



# Creación de App para control de robot con Lego EV3

El mobil learning y la educación virtual ubicua.  
Pedro Fonseca Solano  
Irene Hernández Ruiz  
Universidad Nacional de Costa Rica  
Escuela de Informática

**Resumen:** *en la Escuela de Informática de la Universidad Nacional, se crea el Proyecto Formación de Formadores en Robótica para Colegios en áreas Vulnerables de Costa Rica. Este proyecto pretende capacitar a 72 docentes en el área de robótica educativa para que ellos repliquen su experiencia en sus instituciones. En el marco de este proyecto se construyó una app para el uso del robot Lego EV3, de esta manera los docentes pueden hacer uso de esta app para sus talleres con los estudiantes. En estos días la creación de una app, es una nueva forma para representar el conocimiento y la información según sea nuestra necesidad, cada día más personas necesitan tener la oportunidad de crearlas, y es en este sentido se escribió este tutorial para mostrar cómo se crea una app básica para controlar un robot vía bluetooth.*

**Palabras clave:** robot Lego EV3, APP Inventor, enseñanza, aprendizaje.

## 1. Introducción

La Escuela de Informática, ha incursionado en el área de la robótica con un curso optativo en su plan de estudios, así como en el proyecto de Proyecto Formación de Formadores en Robótica para Colegios en áreas Vulnerables de Costa Rica. Para lo cual se ha trabajado con las tecnologías de Lego NXT y Lego EV3, en cada una de las lecciones desarrolladas se han creado modelos para la construcción de modelos de robots para un uso fácil y que permita desde una metodología de aprender haciendo conocer sobre la programación en un ambiente



gráfico. Actualmente, los países en vías de desarrollo están haciendo uso de la robótica para desarrollar habilidades creativas en los jóvenes (Blank 2006)

Por otra parte, los teléfonos móviles cuentan con sistemas operativos más robustos y con más capacidad para colocar app. De esta forma el acceso constante para nuestro uso cotidiano se vuelve más frecuente.

El acceso a la tecnología debe presentarse como una oportunidad para que más ciudadanos puedan aprender sobre el tema y puedan diseñar nuevas soluciones creativas.

Por ser ambos temas la robótica y el desarrollo de app unos de los mayores auges en estas tecnologías, se desarrolló un app denominada Formadores.apk que permite mostrar cómo se crea una app, básica para controlar un robot vía bluetooth.

Un aspecto fundamental es elegir la plataforma de desarrollo, en este caso se decidió utilizar el APP INVENTOR porque es una herramienta con un ambiente gráfico muy sencillo de utilizar y sus instrucciones son muy claras.

## **2. Historia del APP INVENTOR**

En principio fue desarrollado por el profesor Harold “Hal” Abelson profesor de Ingeniería Eléctrica y Ciencias de la Computación en el Instituto Tecnológico de Massachusetts MIT, y un equipo de Google Education. Cuando pasaba un año de licencia en Google, App Inventor se ejecuta, por medio de un servicio Web administrado por personal del Centro para el aprendizaje móvil del MIT. El App tenía en el 2015 con una comunidad mundial de casi dos millones de usuarios, quienes representaban a 195 países en todo el mundo. Cerca de 85 mil usuarios semanales activos de la plataforma han creado cerca de 4,7 millones de aplicaciones de Android. Se trata de una herramienta de código abierto que permite la programación y la creación de apps para una amplia gama de audiencias sin necesidad de ser un programador.

App Inventor es una herramienta basada en la nube, lo que significa que se puede construir aplicaciones directamente en su navegador web. Este sitio web ofrece



todo el apoyo que necesita para aprender a construir sus propias aplicaciones. El software de App Inventor, o "servicio" es en [ai2.appinventor.mit.edu](http://ai2.appinventor.mit.edu) 1. (Wolber 2011).

Esta plataforma proporciona una serie de componentes que nos permiten establecer comunicación vía bluetooth, para parrear dispositivos como un celular y un robot Lego y componentes LEGO MINDSTORMS, con dos interfaces de alto y bajo nivel con funciones que nos permite el control de motores y leer valores de sensores.

