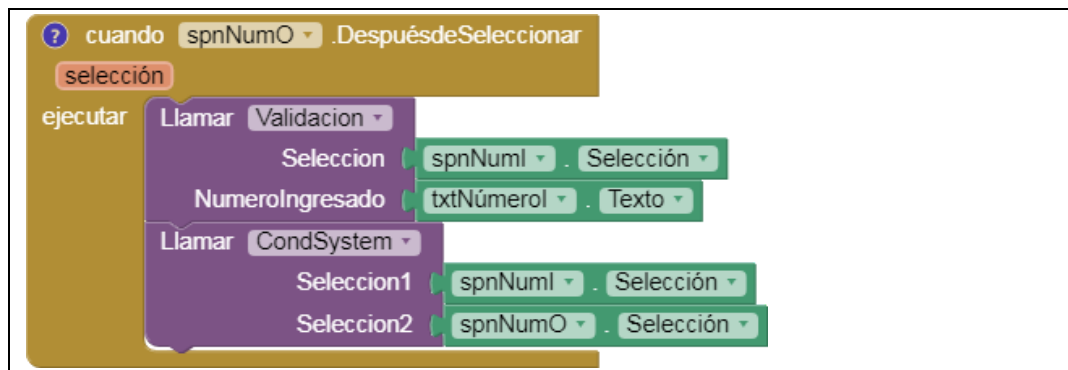
8.2.3.1 Función "DatoNum"

Sí el ítem seleccionado es "Hexadecimal", entonces el teclado permite ingresar caracteres. Si no, el teclado se determina a tipo numérico y solo podrán ingresar números. El valor de verdad estará determinado por la variable local "Es hexadecimal" y después esta se transfiere a su método.



8.2.4 Selección del sistema de salida

Una vez seccionado el sistema de salida, se procede con los siguientes procedimientos, donde uno se procede a verificar la legalidad del número. Sus parámetros son los sistemas de numeración seleccionados y el número ingresado.



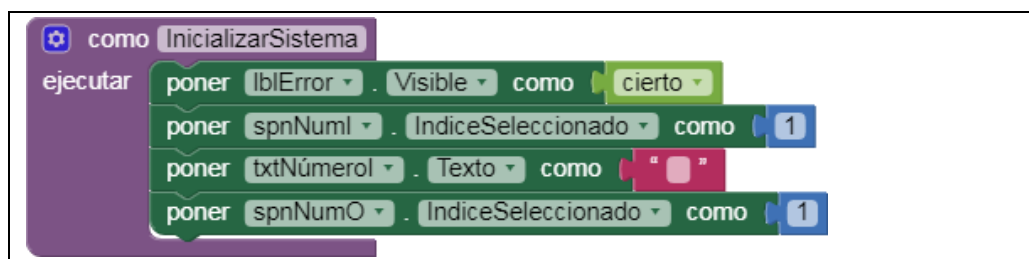
8.2.4.1 Función validación

Resulta ser el procedimiento más largo del programa debido a que se debe tomar cada dígito del sistema ingresado y el uso de sentencias break. El manejo de datos en App inventor es muy flexible ya que se puede tratar un número con determinadas cifras como una cadena de caracteres. Así que en este procedimiento se procede a analizar cifra por cifra hasta la longitud dada por su cadena, además analiza si el número es puramente entero o si posee una parte fraccionaria.

No hay problema en la introducción de números decimales debido a que se configuro el teclado haciéndolo numérico. Hay que recordar que e sistema binario solo se maneja con los dígitos 0 y 1, el octal con números del 0 al 7 y el hexadecimal todos los números decimales y las letras del alfabeto de la A hasta la F. Por lo que, si se introduce un número en la cuál mínimo tenga una cifra incorrecta, actúa la función InicializarSistema, caso contrario, se llama a la función ActivarBoton.

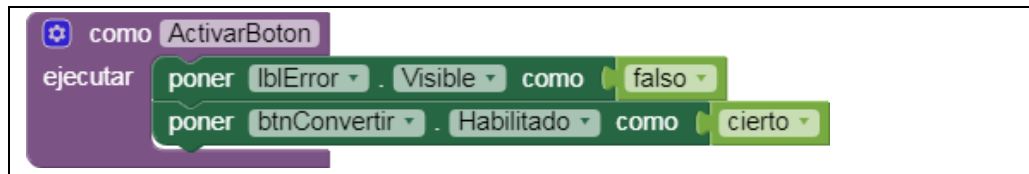
8.2.4.1.1 Función InicializarSistema

Se encarga de inicializar los valores de entrada a como lo estaban al momento de ingresar. Además, muestra la etiqueta "lblError". Esta función procederá a borrar todos los datos ingresados por medio del proceso Inicializar y aparece la etiqueta "lblError", donde informa al usuario que el número ingresado es incorrecto.

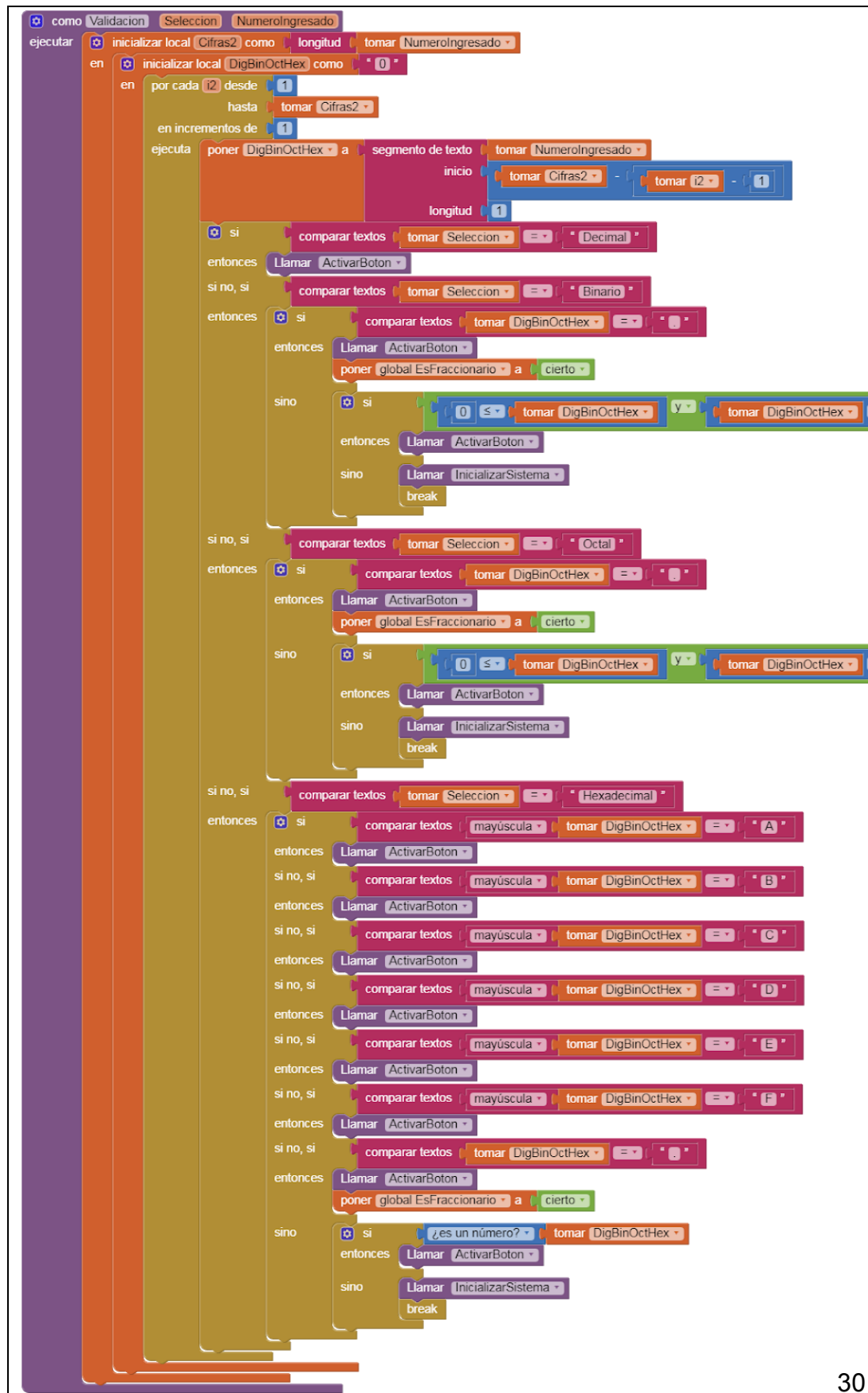


8.2.4.1.2 Función ActivarBoton

Si el número ingresado es correcto, se habilita el botón “btnConvertir” ya que es número es legible a realizar la operación de conversión. Si se Ingresa después de haber ingresado un número incorrecto, desaparece la etiqueta de error.

