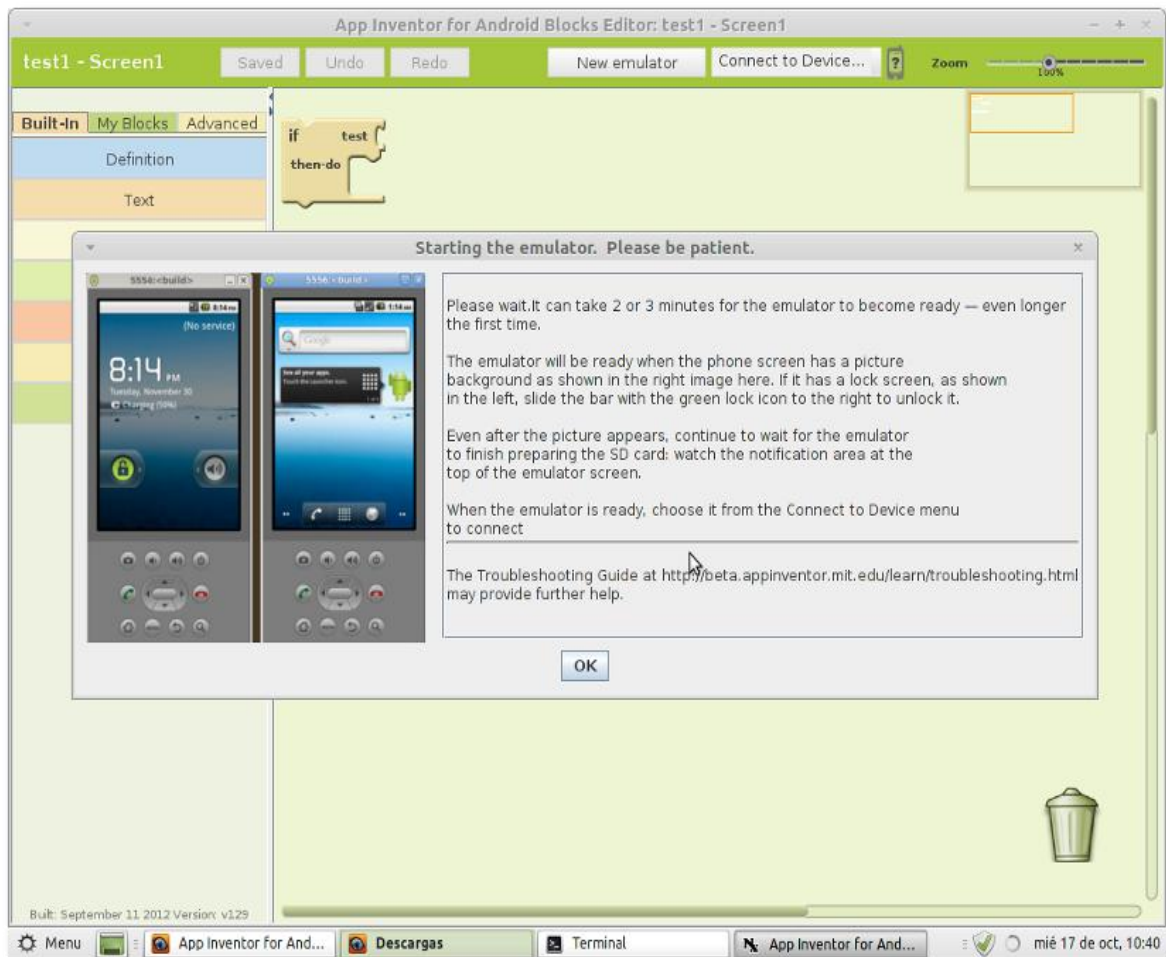
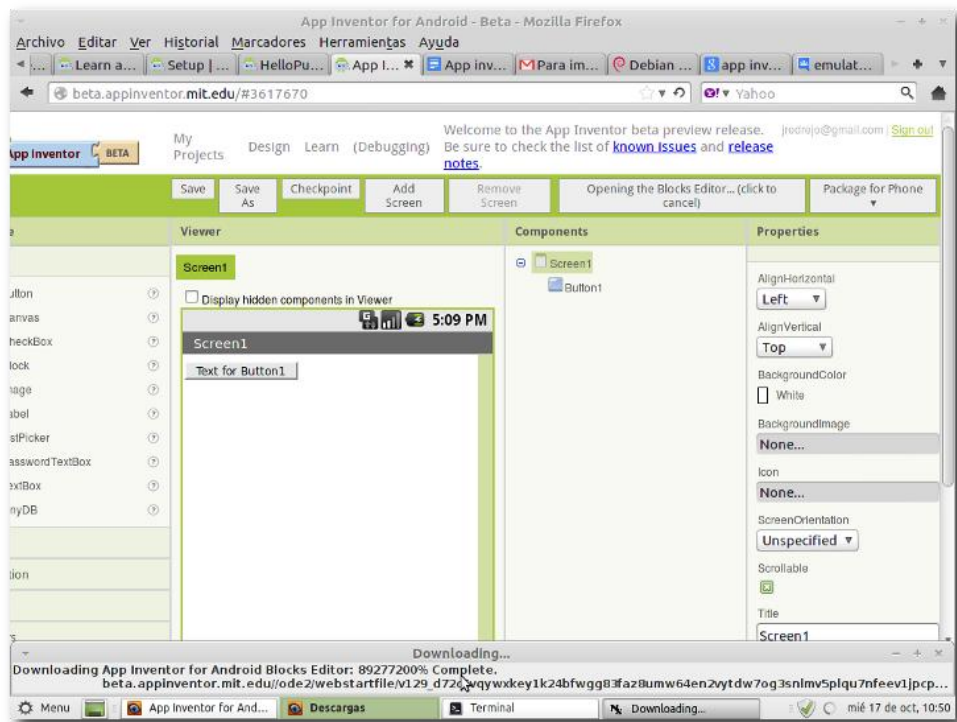


