

CONCEPTOS DE PROGRAMACIÓN

Índice

Índice	0
Que es Programar?	1
App Inventor	1
Programación en App Inventor	2
Organización del espacio de trabajo en App Inventor	3
Elementos en App Inventor	4
Interfaz de usuario	5
Medios	6
Sensores	6
Sociales	7
Conectar App Inventor al celular	7
Wifi	7
Datos	8
Variables	8
Listas	8
Pasar de pantalla	9
Cómo crear y utilizar colores en app inventor	9
Condicionales	10
Subir el proyecto a la galería	12
Compartir la aplicación	12
Acciones sincrónicas y asincrónicas	14
Almacenamiento	15

Que es Programar?

Programar es **encontrar soluciones a problemas** y **automatizarlas** . Cada dispositivo, cada página web y cada aplicación para celular que usamos es una solución a un problema que alguien más decidió pensar. Hay muchos problemas que hoy no están siendo solucionados (simplemente porque hasta ahora nadie lo identificó o se propuso solucionarlo). Por eso queremos acercarles a las chicas la posibilidad de elegir qué problemas son importantes para ellas y de proponer una solución que ayude a resolverlo. Recordemos también que, difícilmente, una app de manera aislada pueda resolver por completo el problema. Hay diferentes actores (gobierno, instituciones, sociedad, etc) que necesitan involucrarse también para poder solucionarlo. Así mismo, la tecnología es una gran aliada para ayudar a que ese problema se pueda resolver. La programación es una de las (pocas) disciplinas que nos permite llevar nuestras ideas a herramientas disponibles para millones de personas en muy poco tiempo, y con muy pocos recursos: sólo se necesita una computadora y aprender a programar.

A través de la programación podemos indicarles a diferentes dispositivos qué es lo que tienen que hacer. Para decirle a los dispositivos qué hacer, se utilizan **palabras** . Esas palabras son parte de un **lenguaje** que tanto las personas como el dispositivo entendemos. Programar es casi un super poder! Es hablar un lenguaje que poca gente habla y poder darle instrucciones a las máquinas. Recordá que cuando decimos máquinas, generalmente pensamos en una computadora, pero podemos hablar y darle instrucciones a cualquier otra máquina: un lavarropas, un televisor, un celular, y muchas más.

Cada una de estas máquinas entiende su propio lenguaje. Como el chino, el inglés y el español son idiomas para hablar con otros humanos, cuando construimos máquinas creamos también lenguajes para poder comunicarnos con ellas y entonces hacer que hagan lo que queremos. Hay muchos lenguajes de programación y muchas formas de comunicarse con las máquinas. En este caso nos vamos a comunicar con el celular usando el idioma de los **bloques** , a través de una herramienta llamada App Inventor.

Pueden leer [el artículo](#) que escribió Caro Hadad, una de las co-fundadoras de Chicas en Tecnología, para el diario La Nación.

App Inventor

[App Inventor](#) es una herramienta creada por Google y llevado adelante por el Massachusetts Institute of Technology (MIT), que nos permite usar el lenguaje de los bloques. Sirve para desarrollar aplicaciones para celulares Android y comparte cierta lógica con otros lenguajes por bloques como [Blockly](#), [PilasBloques](#) y [Scratch Jr](#).

Cuando hablamos de **lenguaje de programación** nos referimos al modo de comunicarnos con las máquinas (sean estas computadoras o cualquier otra).

Los lenguajes (como el castellano) tienen reglas y símbolos que deben conocer los dos interlocutores para codificar y decodificar la información. Lo mismo sucede con los lenguajes de programación: Existen distintos lenguajes que se utilizan para comunicarse con distintos dispositivos, y que han sido construidos para cumplir determinados objetivos.

App Inventor es un **lenguaje de programación por bloques**. Eso significa que para programar se utilizan bloques que pueden ser encastrados como si fuera un rompecabezas. Los lenguajes de programación en bloques son muy utilizados para aprender a programar, ya que permiten desarrollar los programas mediante la lógica y evitar los errores de sintaxis.