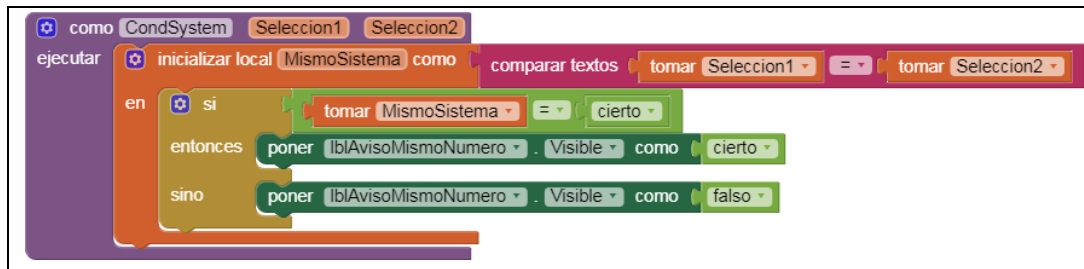


De acuerdo en cada sistema, estas funciones irán acompañadas en cada condicional dado, excepto en el decimal dado solo se deberá habilitar el botón de conversión.



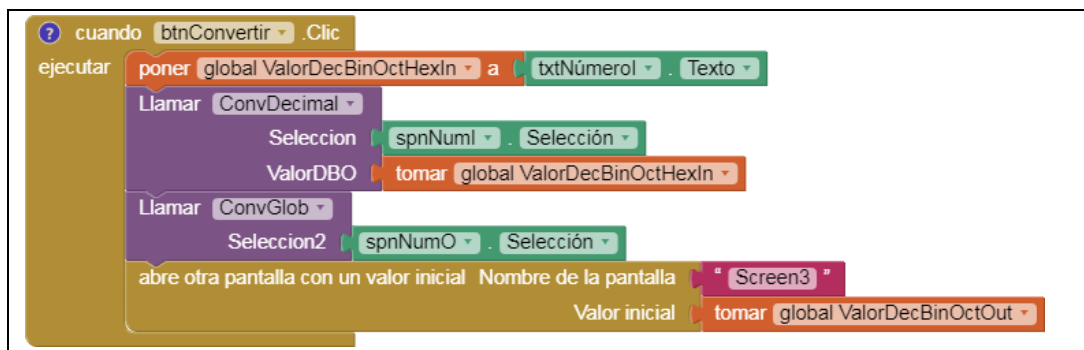
8.2.4.2 Función CondSystem

Esta función se encarga de notificar si se ingresaron dos sistemas de numeración iguales, mostrando su etiqueta "lblAvisoMismoNumero". Aunque no exista restricción alguna para la conversión, es importante darle conocimiento al usuario para agilizar procesos.



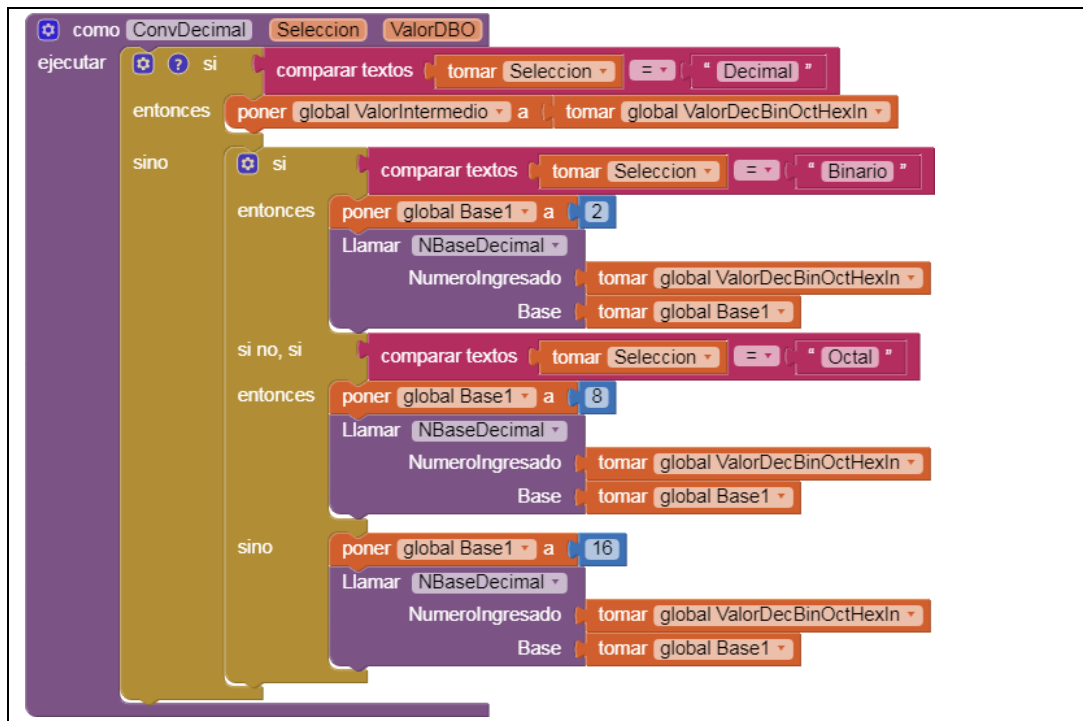
8.2.5 Botón btnConvertirNumero

Se describe como la acción más importante del programa razón de que mayoría de sus procedimientos y cálculos matemáticos se encuentran en él. Es dónde el dato ingresado pasa a contener en una variable del programa. Una vez validado los datos, al hacer click comienza con los siguientes procedimientos y ese valor global del proceso se transfiere a la siguiente pantalla.



8.2.5.1 Función ConvDecimal

Los números ingresados que no sean del sistema decimal pasa a convertirse a su mismo sistema. Se lo realiza debido a que cualquier conversión de cualquier base del sistema numérico posee el mismo proceso solo que el parámetro distinto es su base. Aplicar este proceso es mejor que ingresar su equivalencia binaria por cada dígito, en caso de conversión octal y hexadecimal a binario. Por ello, esta función va a retornar un valor intermedio, dicho valor global al comienzo de la descripción del código. Esta función está constituida por otras funciones de acuerdo a la base seleccionada.