App Inventor es al mismo tiempo un lenguaje de programación, una herramienta de diseño y un entorno de desarrollo de aplicaciones para móviles y tablets que funcionen con el sistema operativo Android. App Inventor permite también ejecutar las aplicaciones en un emulador, por lo que no es imprescindible disponer del teléfono para probar los programas que se hagan.



App Inventor fue desarrollado por Google, pero en Agosto de 2011, fue publicado como Software Libre bajo la licencia Apache 2.0 y su desarrollo se traspasó al Instituto Tecnológico de Massachusetts (MIT). El lenguaje de programación que se usa en App Inventor está claramente inspirado y toma muchos elementos de Scratch³ que está siendo desarrollado por el mismo MIT desde hace más de ocho años.

En la imagen siguiente se puede apreciar la herramienta de diseño en el navegador con la que se construye el interfaz gráfico de la aplicación



¿Por qué App Inventor?

Casi todos los lenguajes de programación tienen una sintaxis que al principiante le suena a chino. Son una mezcla de inglés y extrañas reglas de sintaxis. Como ejemplo, el programa más pequeño que se puede hacer en Java es algo así:

```
class Hola
{
    public static void main(String [ ] args)
    {
        System.out.println("Hola mundo");
    }
}
```

Es evidente que al aprender a programar la sintaxis provoca una curva de aprendizaje significativa. Aunque hay lenguajes más limpios en este sentido que Java, como Python, siempre hay que invertir mucha práctica y tiempo en aprender los comandos, sus reglas de escritura, sentido de los distintos signos de puntuación, etc. Todo este tiempo, es tiempo no empleado en aprender a diseñar algoritmos para resolver problemas, ni para trabajar la creatividad tan necesaria en el mundo informático.

Con App Inventor se aprende a programar usando bloques de programación. Estos bloques están hechos con elementos comunes a la mayoría de los lenguajes de programación existentes. Se colocan bloques para construir bucles, condiciones, variables, etc. que permiten pensar lógicamente y solucionar los problemas de forma metódica, sin perder el tiempo en encontrar el punto y coma o los dos puntos que están donde no deben y producen errores de compilación o ejecución.