### **3. Descripción del APP**

El nombre de la Aplicación es Formadores.apk, y tiene como propósito el control del Robot EV3 por medio de 2 motores (tanque) y un motor mediano.

Esta aplicación utilizará un botón para conectarse al robot EV3 por medio de bluetooth y otro para desconectarse. Además tendrá deslizador para aumentar o disminuir la potencia de los motores, y cinco botones para controlar el movimiento de dos motores que le permiten al robot desplazarse hacia adelante, atrás, derecha e izquierda, un botón como freno y finalmente una flecha hacia arriba para abrir o una flecha hacia abajo para cerrar una garra conectada a un tercer motor.

### **4. Recursos necesarios crear toda la aplicación**

Kit de robótica Lego MINDSTORMS EV3 45554, Celular Inteligente Android, Computador con acceso a Internet, una cuenta en google para acceder a la plataforma.

### **5. Diseño de la aplicación**

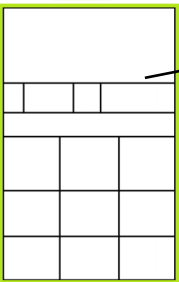
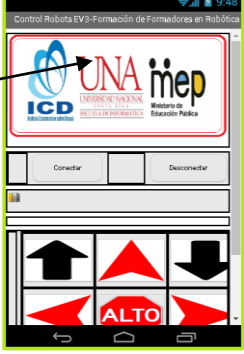



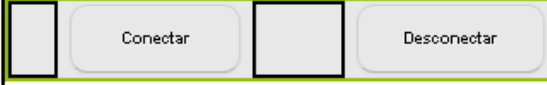


---

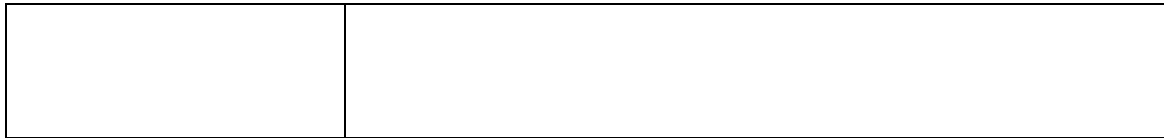
1

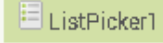


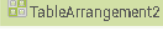





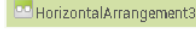

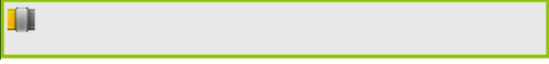










Tomado del sitio <http://appinventor.mit.edu/explore/get-started.html>

El diseño, lo debemos considerar como una plantilla, en donde colocaremos nuestros componentes, y bueno puede ser simple como esta o muy compleja dependiendo de que se quiera realizar, pero hay que familiarizarse con la Disposición (Layout) y los elementos para un diseño como Ud. desee hacerlo (DisposiciónHorizontal, HorizontalScrollArrangement, DisposiciónTabular, DisposiciónVertical, VerticalScrollArrangement), utilizaremos una ventana para mostrar nuestra aplicación y otra para un mensaje final, la plantilla sería el siguiente:

Figura 1: Plantilla para la construcción

<p>La disposición de nuestra plantilla está dentro de una ventana (Screen1), nombre que por defecto se le asigna en la plataforma, véanlo en contorno de color verde.</p> 	
<p>HorizontalArrangement2</p>  <p>La usamos para poner el rótulo de la aplicación como una imagen, aunque puede ser un texto</p>	
<p>HorizontalScrollArrangement1</p>  <p>Este contenedor ó arreglo nos permite insertar 4 elementos, 2 espacios y 2 botones.</p>	
<p>TableArrangement3</p> 	 <p>Primer elemento del vector 1 espacio.</p>



 		<p>Segundo elemento, es un botón ListPicker, selector de lista (muestra lista de dispositivos cercanos con bluetooth activado)</p>
 		<p>Tercer elemento, un espacio</p>
 		<p>4 elemento, botón desconectar, nombrado "fin"</p>
 	 Este arreglo solo contiene un elemento, que es el deslizador (Slider1) 	
  <p>Aquí vamos a crear una matriz o cuadrícula insertar los diferentes componentes que utiliza nuestra aplicación.</p>		
  	<p>Este es el primer elemento de la matriz, la utilizaremos para insertar la flecha hacia arriba para enviar un pulso hacia adelante al motor A.</p>	
  	<p>Este es el 1er elemento de la matriz, es un botón con la imagen respectiva</p>	