App Inventor es libre y gratuito. Una característica muy importante es que App Inventor es código abierto. Que una tecnología sea de código abierto significa que el código está disponible para que cualquier persona pueda verlo y, en muchos casos, colaborar en su desarrollo. En este caso colaboraron empresas e instituciones muy importantes como Google o el MIT (Massachusetts Institute of Technology), así como gente como cualquiera de nosotros que quiere aportar su granito de arena para hacer crecer la plataforma y que cada vez sea mejor. La tecnología de código abierto se puede analizar, entender cómo está construida, modificar, mejorar y compartir.

App Inventor es una herramienta online. Esto significa que vamos a acceder a ella a través de un navegador o browser (como lo son Chrome, Mozilla o Safari). Las aplicaciones que vayamos a construir se van a guardar en la cuenta de App Inventor que se cree. Para crear la cuenta vas a necesitar un usuario de gmail. Todas las aplicaciones que generemos también serán de código abierto. Esto significa que podríamos compartir con otros no solo la solución sino el código y cualquier persona puede colaborar y aportar a nuestros proyectos (y nosotros a los de cualquier persona que haya creado una app!)

Programación en App Inventor

Para conocer cualquier lenguaje, es necesario conocer **el código** (palabras) y **las reglas** para que las oraciones tengan sentido. En el caso de app inventor, existen distintos tipos de palabras, así como también tenemos sustantivos, verbos, pronombres, etc. En el caso de los seres humanos, podemos entender lo que nos dicen aunque no esté bien escrito o bien construidas las oraciones. Sin embargo, cuando hablamos a la máquina, las reglas son mucho más estrictas.

Para que un programa tenga sentido no pueden faltar:

- **Elementos** . Ejemplos de tipos de **elementos** son botones, textos, cámaras de fotos, etc. Es decir, todo aquello con lo que vamos a interactuar.
- **Propiedades y acciones** . Cada tipo de elemento tiene distintas **propiedades** . Las propiedades pueden cambiarse a través de **acciones** . Por ejemplo: el elemento botón, tiene como propiedades el ancho, el alto, el

texto y el color. Una acción que puede modificar la propiedad “texto” puede ser que el usuario haga click allí. Esa acción lo que cambiara es el **valor** de esta propiedad para el elemento.

- **Valores** . Los **valores** son ajustables, podrían ser modificados por otros valores, sin modificar el programa. En el caso del botón que nombramos antes, la propiedad ancho puede tener distintos valores (que lo hagan más o menos ancho).
- **Eventos** . Los **eventos** son acciones externas al programa que ocurren sin que lxs programadorxs puedan controlarlo (por ejemplo, el paso del tiempo, interacción de alguna persona).

Para programar, es necesario pasar de un lenguaje natural a un lenguaje de programación. Veamos un ejemplo! Intentemos escribir el programa que cocine un huevo. No te olvides de identificar elementos, acciones, propiedades y eventos.

Lenguaje Natural

Para cocinar un huevo, se lo sumerge en un recipiente con agua hirviendo durante unos diez minutos

Lenguaje de Programación

Cambiar Recipiente .**Contenido** a: **Agua**

Incrementar Recipiente .**Temperatura** a: **100 Grados**

Añadir a **Recipiente** .**Contenido** : **Huevo**

Cuando pase tiempo (**10 minutos**):

Quitar Huevo de **Recipiente**. **Contenido**

Importante: En App Inventor, los colores nos indican de qué tipo de bloque se trata.

Organización del espacio de trabajo en App Inventor

App Inventor tiene dos modos de trabajo: el modo **Diseñador** y el modo **Bloques** . La gran diferencia entre uno y el otro es que en el modo diseñador vamos a ver todo lo que tiene que ver con qué elementos tenemos en nuestra app (textos, cajitas de textos, imágenes, botones, sensores, etc), y en el modo bloques vamos a indicar cómo esos elementos se usan e interactúan con lxs usuarixs de la app y con otros elementos. Por ejemplo, en el modo diseñador podemos agregar campo de Texto, una una imagen y un botón y en el modo bloques vamos a indicar que al hacer click en el botón, la imagen y el texto se suben a nuestro perfil y se guardan.

En el modo diseñador, veremos una pantalla de celular en la que iremos

poniendo todos los elementos que necesitaremos. (incluyendo también los elementos que no se ven, como la cámara, el reconocimiento de voz, el grabador de sonidos, las notificaciones, etc).



Elementos en App Inventor

El App Inventor cuenta con **elementos o componentes** que que permiten que programemos toda clase de tareas en nuestra aplicación.