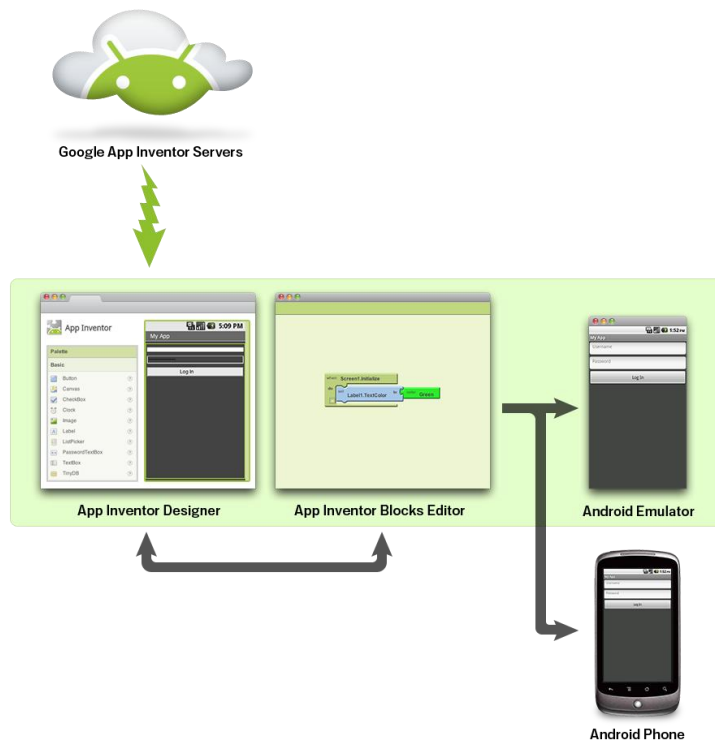


Imagen 3: Diagrama de funcionamiento de App Inventor

Al igual que hicieron con Scratch, los bloques de App Inventor están traducidos a otros idiomas.

Los bloques están clasificados mediante un criterio de colores que indica en parte su función o el componente al que hacen referencia.

Además de los elementos pedagógicos, App Inventor tiene algunas características que lo hacen especialmente adecuado:

1. Es software libre, no es necesario la adquisición de licencias por parte del profesor y los alumnos.
2. Es multiplataforma: requiere sólo de un navegador y la máquina virtual de Java instalada, con “java web start”.
3. Es programación para dispositivos móviles. Los teléfonos y las tabletas están marcando hoy en día buena parte del futuro de la informática y es atractivo para los alumnos poder enseñar en sus propios teléfonos el fruto de su trabajo.

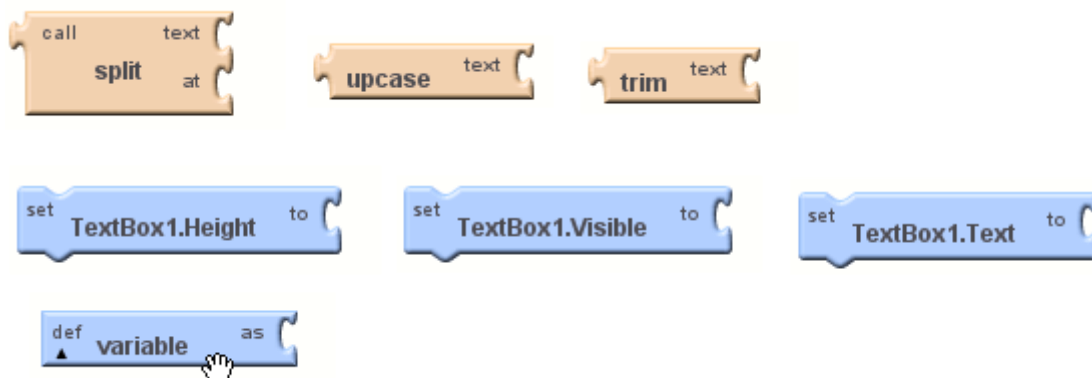
Elementos de programación en App Inventor:

Veamos algunos de esos bloques de App Inventor:

Sentencias:

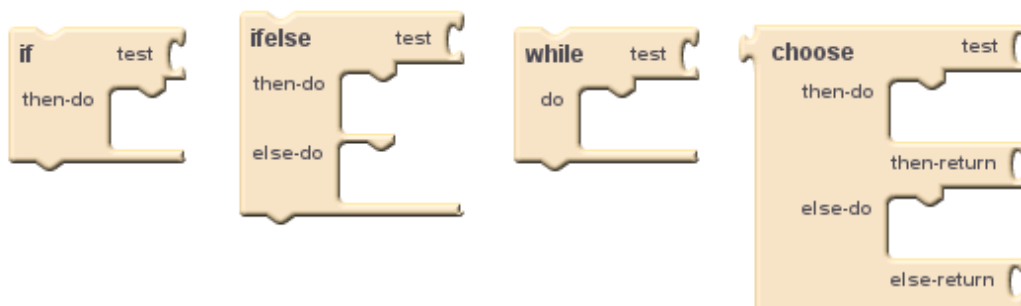
Al programar necesitamos a menudo decirle al ordenador que haga algo. En App Inventor existen numerosos bloques que son sentencias de programación. Se distinguen rápidamente porque expresan una acción a realizar con un verbo en imperativo. Los más habituales son *call*, *set* y *def* (abreviatura de define). Además, los bloques que son sentencias sólo están disponibles en dos colores: azul para modificar variables o propiedades de un objeto y naranja para llamar a funciones.