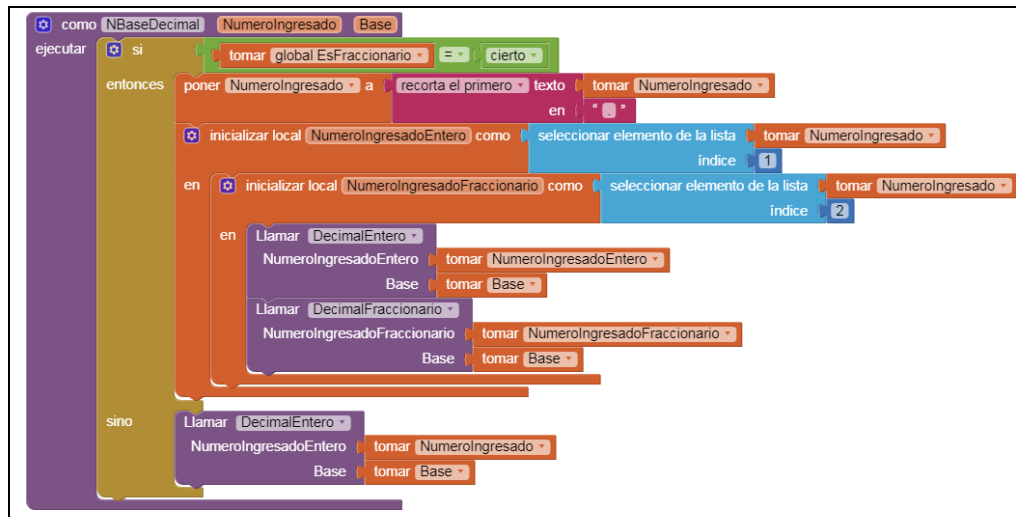


8.2.5.1.1 Función NBaseDecimal

De puede decir que esta función es general ya que consiste en separar la parte entera y la parte decimal del número ingresado. Esto se logra quitando el punto del número por lo cual este proceso retorna

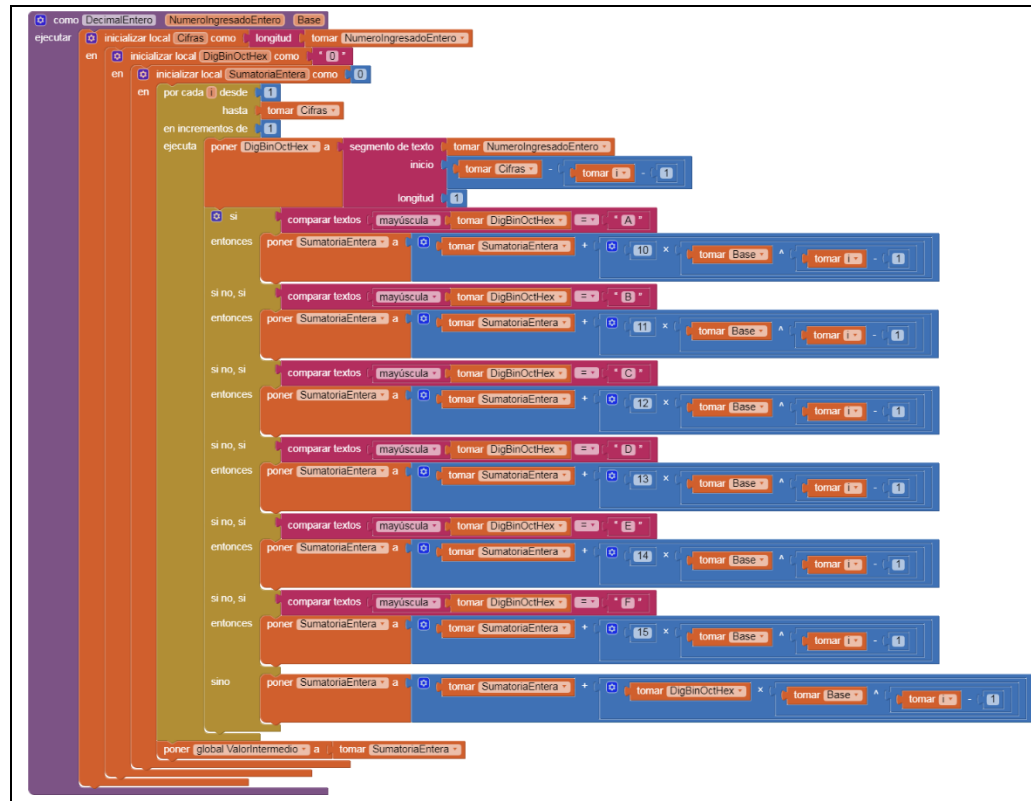
una lista con dos elementos: el número entero y su parte decimal. A partir de ello, se procede a pasar cada valor en una variable llamándolo por su índice, sirviendo para los procesos de conversión.

En caso de que el número ingresado sea entero, no se procede a realizar la separación debido retorna una lista de un elemento. Las listas no pueden operar directamente por lo que es necesario extraer sus elementos en otras variables.



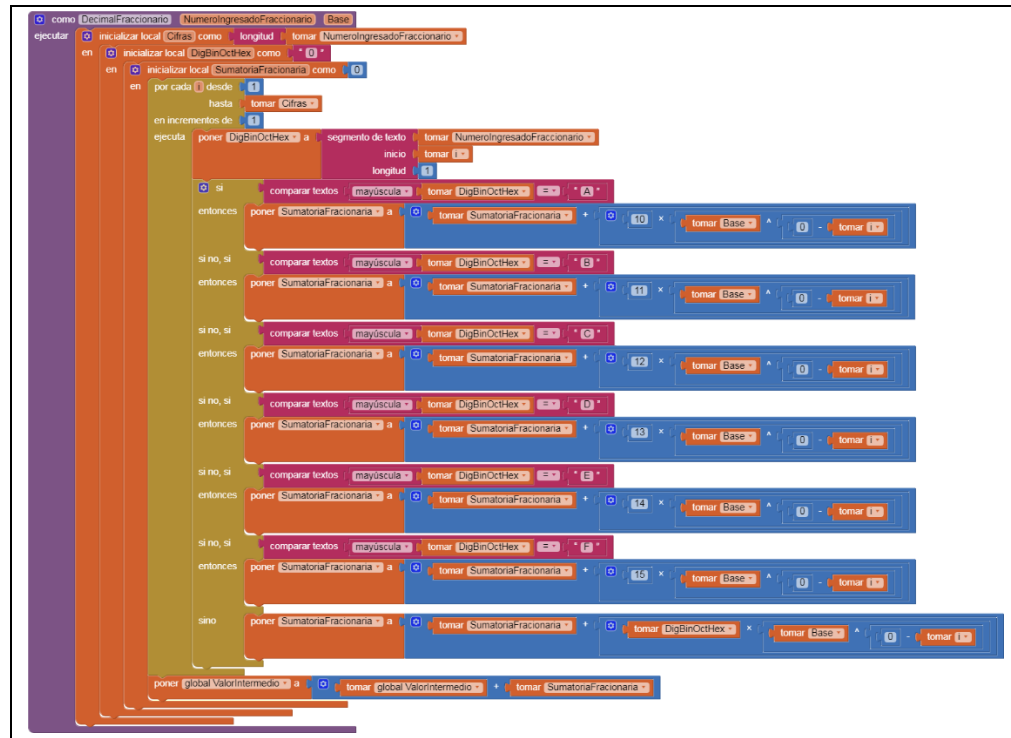
8.2.5.1.2 Función DecimalEntero

Se encarga de transformar la parte entera de cualquier número a base decimal. Se lo realiza tomando cada dígito multiplicándolo por su base elevado a su posición. Se tomará cada dígito por el procedimiento descrito anteriormente, que es seleccionar un elemento de un texto, debido a que el procedimiento incluye el análisis en números hexadecimales, condicionando sus valores equivalentes. Por último, el valor de la variable Resultado pasa al valor ValorIntermedio.