Hay mucha información sobre cada uno de los elementos en el símbolo "?". Incluso hay ejemplos de cómo se usa cada uno.

En este documento vamos a explicar solamente algunos, que son los que más dudas generan. Por supuesto no es necesario usar todos, solamente los explicamos acá para que puedan tenerlo como referencia.

Existen distintos tipos de elementos, que están agrupados en el menú lateral izquierdo en la pestaña de Diseñador.

Algo muy importante para tener en cuenta es que no solo estamos escribiendo una app para que la máquina lo entienda. Tenemos que escribir y armar la app para que otras personas (otros programadores y programadoras) también puedan entender lo que la app hace. Para eso es muy importante, **ordenar** y **nombrar** los componentes que incluimos en nuestro programa. Esto quiere decir,

estructurar los bloques de manera tal que pueda ser fácil entender la secuencia de acciones. Al mismo tiempo, nombrar a los componentes de manera representativa, que puedan auto-explicar que es lo que cada componente hace o cuál es su responsabilidad. Por ejemplo, cuando agregamos un botón el nombre por defecto será *boton1*. Si ese botón nos va a permitir guardar determinada información de un formulario, podemos nombrarlo como *botonGuardar*.

Paleta
Interfaz de usuario
Disposición
Medios
Dibujo y animación
Maps
Sensores
Social
Almacenamiento
Conectividad
LEGO® MINDSTORMS®
Experimental
Extension

Algunos de estos componentes son visibles en la pantalla (por ejemplo, botones, cajas de texto, imágenes), y otros no los vamos en la pantalla (reconocedor de voz, sensor de ubicación).

Interfaz de usuario

Notificador: Este componente puede enviar un mensaje o alerta al usuario. Puede ser para preguntarle si está seguro de que desea borrar un elemento, por ejemplo, o para que tenga diferentes opciones. En la foto de ejemplo para preguntarle si terminó una tarea (para marcarla como lista) o no (no hay cambios). Es importante que le demos la opción de cancelar al usuario, por si llegó a este menú por error.

Visor de Web: Este componente permite mostrar el contenido de una url dentro de la aplicación, Por ejemplo para mostrar un mapa o una página web con información. Lo bueno de este componente es que a diferencia de un link (en el cual el usuario se va de la aplicación) en este caso puede navegar esta página sin salir de nuestra app.

Selector de hora: Este componente permite al usuario elegir una fecha y/u hora. Se pueden usar por ejemplo al registrarse (fecha de nacimiento) o si estamos reportando por ejemplo animales perdidos, indicando cuándo fue la última vez que lo vimos.

Medios

Selector de imagen: permite al usuario elegir una imagen de las que tiene guardadas en el teléfono. Se puede usar por ejemplo para subir una foto a nuestra aplicación. También se puede usar la cámara para sacar una foto nueva desde la app.

Reproductor de video: permite colocar un video en la aplicación para que el usuario pueda reproducir. Sólo reproduce videos en formato .wmv, .3go o .mp4, y no pueden ser más grandes de 1MB. Se puede usar por ejemplo para mostrar una introducción o tutorial. Ojo, este componente es solo para reproducir video, como si fuera un visor de imagen, no para grabar un video nuevo ni para seleccionar uno que tengamos en el celular.

Reproductor y sonido: ambos permiten reproducir sonidos, pero el reproductor es más útil para audios largos, y el sonido para audios cortos, como los “bing” de los notificadoros o para tener efectos. Por ejemplo si es un juego para mostrar una musiquita cuando completa un objetivo.

Sensores

A través de estos elementos, podemos usar funciones propias de los celulares. App Inventor nos permite acceder a ellas y usarlas en nuestra app.

Acelerómetro: Este componente puede determinar si el teléfono se agita o si se sostiene derecho o de costado. Esto es lo que utilizan los celulares para rotar la pantalla, por ejemplo. Nosotrxs lo podemos usar para

Pedómetro: este sensor usa el acelerómetro para medir cuántos pasos ha dado el usuario que tiene en su mano el teléfono. También puede estimar la distancia caminada. Es lo que usa el Pokemon Go para evolucionar los huevos :). Se puede usar por ejemplo si queremos hacer apps de ejercicio.