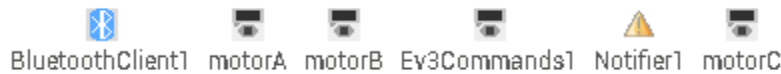
Fuente: Elaboración propia

### 6. Componentes no visibles

Para realizar nuestra aplicación utilizaremos además componentes de Conectividad para el caso de la conexión vía bluetooth, LEGO MINDSTORMS, y el notificador de Interfaz de usuario, algunos de estos componentes se les denomina





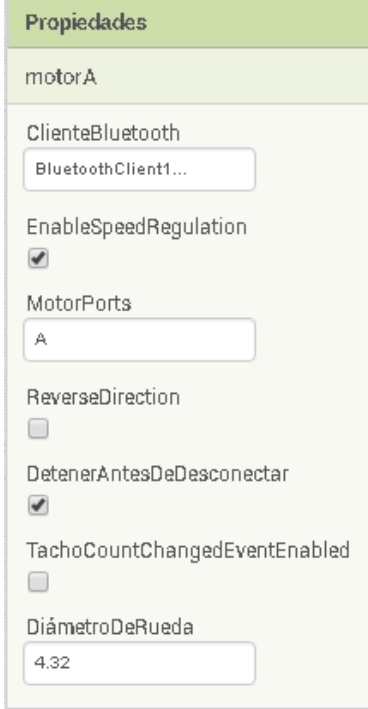
no visibles, y se ubican debajo de nuestra pantalla de diseño con su nombre respectivo, al seleccionar cualquier componente, de inmediato en el cuadro de propiedades nos muestra la información de dicho componente.


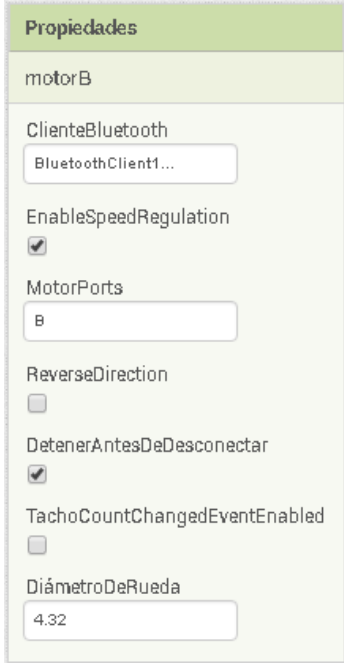
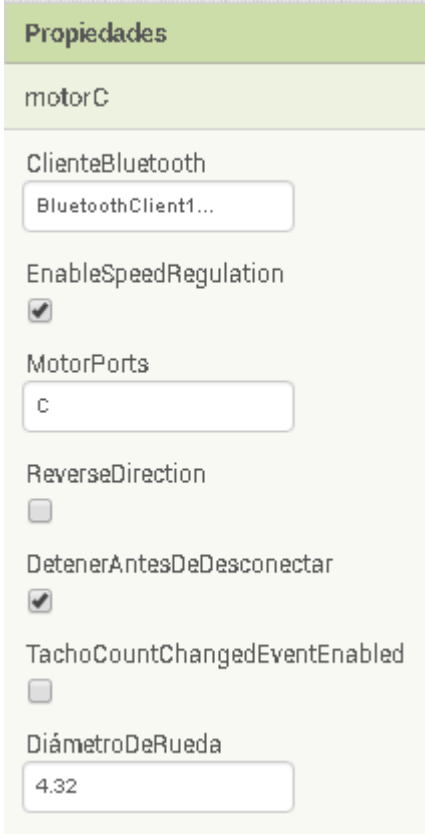
Figura 2: Componentes no visibles