En la imagen siguiente se ven algunos ejemplos de las sentencias para dividir un texto en partes, convertirlo a mayúsculas, cambiar la altura, texto o visibilidad de un recuadro de texto o definir una variable.



Condiciones:

Con frecuencia necesitamos realizar distintas acciones en función de que ocurra o no algo. En App Inventor para condicionar nuestro programa tenemos las estructuras *if-then*, *if-then-else*, *while* y *choose*:

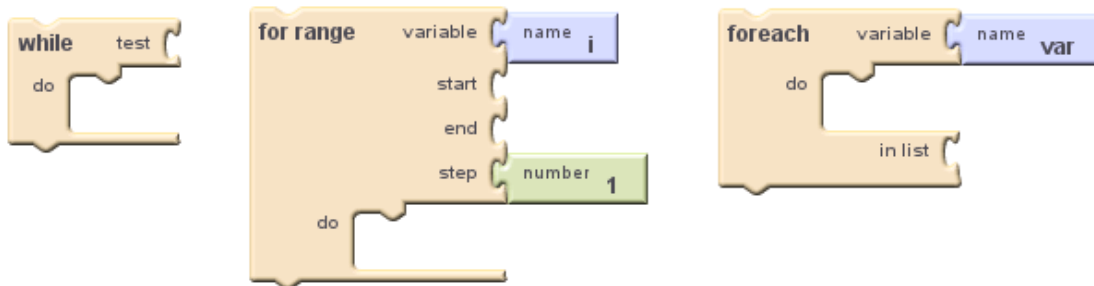


En todos estos bloques se puede ver como tenemos en la parte superior un espacio para encajar el test o pregunta que se hace el programa para decidir por donde continuar su flujo. Las tres primeras estructuras son habituales en la mayoría de los lenguajes, la última, *choose*, es prácticamente igual a *if-then-else* con la diferencia de que permite devolver una variable en función de que se cumpla o no el test.

Bucles

Un bucle provoca la ejecución repetida de varias sentencias.

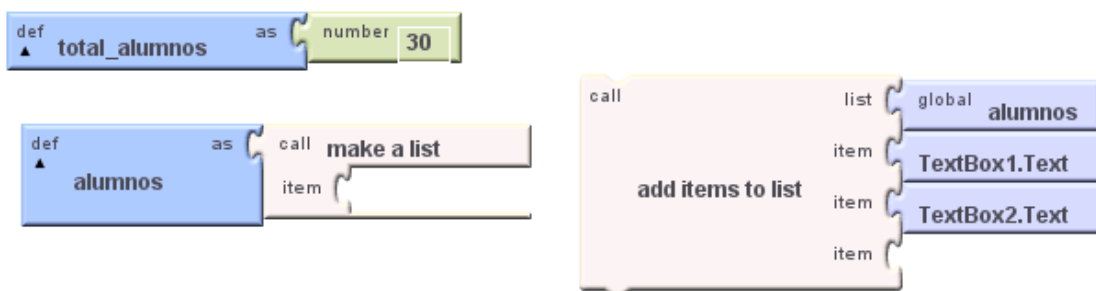
En App inventor disponemos de los bucles *while*, *for range* y *foreach*.



El primero ejecutará el bloque de sentencias que se encajen en el apartado do mientras se cumpla la condición que refleje el test. *for range* es el típico bucle “*for*” de otros lenguajes de programación y, como tal, tiene las opciones de elegir el nombre de la variable que itera, su comienzo, fin y el tamaño de los saltos de iteración. Finalmente, *foreach* es un iterador sobre los elementos de una lista.

Variables

Disponemos de las herramientas para definir y cambiar variables. El tipo puede ser numérico o un texto y “se declara” al asignarle un valor por primera vez.