Giróscopo: es un acelerómetro más preciso que permite saber la posición en la que se encuentra el celular con respecto al piso. Si está perpendicular o en un

ángulo, y en qué ángulo. Es muy raro que usemos esto en nuestra aplicación pero por ejemplo se usa para las apps que permiten ver si un estante está derecho o no.

Sensor de orientación: este sensor recoge la latitud y longitud de la ubicación del celular. Se utiliza en muchísimas aplicaciones, como google maps, waze y otras. Nosotras podemos usarlo para (por ejemplo) marcar un punto en un mapa.

Sociales

Estos componentes permiten acceder a otras funciones del teléfono.

Selector de contacto: permite elegir un contacto de la agenda. Por ejemplo se puede usar para seleccionar personas para contactar ante una emergencia.

Selector de Email: permite escribir un mail. Esto va a abrir cualquier aplicación de mail que tengamos instalada en el teléfono y crear un mail nuevo que se enviará desde el mail que esté configurado en nuestro celu. No es anónimo.

Llamada de teléfono: permite hacer una llamada. Ojo! esto va hacer la llamada como si estuviéramos llamando nosotros desde nuestro teléfono y tiene el costo de una llamada común.

Selector de número de teléfono: permite elegir un contacto de la lista de contactos.

Compartir: permite a los usuarios compartir mensajes, imágenes u otro contenido en la aplicación con otras aplicaciones que tenga en el celular. Esto se puede usar para postear algo que creamos en nuestra aplicación en Twitter o en Instagram por ejemplo

Enviar texto: permite mandar un mensaje de texto desde nuestro número.

Conectar App Inventor al celular

Existen dos métodos para conectar nuestro celular y probar nuestra aplicación. Este modo de conexión no es permanente y puede desconectarse. Es sólo para ir probando lo que vamos programando. Cuando tengamos una versión lista recomendamos instalarla en el celular.

Wifi

Es la mejor manera y la más sencilla. Tanto el celular como la computadora deben estar conectados **a la misma red de wifi** . El celular debe tener descargado el MIT AI2 Companion (que se descarga en el Play Store). Una vez que estén ambos

pasos realizados, ir a nuestro proyecto en App Inventor y hacer click en Conectar > AI Companion.



Se abrirá un código QR que vamos a escanear con la aplicación “MIT AI2 Companion” . También se puede copiar el código a mano sin escanear el QR.

Datos

Variables

Las variables son datos cuyo valor cambia, ya sea por el paso del tiempo o por los eventos que suceden en la aplicación. Por ejemplo, cuando un juego va sumando el puntaje, el primer valor de puntaje es 0, luego, se le suma 10 y se guarda el puntaje. El puntaje ahora es 10, entonces cuando sume otros 10 puntos, será 20, y así siempre se guardará con el mismo nombre. Al finalizar el juego, cuando se pida el puntaje dará el valor final de la variable.

Si necesitamos guardar un valor como este por ejemplo para mostrarlo al usuario y que su valor actualizado se vaya mostrando podemos “guardarlo” en una variable. También se puede usar por ejemplo para pasar información de una pantalla a la otra.

Es importante destacar que esta info no queda guardada para siempre en el teléfono. En cuanto cerremos la aplicación las variables se borran. Si queremos guardar info de forma permanente ir al apartado Almacenamiento.

Listas

Una lista es una forma de guardar un grupo de datos. Si en vez de querer guardar un solo valor (como en una variable) necesitamos guardar varios, se puede usar una lista.

Los elementos se guardan en orden, con un índice en particular. Por ejemplo, un listado de nombres: 1. Karina, 2. Esteban, 3. Sebastián, 4. Melisa. Siendo el número el índice y el nombre el elemento de la lista.

Para ubicar un elemento de la lista, usamos su índice, y así podemos eliminar o reemplazar elementos. Podemos crear listas con elementos fijos, o guardar la lista en una variable y agregar o sacar elementos.

Pasar de pantalla

Casi todas las aplicaciones necesitan de varias pantallas para funcionar. Para hacer esta tarea de manera sencilla es muy importante que las pantallas sean nombradas de manera representativa para que sea más fácil ir y venir entre ellas.

En el caso de App inventor, la única pantalla que no puede ser renombrada es la primer pantalla en crearse.

Para pasar de pantalla, entonces, debemos conocer el nombre de la pantalla a la que queremos pasar y luego programar lo siguiente:

En este ejemplo, queremos que al apretar el botón “Siguiente” se pase a la “Pantalla2”.