Fuente: Elaboración propia

Figura 3: Descripción de los Componentes no visibles

 ClienteBluetooth		<p>Componente de cliente Bluetooth, para establecer comunicación con Robots EV3 o NXT. Por ejemplo con la propiedad AddressesAndNames, se obtienen las direcciones y los nombres de los dispositivos Bluetooth vinculados</p>
 Ev3Motors  motorA <p>Cada vez que incluya un componente EV3Motors, los debe arrastrar en la screen1, para que sea incluido como un elemento más de su aplicación. Le puede cambiar el nombre como se estila A, B, C, D.</p>		<p>Un componente que proporciona dos interfaces de alto y de bajo nivel para robots LEGO MINDSTORMS EV3, con funciones que pueden controlar los motores.</p> <p>Es muy importante que cada vez que su utiliza un motor, se especifique el cliente Bluetooth, ya que aunque tenga el componente nombrado, si no indica que la conexión es por esta vía, nunca podrá realizar conexión alguna y por la tanto la aplicación no servirá.</p> <p>Y si usa 4 motores debe hacerlo para cada uno, excepción los incluya todos en solo, pero será como si fuese 1, pues todos tendrán las misma propiedades.</p>

 <p>                     RotateInDuration                      (Fuente de número, número milisegundos, useBrake booleano)                      Girar los motores en un período de tiempo.                      RotateInDistance                      (Fuente de número, número de distancia, useBrake booleano)                      Girar los motores en una distancia.                      RotateSynclnDefinitely                      (Fuente de número,                 </p>	 <p> <b>Propiedades</b>                      motorB                      ClienteBluetooth                      BluetoothClient1...                      EnableSpeedRegulation <input checked="" type="checkbox"/>                      MotorPorts                      B                      ReverseDirection <input type="checkbox"/>                      DetenerAntesDeDesconectar <input checked="" type="checkbox"/>                      TachoCountChangedEventEnabled <input type="checkbox"/>                      DiámetroDeRueda                      4.32                 </p>	<p>                     Propiedades de los motores                      BluetoothClient                      El componente BluetoothClient que se debe utilizar para la comunicación. Se debe establecer en el Diseñador                      MotorPorts                      Los puertos del motor que los motores están conectados. Los puertos se especifican mediante una secuencia de letras del puerto. Se debe establecer en el Diseñador                      WheelDiameter                      El diámetro de las ruedas unidas en los motores en centímetros.                      ReverseDirection                      Se especifica si la dirección de los motores se invierte.                 </p>
<p>                     RotateSynclnDuration                      (Fuente de número, número de milisegundos, el número turnRatio, useBrake booleano)                      Girar los motores a la misma velocidad en un periodo de tiempo.                      RotateSynclnTachoCounts                      (Fuente de número, número tachoCounts, número turnRatio, useBrake booleano)                      Girar los motores a la misma velocidad en un número de recuentos de taco.                      Stop (UseBrake booleano)                      Detener los motores del robot.                 </p>	 <p> <b>Propiedades</b>                      motorC                      ClienteBluetooth                      BluetoothClient1...                      EnableSpeedRegulation <input checked="" type="checkbox"/>                      MotorPorts                      C                      ReverseDirection <input type="checkbox"/>                      DetenerAntesDeDesconectar <input checked="" type="checkbox"/>                      TachoCountChangedEventEnabled <input type="checkbox"/>                      DiámetroDeRueda                      4.32                 </p>	<p>                     EnableSpeedRegulation                      El robot ajusta la potencia para mantener la velocidad si está habilitada la regulación de la velocidad.                      StopBeforeDisconnect                      Ya sea para detener el motor antes de desconectar.                      TachoCountChangedEventEnabled                      Si el evento TachoCountChanged debe disparar cuando se cambia el ángulo.                 </p>

 Ev3Commands	<p><b>Propiedades</b></p> <p>Ev3Commands1</p> <p>ClienteBluetooth</p> <p>BluetoothClient1...</p>	<p>Un componente que proporciona una interfaz de bajo nivel a un robot LEGO MINDSTORMS EV3, con funciones para enviar comandos directos del sistema o de los robots EV3.</p>
 Notifier1	<p><b>Propiedades</b></p> <p>Notifier1</p> <p>ColorDeFondo</p> <p><input checked="" type="checkbox"/> Gris oscuro</p> <p>DuraciónDelNotifier</p> <p>Largo ▾</p> <p>ColorDeTexto</p> <p><input type="checkbox"/> Blanco</p>	<p>El componente Notificador muestra cuadros con alertas, mensajes y alertas temporales, y hace anotaciones en el registro de Android por medio de métodos.</p>

Fuente: Elaboración propia

## 6. Programación de Bloques

Una de las formas sencillas para aprender a programar, es la de bloques, que permite una mejor comprensión de los elementos propios de la programación Kamriani (2016), para ello se decidió utilizar el ambiente de desarrollo Scratch.