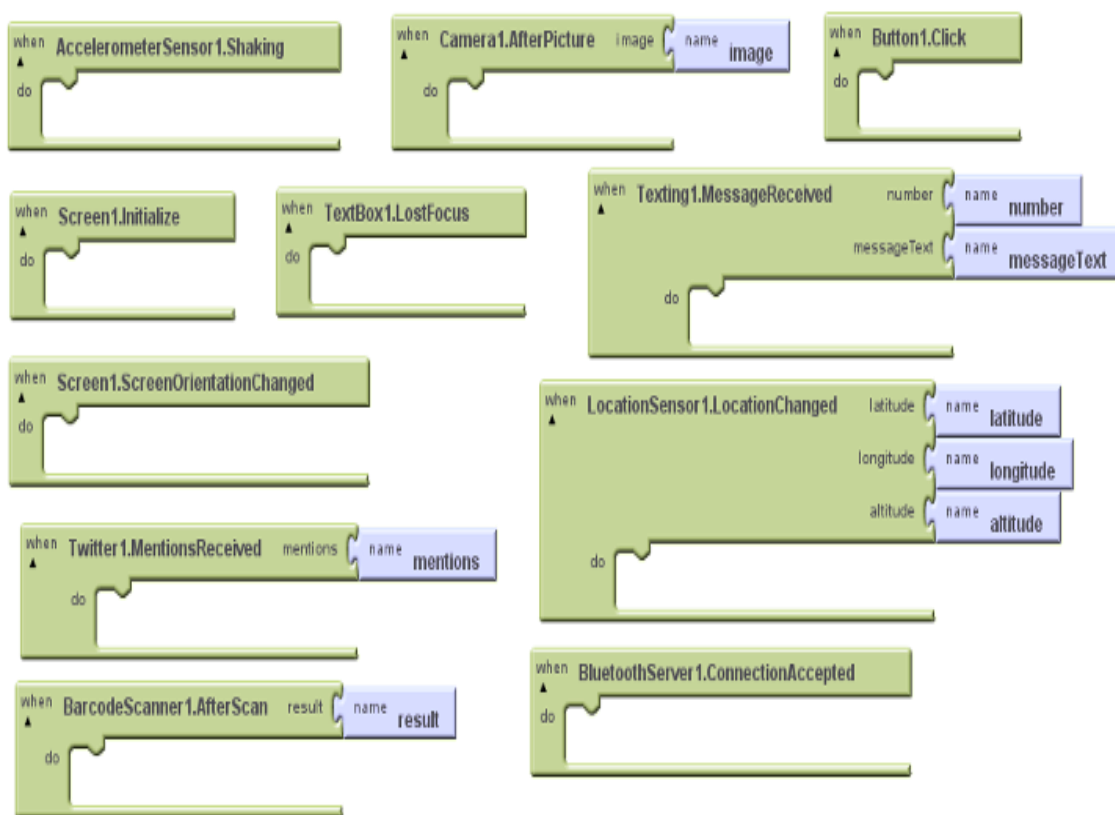


En el gráfico anterior se ve como se declara una variable llamada “total_alumnos” y se le da un valor numérico de 30. Además se ven dos bloques muy importante en App Inventor. Su lenguaje cuenta con una estructura de datos llamada *list* (lista) que es similar a los Arrays de otros lenguajes de programación. Se ve el bloque con el que se define la lista “alumnos” (se define vacía al no añadirle ningún *item*) y también se ve el bloque en el que se le añaden dos elementos a la lista alumnos. En este ejemplo se le añade el texto que contienen los cuadros de texto “TextBox1” y “TextBox2”

Eventos:

Es indispensable que los programas sean capaces de responder a la interacción del usuario con el interfaz. Es ahí donde toman su importancia los bloques de App Inventor, para ejecutar acciones en respuesta a los eventos que se produzcan en el teléfono o el tablet.

El nombre de los distintos eventos dependen del objeto que los provoca. Un botón avisa de cuando se ha hecho clic sobre él, mientras que el acelerómetro avisa de cuando se ha agitado el móvil, la mensajería de cuando se ha recibido un mensaje, el objeto de Twitter de cuando hemos sido mencionados en esa red social, la cámara de fotos de cuando se ha hecho una foto y así sucesivamente.



En la imagen anterior se pueden ver algunos de estos eventos. De esta selección se aprecia que con App Inventor no sólo se dispone de objetos que se pueden colocar en la pantalla como un botón o imagen, sino también de otros muchos objetos que nos dan acceso a las funcionalidades que tienen los teléfonos y tablets Android.