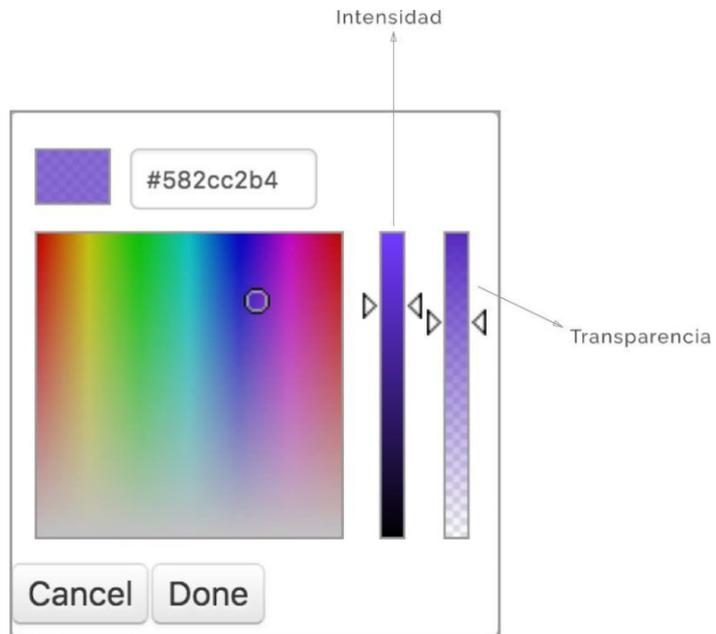


Al pasar de pantalla tenemos también la opción de enviar de una pantalla a la otra ciertos valores, llamados valor inicial. Esto nos permite, por ejemplo, pasar el valor de una variable de una pantalla a la otra. Por ejemplo, si queremos que siempre se muestre el nombre del usuario y pedimos que lo ingrese en la primera pantalla, sucesivamente pasaremos el nombre como valor inicial.

Cómo crear y utilizar colores en app inventor

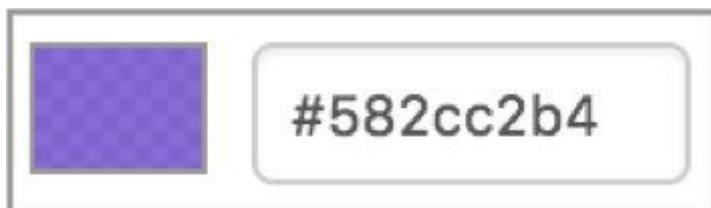
Seguramente vamos a querer usar colores en nuestra aplicación: botones, textos, mensajes. Por ejemplo, si una persona ingresa mal su usuario, queremos que lo note mostrando un mensaje *“Hey! parece que escribiste mal tu usuario.”*, se lo mostramos en color rojo para que note que algo se ingresó incorrectamente.

Dentro de las propiedades de cada bloque, existen algunas que sirven para definir colores al componente; por ejemplo, el color de fondo, el color de los textos, etc. El selector ofrece una serie de colores (azul, verde, negro, etc) para elegir, pero si quisiéramos un color que no se encuentra dentro de lo que están pre-armados, el selector nos permite definir nuestros propios colores con la opción **Custom**.



La opción “Custom..” nos abrirá un cuadro con una paleta de colores donde podremos elegir el color que queremos en sus diferentes gamas. También nos permite definir la **intensidad** del color (más claro u oscuro) y con qué nivel de **transparencia** queremos que se vea.

En el cuadro nos mostrará en la parte superior como se vería nuestro color junto con un **código de color** que se puede copiar por si queremos usar el mismo color en otro lado.



Condicionales

Hay muchas situaciones en las que una siguiente acción depende de algún valor que se ingresó o algo que se seleccionó previamente. El modo condicional es aquel que se usa en frases como “Si yo tuviera un auto, **saldría** todos los fines de semana con mis amigos”. Los condicionales en programación cumplen una

condición no se cumple (falso), saltará el código que se encuentra en “entonces” y correrá sólo el de “si no”.