A continuación, se presenta una imagen del app desarrollado:

Figura 4: App desarrollado



Fuente: Elaboración propia

Los pasos necesarios para la creación del APP son:

Inicialización de variable Poder, que se utilizará para la potencia de los 3 motores

```
inicializar global poder como 75 × 2
```

Una vez inicializada la aplicación se cargan las variables y los procedimientos que no son generados por un botón.

**Conectar**

Al hacer clic a este botón, se ejecutará el bloque:

```
cuando ListPicker1 .AntesDeSelección
ejecutar poner ListPicker1 .Elementos como BluetoothClient1 .DireccionesYNombres
```

Este bloque hará que en una lista se guarden las direcciones y nombres de los dispositivos bluetooth activos.

Nótese que dice .AntesDeSelección, pues al hacer clic, aparecerá en la pantalla la listas los dispositivos activos

Una vez seleccionado un dispositivo de la lista, o sea .DespuésDeSelección, se envía la señal desde el celular al robot, con los datos de la dirección y el nombre, en caso contrario, entonces la lista no mostrará nada (falso) y activa el botón llamado "fin" como cierto, y si no al botón "fin" le pone como nombre "Desconectado".

```
cuando ListPicker1 .DespuésDeSelección
ejecutar si
    llamar BluetoothClient1 .Conectar dirección ListPicker1 .Selección
entonces poner ListPicker1 .Visible como falso
    poner fin .Visible como cierto
si no poner fin .Texto como "Desconectado"
```

El deslizador (slider1) lo usamos para trasladar su valor en una variable que usaran los motores.

```
when Slider1 .PositionChanged
thumbPosition
do set global poder to round get thumbPosition
```

Cada vez que movamos el deslizador se actualiza la variable obtenida del get a thumbPosition, y se obtiene el valor, se redondea y se asigna a la variable llamada "poder"



Botón para abrir garra, manda una señal al motor A (btnAbrir)

```

when btnAbrir .Click
do
  call motorA .RotateInDuration
    power get global poder
    milliseconds 100
    useBrake true
  
```

Cada vez que se hace clic al botón (btnAbrir) se envía un pulso de 100 milisegundos al motor A y se frena el motor A



Este botón lo utilizamos para que los motores B y C, se enciendan btnAdelante,

```

when btnAdelante .Click
do
  call motorB .RotateInDuration
    power Slider1 .ThumbPosition
    milliseconds 100
    useBrake false
  call motorC .RotateInDuration
    power Slider1 .ThumbPosition
    milliseconds 100
    useBrake false
  
```

Cuando se hace clic al (btnAdelante) se envía una señal a los motores B y C de 100 milisegundos, sin freno con el valor del slider o igual de la variable poder

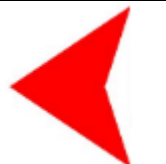


Botón para cerrar garra, manda una señal al motor A (btnCerrar)

```

when btnCerrar .Click
do
  call motorA .RotateInDuration
    power get global poder * -1
    milliseconds 100
    useBrake true
  
```

Cuando se hace clic al botón (btnCerrar) se envía un pulso de -100 (reversa) milisegundos al motor A y se frena el motor A, si nuestro robot tiene una garra la cerrará.



Este botón (btnIzquierda) lo utilizamos para que los motores un giro hacia la izquierda, un motor gira hacia adelante y otro hacia atrás,

```

when btnIzquierda .TouchDown
do
  call motorB .RotateInDuration
    power get global poder
    milliseconds 100
    useBrake true
  call motorC .RotateInDuration
    power get global poder * -1
    milliseconds 100
    useBrake true
  
```

Cuando presionamos el este botón se hará un giro hacia la izquierda, también depende como usted alambrió los motores B y C.