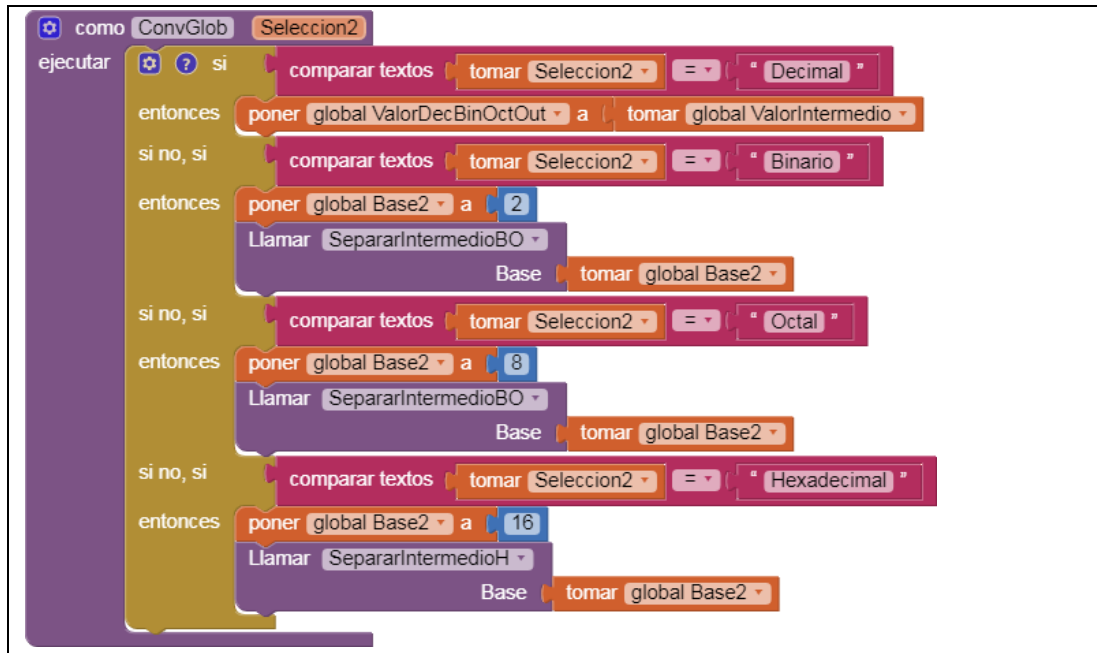
8.2.5.1.3 Función DecimalFraccionario

El proceso es parecido con respecto a los números enteros, sólo que sus posiciones son negativas. De llegar a este proceso, el valor Intermedio total se debe sumar este valor con el valor entero.



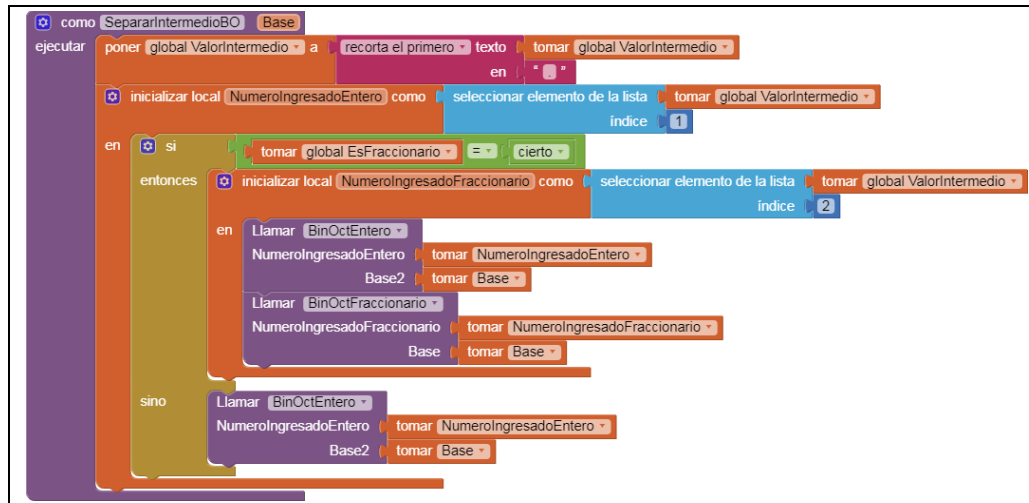
8.2.5.2 Función ConvGlobal

Se define como el último proceso de carácter general del programa. Este se encarga de convertir el valor intermedio al número deseado llamando específicamente a cada función. En aso de que el sistema de salida seleccionado des Decimal, el valor intermedio pasa a ser el valor de salida.



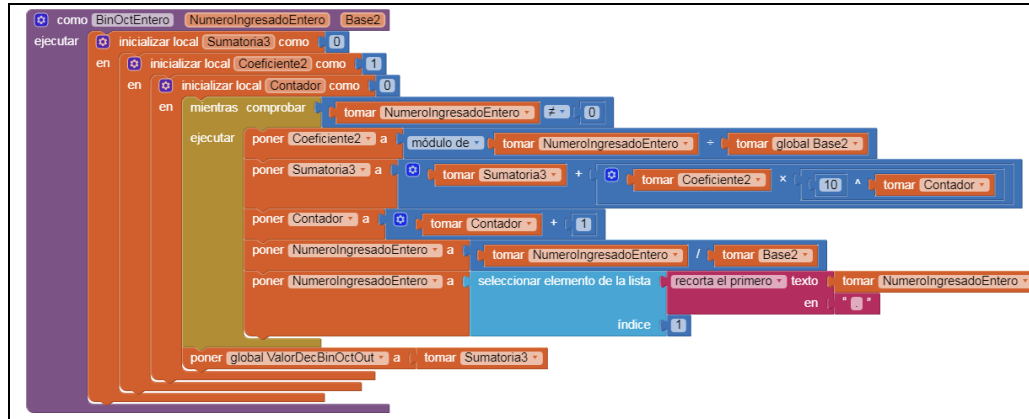
8.2.5.2.1 Función SepararIntermedioBO

Esta función se encarga de separar la parte entera y fraccionaria del valor intermedio, en caso de que existiera, previo a convertir al sistema de numeración binario u octal. Luego de ello, cada parte del número ingresa a la función que define el proceso de conversión



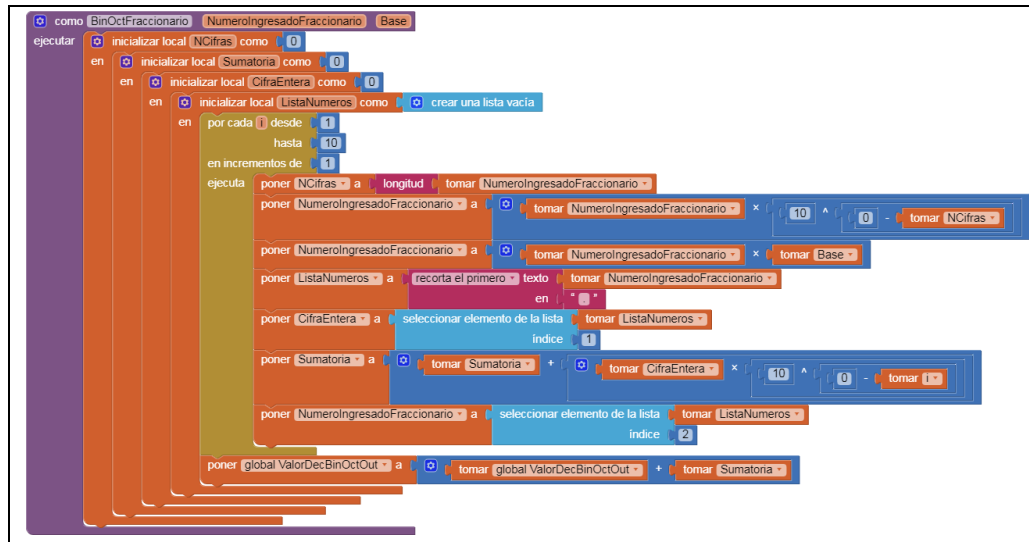
8.2.5.2.2 Función BinOctEntero

El proceso se encarga de realizar divisiones sucesivas con respecto a la base deseada. El residuo de esa división se multiplica por la potencia de diez elevado a la posición y se acumula en una variable hasta que el cociente sea cero. El valor acumulado pasa a la variable global ValorDecBinOctHexOut.