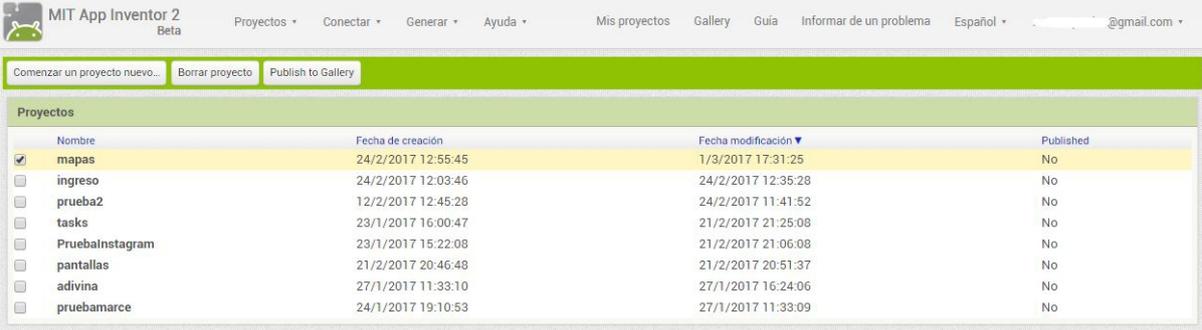
Es posible agregar tantas condiciones como sean necesarias. En los condicionales encadenados, es muy importante tener en cuenta el orden en el que el programa va evaluando las condiciones. El código siempre comenzará con la primera condición e irá de arriba hacia abajo hasta que encuentre una condición que se cumpla. Cuando encuentra la condición verdadera, va a realizar el código que se encuentra asociado a ella.

Subir el proyecto a la galería

Antes de subir el video de finalización del Club a youtube, deberán subir la aplicación a la galería de App Inventor. De este modo, todas las participantes de los programas de Chicas en Tecnología, y usuarios de App Inventor podrán ver su aplicación, descargarla e inclusive ver el código y aprender de él.

Antes de subir el proyecto a la galería, en las propiedades de la pantalla Screen1 tendrán la posibilidad de subir un ícono para la app, cambiarle el nombre (se pueden poner espacios), y poner el número de versión (por ejemplo 1.0, 1.1, 2.0, dependiendo de los cambios que hayan hecho).

Una vez que hayan hecho esto, deben ir a la sección **Mis proyectos** , seleccionar el proyecto y hacer click en **Publicar en Galería (Publish to Gallery)**



The screenshot shows the MIT App Inventor 2 Beta web interface. At the top, there is a navigation bar with options like 'Proyectos', 'Conectar', 'Generar', 'Ayuda', 'Mis proyectos', 'Gallery', 'Guía', 'Informar de un problema', and 'Español'. Below the navigation bar, there are three buttons: 'Comenzar un proyecto nuevo...', 'Borrar proyecto', and 'Publish to Gallery'. The main content area displays a table titled 'Proyectos' with the following columns: 'Nombre', 'Fecha de creación', 'Fecha modificación', and 'Published'. The table contains several rows of project data, with the first row 'mapas' highlighted in yellow and having a checked checkbox in the 'Published' column.

Nombre	Fecha de creación	Fecha modificación	Published
<input checked="" type="checkbox"/> mapas	24/2/2017 12:55:45	1/3/2017 17:31:25	No
<input type="checkbox"/> ingreso	24/2/2017 12:03:46	24/2/2017 12:35:28	No
<input type="checkbox"/> prueba2	12/2/2017 12:45:28	24/2/2017 11:41:52	No
<input type="checkbox"/> tasks	23/1/2017 16:00:47	21/2/2017 21:25:08	No
<input type="checkbox"/> PruebaInstagram	23/1/2017 15:22:08	21/2/2017 21:06:08	No
<input type="checkbox"/> pantallas	21/2/2017 20:46:48	21/2/2017 20:51:37	No
<input type="checkbox"/> adivina	27/1/2017 11:33:10	27/1/2017 16:24:06	No
<input type="checkbox"/> pruebamarce	24/1/2017 19:10:53	27/1/2017 11:33:09	No

Una vez publicado, podrán ver que dice “Si” en la columna Published.

Subir el proyecto a la galería es muy importante

Compartir la aplicación

Una vez que esté terminada, es posible compartir la aplicación con amigos, familiares y compañeros. Existen dos modos de hacerlo:

- Las apps programadas en APP Inventor se pueden subir al **Play Store** .