Este botón, hace que los motores se detengan, btnAlto, cuando se hace clic a este botón, observe la instrucción llama a los motores los detiene y frena.

```
when btnAlto .Click
do
  call motorA .Stop
  useBrake true
  call motorB .Stop
  useBrake true
  call motorC .Stop
  useBrake true
```



Este botón lo utilizamos para hacer un giro hacia la derecha, entonces encendemos y el motor C en modo invertido (hacia atrás) y el motor B normal, hacia adelante, btnDerecha.

```
when btnDerecha .TouchDown
do
  call motorB .RotateInDuration
  power get global poder * -1
  milliseconds 100
  useBrake true
  call motorC .RotateInDuration
  power get global poder
  milliseconds 100
  useBrake true
```

```
when fin .Click
do
  call motorA .Stop
  useBrake true
  call motorB .Stop
  useBrake true
  call motorC .Stop
  useBrake true
  call BluetoothClient1 .Disconnect
  open another screen screenName "Acerca"
```

Aquí al hacer clic al botón "fin" con título Desconectar, se apagan todos los motores, se desconecta el bluetooth y abre una nueva ventana llamada "Acerca", que muestra una imagen y dos botones, uno para salir y otro para regresar a la aplicación



Esta es la ventana "Acerca", para salir o regresar de la aplicación. Si decide salir le envía un mensaje "hablado" dando las gracias por usar la aplicación

```
when salir .Click
do
  call TextToSpeech1 .Speak
  message "La Universidad Nacional le da las gracias por usar esta aplicación"
  close application
```

```
when Acerca .ErrorOccurred
component functionName errorNumber message
do
  set message to "debe activar el Bluetooth"
```

Si trata de conectarse de nuevo, debe activar el bluetooth, aquí se le envía un mensaje cuando ocurre el error de que el buetooth no está activado.

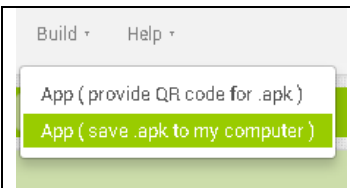
```
when regresar .Click
do
  open another screen screenName "Screen1"
```

Regreso a la aplicación, se vuelve a llamar a la pantalla Screen1, para volver al inicio

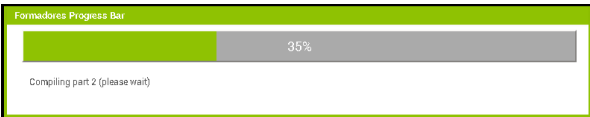
de la aplicación. Recuerde que cada pantalla tiene asociada un área independiente para los bloques de programación. Desde la pantalla “Acerca” no podrá ver los bloques de la pantalla principal “Screen1”.

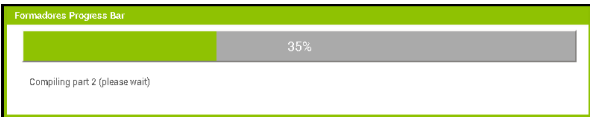
## 6. Generando e instalando la aplicación

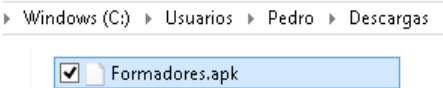
Para crear la aplicación debe generarla y guardarla en su computador y luego pasarla al celular o lo puede hacer directamente si tiene conectado su celular en su computador.

 A screenshot of a software interface showing a 'Build' menu. The menu is open, displaying two options: 'App ( provide QR code for .apk )' and 'App ( save .apk to my computer )'. The second option is highlighted in green.	Debe ir a la opción Build y seleccionar App (save .apk to my computer)
--	--

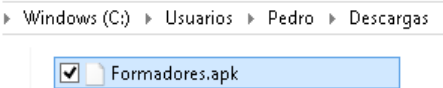
Esto hará que le aparezca una ventana de progreso sobre la compilación de la aplicación

A screenshot of a progress bar titled 'Formadores Progress Bar'. The bar is partially filled with green, indicating 35% completion. Below the bar, it says 'Compiling part 2 (please wait)'.