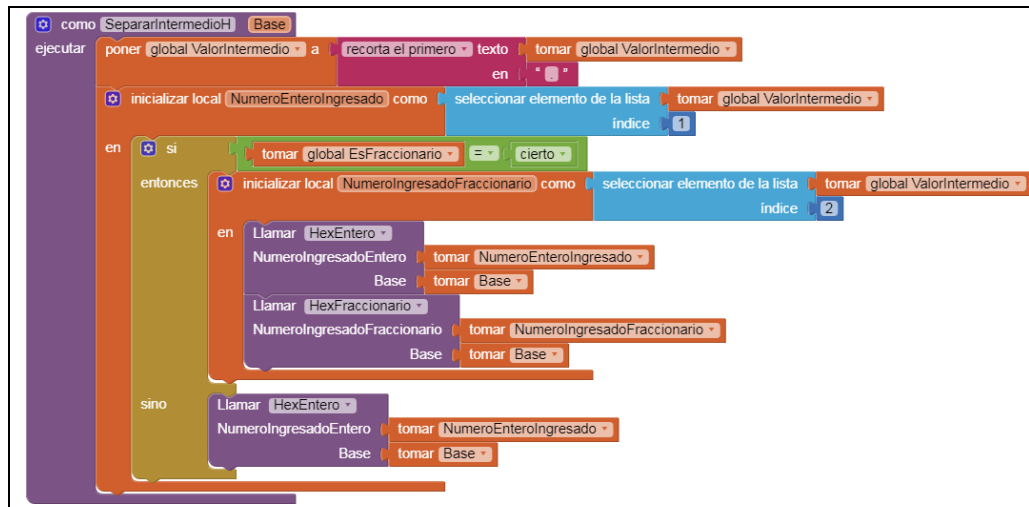
8.2.5.2.3 Función BinOctFraccionario

La parte fraccionaria ingresada ahora es un entero local. Por lo tanto, se multiplica hasta obtener un número fraccionario de parte entera 0. Luego, se procede a multiplicar ese número por su base y se toma su parte entera como dígito. Esto se repite sucesivamente hasta que su parte fraccionaria es 0 o en un número determinado de diez iteraciones.



8.2.5.2.4 Función SepararIntermedioH

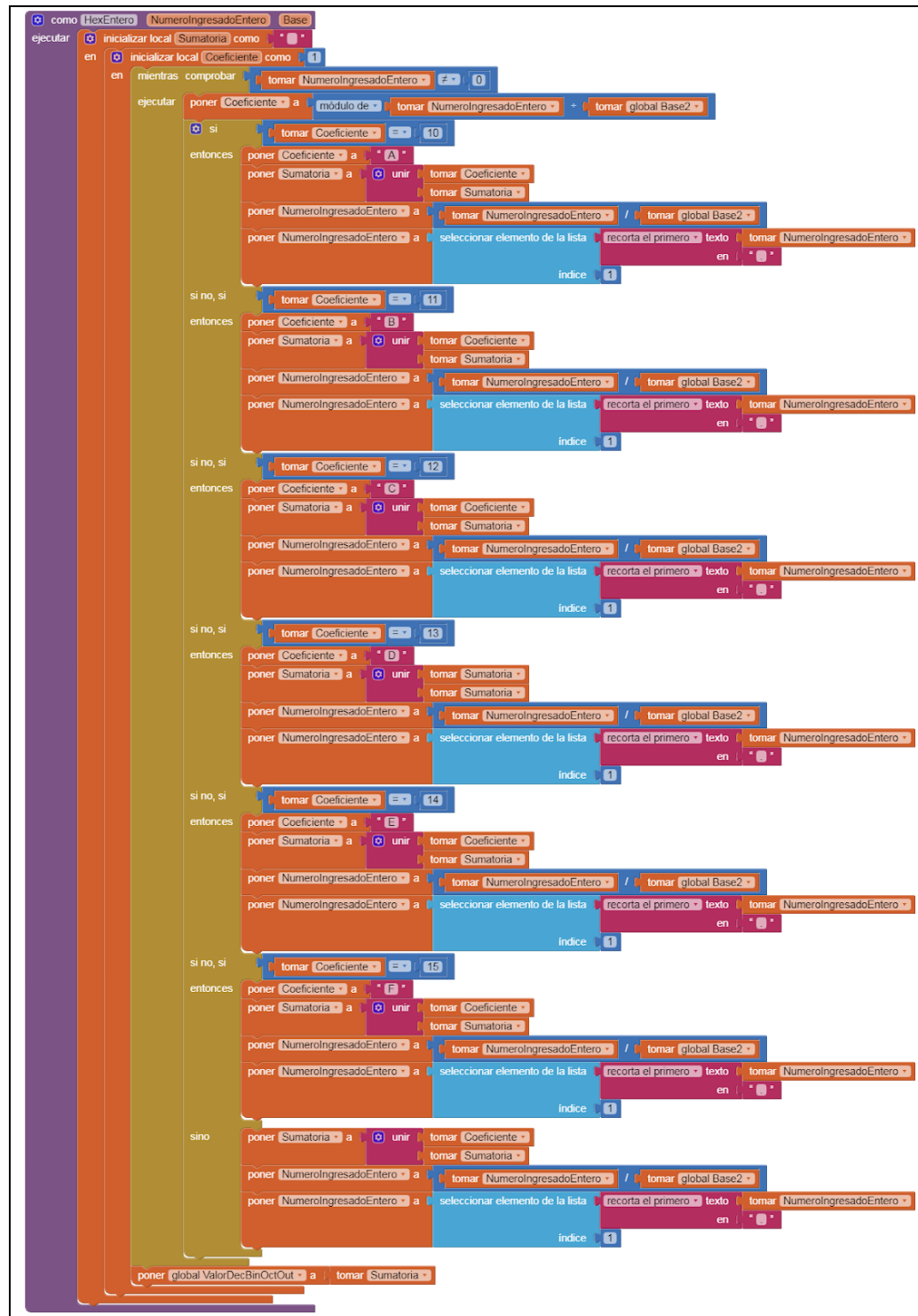
Esta función se ejecuta específicamente solo cuando desea el sistema hexadecimal como salida. Al igual que el proceso de conversión de los sistemas binario y octal, se procede a separar el número en su parte entera y fraccionaria si hubiere.