Tenemos acceso a Internet, Bluetooth, hacer llamadas, enviar y recibir mensajes, las coordenadas de posición GPS, etc.

También se puede apreciar como todos los eventos tienen la estructura “When evento do”, y son de color verde. Tienen este color porque son parte de la sección “My blocks” en el entorno de desarrollo. “My blocks” es donde se leen o manipulan las propiedades de los distintos elementos

que hayan sido colocados manualmente en el entorno de diseño: botones, brújula, etiquetas, cuadros de texto, etc.

Como usar App Inventor:

App Inventor requiere que el ordenador tenga alguno de estos sistemas operativos:

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

Necesita también de conexión a Internet (los programas y las aplicaciones se cargan siempre desde Internet) y uno de los siguientes navegadores web con estas versiones mínimas:

- Mozilla Firefox 3.6, sin la extensión NoScript instalada
- Apple Safari 5.0
- Google Chrome 4.0
- Microsoft Internet Explorer 7

Se necesita también una cuenta de correo electrónico de Gmail y Java Web Start instalado en el ordenador.

El software para ejecutar los programas se descarga de <http://appinventor.mit.edu/explore/content/install-app-inventor-software.html>, donde se elegirá una opción u otra dependiendo del sistema operativo que se esté usando. En el caso de Macintosh o Windows se dispone del típico instalador para estos entornos. En sistemas Gnu/Linux Debian o derivados se proporcionan los paquetes deb de instalación⁴ o la opción de instalar directamente desde un archivo comprimido tar.gz⁵. Nota importante: En el caso de instalar en un sistema Linux de 64 bits es necesario asegurarse de que están instalados algunos paquetes para que el software funcione.

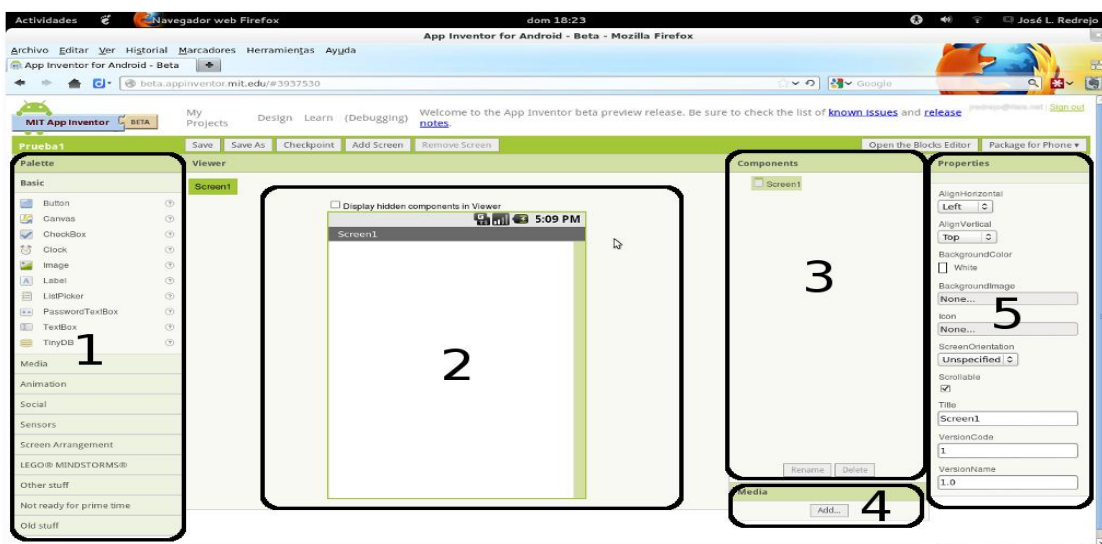
Este software permitirá no solo ejecutar los programas en nuestro móvil o en el emulador, sino que dispone además de un intuitivo interfaz de depuración que ayuda a encontrar errores de programación.

Para que App Inventor pueda comunicarse desde el ordenador con el móvil usando el cable USB es necesario ir a los ajustes del móvil ->Opciones de desarrollador y activar la opción “Depuración de USB”. Importante: Es conveniente instalar este software, pero no necesario para hacer los programas. Si no se instala saldrá un mensaje de error cada vez que se carga la aplicación de desarrollo avisando de que no encuentra el entorno de emulación, pero se puede realizar el programa igualmente.