Si nuestro interés está en subirlas, tenemos que:

1. Descargar el archivo .apk haciendo click en la opción **Generar** de la barra de menú.

función similar: permiten a los programas tomar decisiones en base al cumplimiento de ciertas condiciones.

Nuestro ejemplo podría ser mostrado del siguiente modo:

¿Tengo un auto?	
Sí	No
Salgo todos los fines de semana con amigos	No salgo todos los fines de semana con amigos

Podemos pensar en varios ejemplos en los que esto nos sería útil en programación, como por ejemplo el siguiente:

Hice una aplicación que me permite saber cómo se pronuncia una palabra, para eso, puse un campo de texto para ingresar la palabra, un botón y el componente de texto a voz.
¿Pero qué sucede si el usuario no ingresa texto y aprieta el botón? En ese caso, quiero que lo indique con una notificación.

Usuario aprieta el botón	
¿El campo de texto está vacío?	
Sí	No
Notificar al usuario que no ha ingresado texto	Llamar al texto a voz y pronunciar el texto ingresado.

En App Inventor, el gráfico anterior se vería así:



Este grupo de bloques funciona del siguiente modo:

Cuando el usuario haga click en el *boton_leer_texto* , la aplicación se fijará si se cumple la condición, si se cumple (es verdadera) correrá el código que se encuentra en “entonces” y saltará el siguiente (“si no”). En cambio, si la

2. Dentro de generar, seleccionar “App (guardar archivo .apk en mi ordenador)”.



3. Elegir un lugar en la computadora donde guardar el archivo.
4. Una vez descargado el archivo debemos ir a la [página de publicación de Google](#) y seguir los pasos para publicar la aplicación en Google Play.



En general, los desarrolladores deben pagar para poder subir su aplicación a Google Play. Sin embargo, Chicas en Tecnología tiene una cuenta allí para que las participantes que quieran publicar sus aplicaciones, puedan hacerlo de forma gratuita.

Conocé [Grandes Mujeres - Chicas en Tecnología](#), una aplicación creada por el equipo CET.

Si querés ver cómo se programó esta aplicación, podés entrar al siguiente link: ai2.AppInventor.mit.edu/?galleryId=6386814306484224.

Como ya sabemos, las aplicaciones van sufriendo cambios y son mejoradas constantemente. Por esta razón podemos definir el **número de versión** de nuestra aplicación. Si quieren subir una nueva versión de la misma aplicación, deben modificar estos datos en la primera pantalla, en las propiedades de pantalla.



Al final encontraremos los siguientes campos:



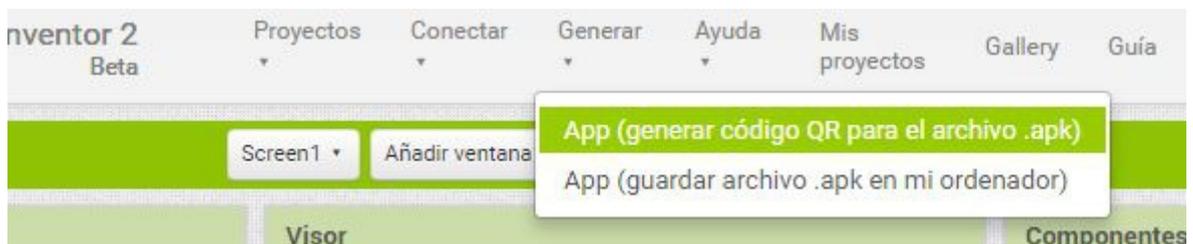
The image shows a form with two input fields. The first field is labeled 'CódigoDeVersión' and contains the number '1'. The second field is labeled 'NombreDeVersión' and contains the text '1.0'.

El Código de versión es un número que no será visible para los usuarios de Google Play. Se utiliza para ver si ha habido cambios en la aplicación. Comienza en 1 y debe ser incrementada en cada cambio, aunque sea pequeño.

El nombre de la versión es aquel que sí ven los usuarios. Se utiliza el siguiente código: se incrementa el número de la izquierda si hay un cambio grande, y el de la derecha cuando los cambios son menores.

Tanto el código de versión como el nombre de versión deben ser modificados cuando subimos una nueva versión de la aplicación. No es necesario hacer estos cambios cuando la subimos por primera vez.