8.2.5.2.4 Función HexEntero

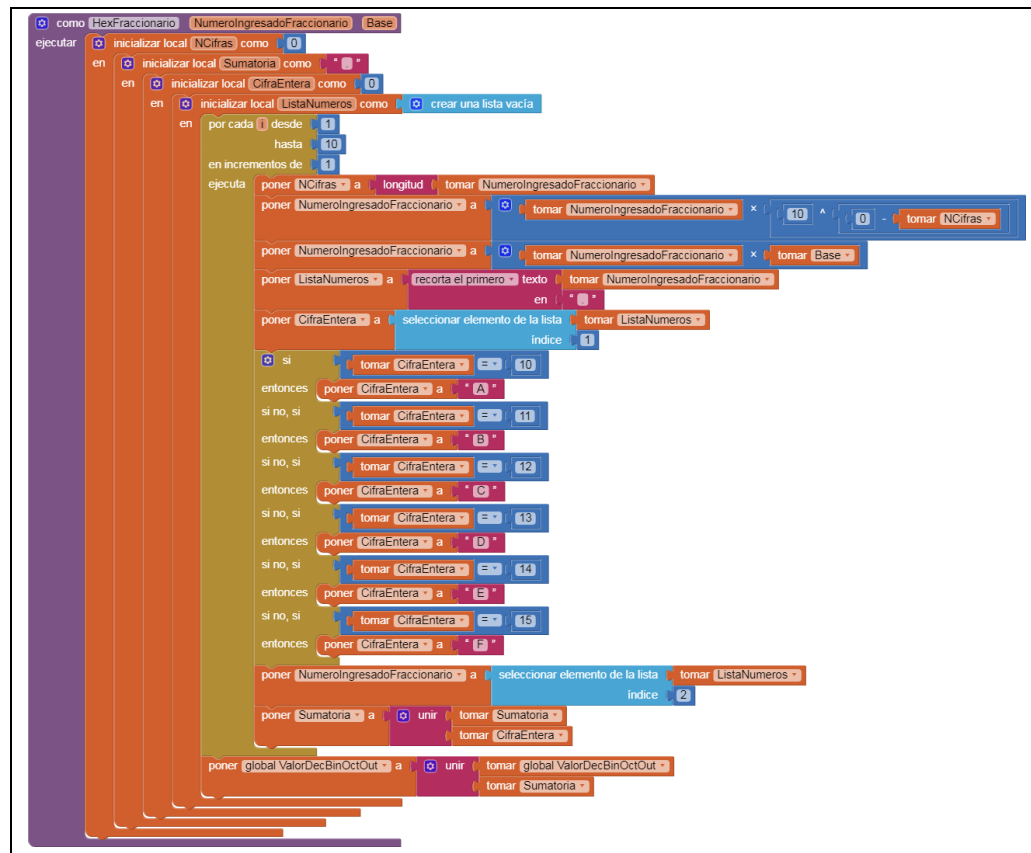
Se enfoca en realizar las conversiones a los dígitos hexadecimales por medio de divisiones sucesivas a 16. La diferencia respecto a la anterior función es que cada residuo generado se agrupa como

cadena de caracteres, y si los residuos resultan ser números entre 10 a 15 se procede a transformarlos a sus letras correspondientes.



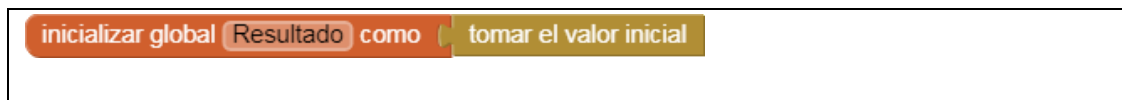
8.2.5.2.4 Función HexFraccionario

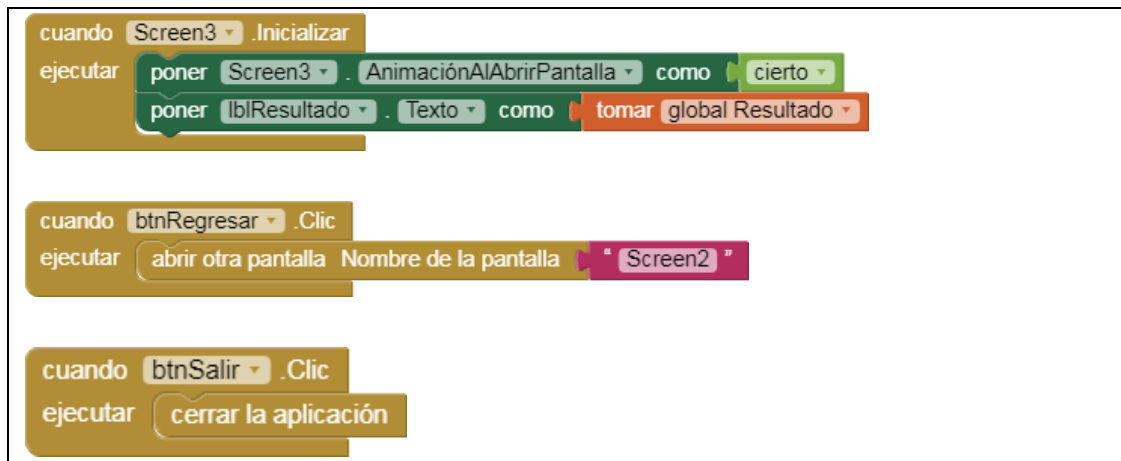
Su proceso es similar a lo que desarrolla la función HexEntero, solo con multiplicaciones sucesivas de la parte fraccionaria por su base hasta que el mismo sea 0. Sus partes enteras reunidas se agregan como caracteres, de izquierda a derecha.



8.3 PANTALLA DE RESULTADO

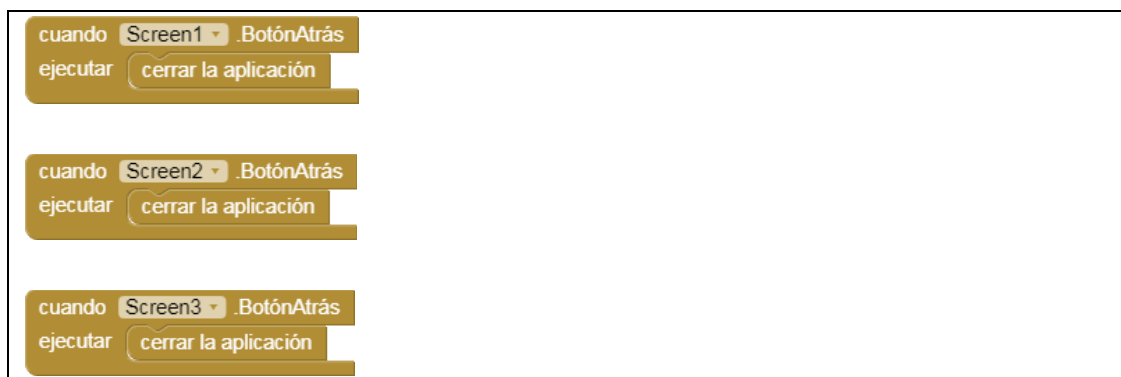
Se encarga de mostrar el resultado de la conversión. El dato transferido se muestra en una etiqueta de texto. Además, define los métodos para los dos botones existentes: Regresar y Salir; donde el primero permite realizar mas conversiones a otros números y otro para salir de la aplicación.





8.4 BOTON ATRÁS

Este objeto está activado en cada pantalla ya que se encargará de que se pueda salir de la aplicación, evitando la generación de pestañas que son copias de los resultados.



9. DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Los elementos necesarios para el proceso de diseño, codificación y pruebas son:

Ordenador y el sistema operativo

- Macintosh (con procesador Intel): Mac OS X 10.5, 10.6
- De Windows: Windows XP, Windows Vista, Windows 7

- GNU / Linux: Ubuntu 8 +, Debian 5 +

Navegador

- Mozilla Firefox 3.6 o superior
- Apple Safari 5.0 o superior
- Google Chrome 4.0 o alta
- App Inventor no soporta Microsoft Internet Explorer. Los usuarios de Windows deben utilizar Chrome o Firefox

Teléfono

Las aplicaciones creadas con App Inventor pueden funcionar en cualquier teléfono Android, siempre que cuenten instalados con una tarjeta SD. El entorno de desarrollo y el propio software de configuración soportan directamente los siguientes teléfonos:

- Google: Nexus S, Nexus One, ADP1, ADP2, Ion
- T-Mobile: G1 *, myTouch 3G *
- Motorola: Droid *
- hardware similar en otras compañías

10.CONCLUSIONES

- Se puede concluir que el desarrollo de la aplicación Conversión de Sistemas de Numeración resultó un proceso eficaz al momento de la transformación de sistemas en la plataforma App Inventor.
- Se concluye que el sistema decimal posee un método de conversión global ya que su forma de conversión puede ser aplicado a cualquier otro sistema.