--

A screenshot of a Windows file explorer window. The path is 'Windows (C:) > Usuarios > Pedro > Descargas'. A file named 'Formadores.apk' is selected and checked.



Esta aplicación será guarda en su disco duro en la carpeta descargas, lo que procede es copiarla en su celular e instalarla.

 <p>Esta aplicación será guarda en su disco duro en la carpeta descargas, lo que procede es copiarla en su celular e instalarla.</p>
--

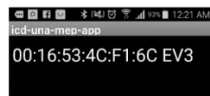
Una vez cargada en tu celular la aplicación, debes instalarla, ubica la aplicación en el icono Mis Archivos , haz un clic ve a la opción archivos recientes, y a parecerá la siguiente imagen.

A screenshot of a mobile file manager interface. The title bar says 'Mis archivos'. Under the 'Archivos recientes' section, a file named 'Formadores.apk' is listed with a small icon next to it.


---

	<p>Haz un clic para instalarla y te aparece una pantalla de seguridad, donde debes activar la opción Orígenes desconocidos.</p> <p>Te enviara un cuadro con un mensaje de advertencias de que las aplicaciones de orígenes desconocidos....</p> <p>Le das aceptar, en mi caso como ya tenía la aplicación instalada yo use la opción de TODO</p>
	<p>Finalmente aparecerá la pantalla de la izquierda y le das la opción abrir y te aparecerá la aplicación. A este nivel debes tener un robot EV3 con tres motores B y C, para ir hacia adelante y motor A, para una garra o “kick” patada. Recuerda parrear los dispositivos con el bluetooth del celular al robot y ya podás empezar a controlar los motores.</p>

Al hacer clic en la opción conectar, siempre y cuando hayas “pariado”<sup>2</sup> el cel con el robot, te aparecerá una venta como esta, seleccionas la dirección y el nombre y listo, ya puedes iniciar el control.



## 7. Conclusiones

Al menos 20 estudiantes han utilizado esta app para utilizar sus modelos de robots, quienes han podido “mover” el robot haciendo uso de esta app, los estudiantes han manifestado que su uso es sencillo e innovador.

Los estudiantes se motivan mucho más al tener en su teléfono móvil un app para que les permite utilizar el modelo aprendido.

<sup>2</sup> La conexión vía bluetooth exige una contraseña, para efectos prácticos use 1234F

Se le ha brindado una capacitación tanto en el entorno físico como el virtual donde el estudiante no requiere de un robot físico para poder programar. El uso de estas nuevas tecnologías en los estudiantes permiten que ellos se adentren en este mundo nuevo de desarrollo tecnológico y disminuir la brecha en el acceso a las tecnologías, motivando a que los estudiantes estudien carreras orientadas a las áreas TIC.

## 7. Bibliografía

- Blank, D, Robots Make Computer Science Personal, Communications of the ACM,49(12):25-27, 2006 <http://myro.roboteducation.org/~dblank/cacm-12-2006.pdf>
- David Wolber, Hal Abelson, Ellen Spertus, Liz Looney .App Inventor, O´Reilly, Apr 27, 2011
- Scratch Programming in easy steps: Covers versions 2.0 and 1.4, By Sean McManus, Oct 3, 2013
- App Inventor 2: Create Your Own Android Apps, By David Wolber, Hal Abelson, Ellen Spertus, Liz Looney, O'Reilly Media, Inc., Oct 13, 2014
- Felicia Kamriani, Krishnendu Roy, A step-by-step introductory guide to mobile app development with App Inventor 2, Packt Publishing Ltd, Apr 14, 2016



### **Currículum Pedro Fonseca Solano**



Licenciatura en informática con énfasis en sistemas de información, Universidad Nacional, Costa Rica, Bachillerato: Enseñanza de la de Computación, Universidad Nacional, Costa Rica, docente e investigador de la Escuela de Informática de la Universidad Nacional de Costa Rica.

### **Currículum Irene Hernández Ruiz**



Maestría en Administración de Tecnología de la Información, Universidad Nacional, Costa Rica, Bachillerato: Ingeniería en Sistemas de Información, Universidad Nacional, Costa Rica, docente e investigadora de la Escuela de Informática de la Universidad Nacional de Costa Rica.