- También se puede generar un QR e instalar en el teléfono. Esto se hace haciendo click en Generar y luego en App (generar código QR para el archivo .apk). Tengan en cuenta que este código sólo dura 2 hs.



Recomendamos compartir la aplicación de este modo, o descargar el archivo .apk y subirlo a algún servicio de almacenamiento para enviarlo a sus amigos.

Acciones sincrónicas y asincrónicas

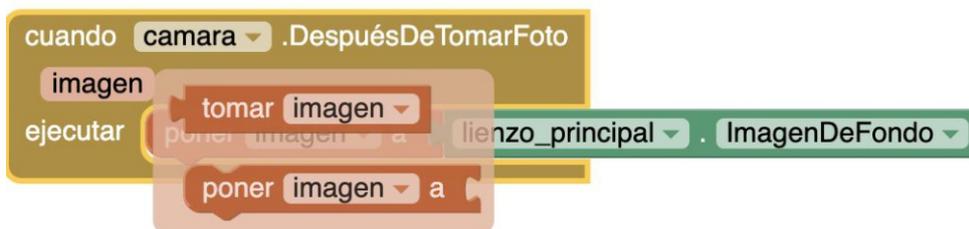
Las acciones **sincrónicas** son aquellas que suceden de manera automática. Por ejemplo, en el ejercicio de la adivina al mover nuestro celular nos da una respuesta automáticamente.



Las acciones **asincrónicas** son aquellas que no suceden al mismo tiempo ni automáticamente. En general, este tipo de acciones dependen de contenido que ingresa el usuario (como fotos o texto) pero puede ser por ejemplo cosas que no controlamos, como por ejemplo “que entre una llamada” o “llega un mensaje de texto”.

Por ejemplo: si quiero realizar una aplicación que saque fotos y me permita intervenirlas, debo llamar al componente cámara, que me permitirá sacar una fotografía. Ese componente nos devolverá una fotografía, y allí debo pedirle a nuestro lienzo que utilice dicha fotografía como fondo.

Es decir, le dimos determinada información a nuestro dispositivo, para que con ella realice un proceso, y luego nos devuelva un resultado. Pero este resultado no se mostrará a menos que el usuario le pida al programa, en una acción subsiguiente, que lo muestre. Esto representa un conjunto de acciones encadenadas asincrónicas.



Almacenamiento

Muchas de las aplicaciones que utilizamos cuentan con una base de datos que permite guardar información. Por ejemplo, nuestro usuario y contraseña de las aplicaciones están guardadas en una base de datos, y cuando ingresamos, compara la información que pusimos con la que tiene guardada para dejarnos

acceder. Los puntos en los mapas también pueden guardarse en una base de datos.

El objetivo de Clubes es hacer un prototipo, es decir, una aplicación que nos permite validar la idea pero no necesariamente tiene todas las funcionalidades completas y 100% terminadas. Muchas de estas funcionalidades se van a simular. Tener una base de datos real es una de las funcionalidades que no vale la pena implementar para probar la idea. Si la quisiéramos agregarla hay tiempo después. Por eso esta es una de las actividades de Comunidad! Es muy importante tener esto en cuenta y es por eso que hay que alentarlas a que las soluciones accesibles para ellas son super validas para terminar su proyecto. Insistimos, la idea es que, con esta experiencia, las chicas puedan hacer un prototipo funcional de la aplicación, y luego podrán seguir completándola.

Entonces, para mostrar el funcionamiento de la aplicación, pueden utilizar alguna de las siguientes opciones:

- “Hardcodear”: fijar contenidos directamente en la aplicación. Por ejemplo, usando etiquetas para hacer un listado de contactos o de libros. Es una manera de mostrar cómo se vería la aplicación con contenido inventado por las chicas, y fijo. Inclusive es posible hacer una pantalla de acceso con usuario y contraseña y que sólo se pueda ingresar con ciertos usuarios y contraseñas :D.
- Usar listas: es una opción parecida a la anterior, sólo que en vez de aparecer contenido fijo, es variable, y el usuario puede ingresar los datos que quiere que aparezcan.

Estas opciones permiten mostrar la aplicación y cómo se verá, simulando el uso y el guardado de datos. Recuerden que cuando se cierra la aplicación, cualquier información que hayamos guardado se va a perder y se va a tener que cargar en el próximo uso. Esto es totalmente válido en un prototipo.