- Se determinó que no es necesario el conocimiento de la programación avanzada para el desarrollo de aplicaciones en Android a través de App Inventor.
- La plataforma App Inventor que permite desarrollar aplicaciones sencillas pero funcionales sin la necesidad un gran programador.

11.RECOMENDACIONES

- Se recomienda utilizar la aplicación Conversión de Sistemas de Numeración cuando se requiera solucionar problemas entre sistemas de numeración.
- Para facilitar el proceso de conversión entre sistemas, se recomienda pasar al sistema decimal ya que pasar de este sistema a otro es más sencillo.
- Al momento de programar es recomendable realizar primero el diagrama de flujo y hacer la prueba de escritorio antes que utilizar el lenguaje de programación.
- Se recomienda consultar las distintas funcionalidades que ofrece App Inventor para un mejor desarrollo de las aplicaciones.

13.CRONOGRAMA

LUN.	MAR.	MIÉ.	JUE.	VIE.	SÁB.	DOM.
25	26	27	28	29	30 2 elementos Invitar al grupo de trabajo Introducción a App Inventor	31 de may. 1 elemento Realización del artículo I
1 de jun. 1 elemento Realización del artículo II	2 1 elemento Realización del artículo III	3 1 elemento Diseño externo de la app	4 1 elemento Realización del artículo IV	5 1 elemento Diseño interno de la app (Estructuras con diagramas de bloques)	6 1 elemento Diseño interno de la app (Estructuras con diagramas de bloques)	7 1 elemento Desarrollo del informe I



14. BIBLIOGRAFÍA

Qué es AppInventor? - Curso de introducción a la programación de apps con AppInventor. (n.d.). Retrieved June 14, 2020, from <https://www.programoergosum.com/cursos-online/appinventor/27-curso-de-programacion-con-app-inventor/primeros-pasos>

App inventor. Renombrar pantallas. Cambiar nombre de ventanas. Copiar pantallas ventanas. (n.d.). Retrieved June 14, 2020, from <http://kio4.com/appinventor/127renombrarpantallas.htm>

Programación Unidad 1: Introducción a la Programación y App Inventor - Technovation Girls. (n.d.). Retrieved June 14, 2020, from <https://technovationchallenge.org/curriculum/programacion-unidad-1-introduccion-a-la-programacion-y-app-inventor/?lang=es>

Sistemas binarios: sistemas de numeración. (n.d.). Retrieved June 14, 2020, from <http://platea.pntic.mec.es/~lgonzale/tic/binarios/numeracion.html>

Tema 1.2. Breve historia de Windows y otros sistemas operativos. (n.d.). Retrieved June 14, 2020, from http://platea.pntic.mec.es/vgonzale/pc_10/archivos/_124/Tema_1.2.htm

¿Qué es Android?. xatakandroid (n.d.). Retrieved June 14, 2020, from <https://www.xatakandroid.com/sistema-operativo/que-es-android>

Li, Y., Pan, Y., Liu, W., & Zhang, X. (2018). An automated evaluation system for app inventor apps. *Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and IEEE 3*, 236–242. <https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00048>

Moura Oliveira, P. B. (2015). *Teaching automation and control with App Inventor applications. 2015 IEEE Global Engineering Education Conference (EDUCON)*. doi:10.1109/educon.2015.7096077 url to share this paper:

[sci-hub.tw/10.1109/educon.2015.7096077](https://doi.org/10.1109/educon.2015.7096077)

Sağlam, M. (2019). No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title. *FLEPS 2019 - IEEE International Conference on Flexible and Printable Sensors and Systems, Proceedings*, 6(1), 1–46. <https://doi.org/10.1016/j.surfcoat.2019.125084>

15. ANEXOS

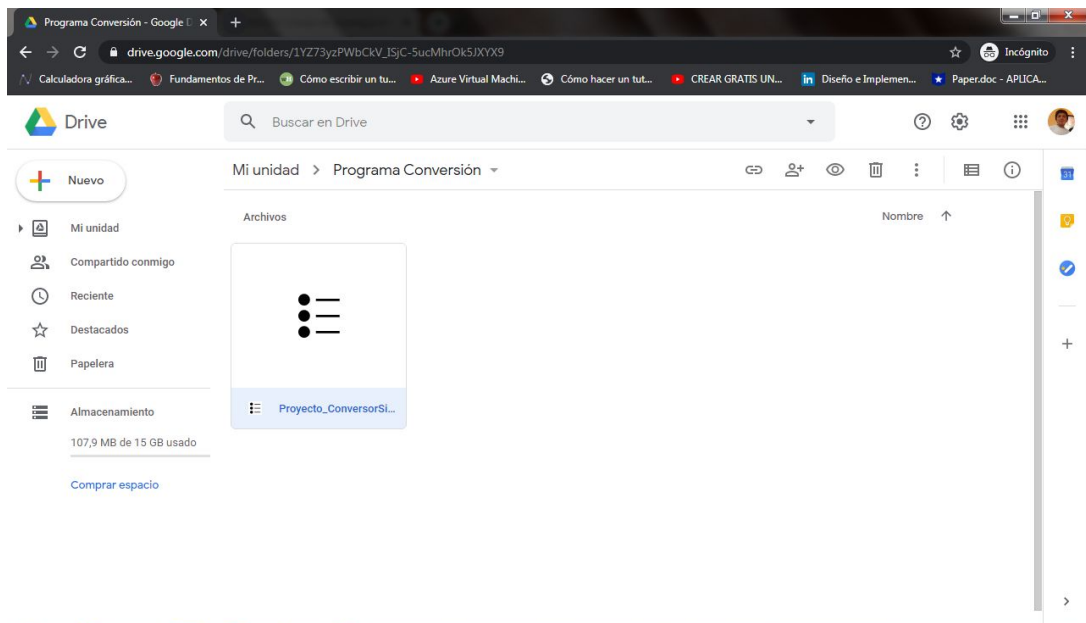
15.1 MANUAL DE USUARIO

15.1.1 Requerimientos

- 1) Conexión a Internet
- 2) 5 MB de memoria disponible
- 3) Teléfono con sistema operativo Android

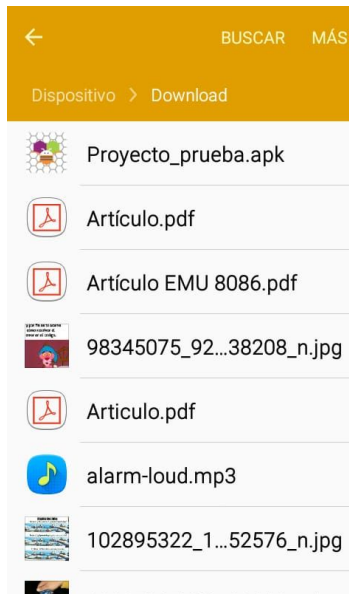
15.1.2 Instalación

Esta aplicación no se encuentra en Google Play, por lo que se especifica la dirección de descarga.



<https://drive.google.com/file/d/1IGOT-c8PnhEtHxWVObfgF8KmRSvx5LNv/view?usp=sharing>

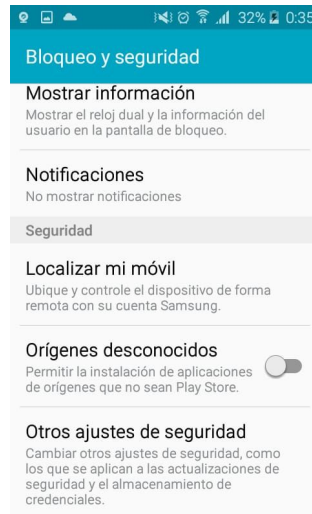
El link anterior permite acceder de forma directa al archivo de extensión APK que contiene la aplicación. Una vez descargado el archivo se procede a ejecutarlo.



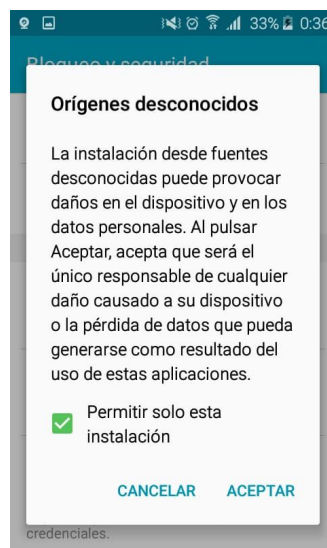
Por defecto el sistema Android no permite la instalación de aplicaciones que no sean descargadas a través de la Play Store, por lo tanto cuando se ejecute el programa saltará un mensaje que avisa al usuario que la instalación, no es posible.



En la parte inferior se encuentran las opciones 'Cancelar' y 'Ajustes'. Se selecciona la opción 'Ajustes', seguido se muestra el menú de opciones de 'Bloqueo y Seguridad'.

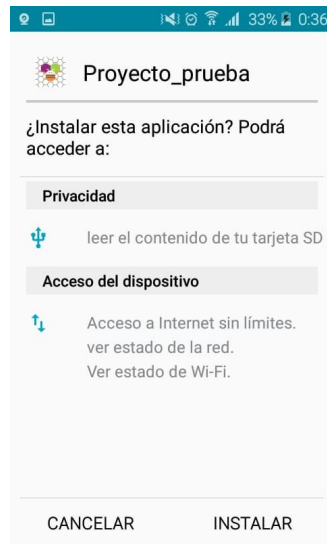


Se selecciona la opción 'Orígenes desconocidos', donde nos sale el siguiente mensaje:

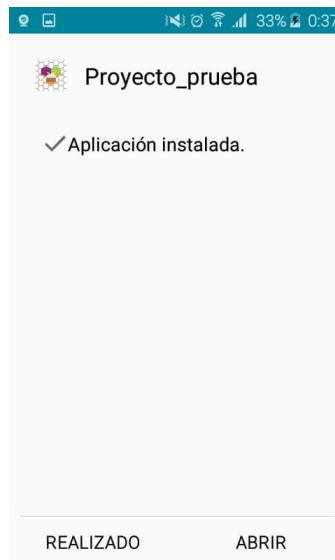


Se selecciona la opción 'Permitir sólo esta instalación' y se da click en el botón 'Aceptar'.

A continuación aparecerá el siguiente mensaje, donde se debe seleccionar la opción 'Instalar', para que dé comienzo a la instalación normal de la aplicación.



Una vez terminado todo el proceso de instalación aparecerá la siguiente pantalla donde se selecciona la opción 'Abrir', que nos conducirá a la pantalla de bienvenida de la aplicación.

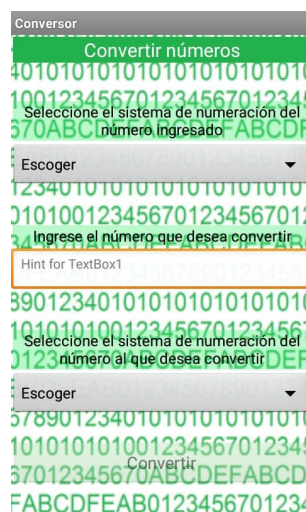


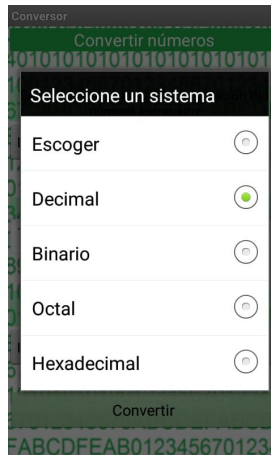
15.1.3 Ejecución del programa

Lo primero en aparecer es la pantalla de bienvenida donde se muestra el nombre de la aplicación y la versión que se está utilizando.

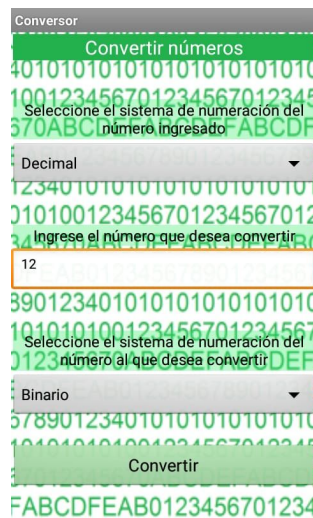


A continuación se muestra la pantalla de ingreso de datos, donde lo primero que va a pedir el programa es ingresar el sistema que desea ser transformado, por consiguiente se desplegará una pantalla con todas las opciones de los diferentes sistemas de numeración existentes.

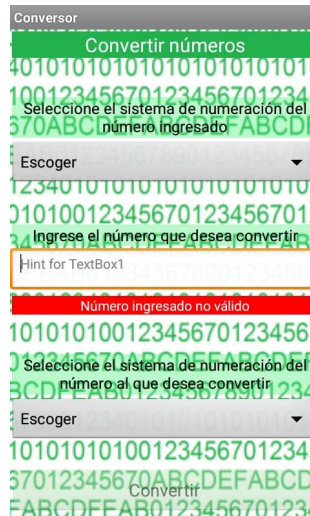




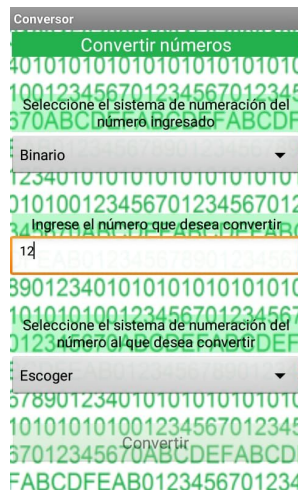
A continuación, se le pedirá al usuario ingresar el número que desea ser convertido, es decir como entrada le pedirá el valor que usted desea ingresar para ser transformado a otro sistema.



Si el numero ingresado no pertenece al sistema seleccionado se desplegará el siguiente mensaje indicando al usuario que el número es incorrecto, además de borrar el contenido de la caja de texto.



Cuando se ingresa el valor y que este es el correcto se le pedirá al usuario que elija el sistema de numeración a ser convertido su valor ingresado.



Si se selecciona el mismo sistema de numeración que el ingresado saltará el siguiente mensaje indicando al usuario que los sistemas son iguales.

The screenshot shows a mobile application titled 'Conversor' with a green header 'Convertir números'. Below the header, there is a prompt 'Seleccione el sistema de numeración del número ingresado' followed by a dropdown menu currently set to 'Octal'. Below this is a text input field with the number '45' entered, preceded by the prompt 'Ingrese el número que desea convertir'. Another dropdown menu, also set to 'Octal', is followed by the prompt 'Seleccione el sistema de numeración del número al que desea convertir'. At the bottom is a green 'Convertir' button. A yellow status bar at the very bottom reads 'Los sistemas seleccionados son los mismos'.

Cuando se selecciona el sistema al que se desea convertir se oprime la tecla 'Convertir' y automáticamente tendremos el resultado de la conversión dando finalizado al programa.

The screenshot shows the 'Resultado' screen of the application. It features a green header 'Resultado'. Below it, a green bar contains the text 'Su resultado es:'. Underneath this bar, the number '45' is displayed in a white box. At the bottom of the screen are two green buttons labeled 'Regresar' and 'Salir'.