Si el ordenador está conectado a Internet, dispone del sistema operativo y el navegador adecuado y tiene Java instalado ya se puede empezar a programar. En el navegador hay que introducir la dirección <http://beta.appinventor.mit.edu/>. Entonces pedirá nuestros datos de acceso al correo de Gmail y, si es la primera vez que se accede, pedirá permiso para usar esa cuenta de correo. Una vez que se le concede el permiso se carga la siguiente pantalla:



Se puede apreciar que se trata de una simple vista del listado de proyectos hechos por el usuario. En App Inventor los proyectos se guardan “en la nube”, es decir, en los servidores del MIT. Por tanto, podemos abrir nuestros proyectos desde cualquier ordenador con conexión a Internet sin que tengamos que llevar los archivos de un lado a otro. En cualquier caso, este mismo interfaz da las opciones para descargar el archivo del proyecto si se quiere. Haciendo clic sobre el botón *New*, pedirá el nombre que queramos darle al proyecto e inmediatamente se abrirá la siguiente ventana:

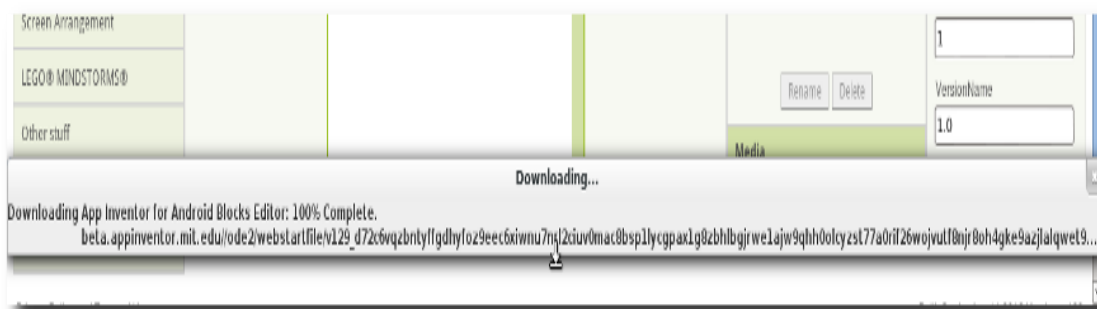


Se trata de la ventana del diseñador en la que se construye, mediante el ratón, sin necesidad de usar ningún lenguaje de programación, todo el interfaz gráfico que va a tener la aplicación. En ella se han marcado con números cada una de sus partes principales:

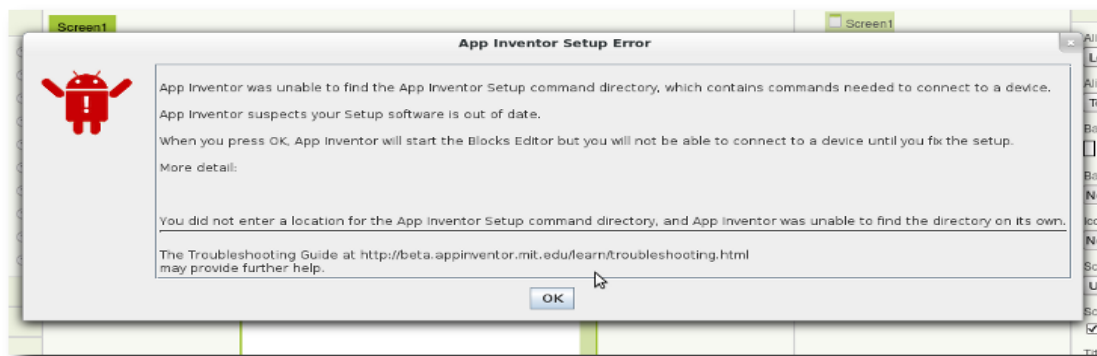
1. La **paleta** contiene todos los elementos que podemos insertar en nuestra aplicación. Hay elementos gráficos como cuadros de texto, botones, lienzo de dibujo (Canvas) y elementos que no se ven en la pantalla del móvil, como base de datos (TinyDB), acelerómetro, cámara de vídeo, etc.
2. **Viewer:** El visor de la pantalla, simula la apariencia visual que tendrá la aplicación en el móvil. Para añadir un elemento a la pantalla hay que arrastrarlo desde la paleta y soltarlo en el visor. Los elementos que no tengan visibilidad hay que arrastrarlos también al viewer y automáticamente se desplazarán debajo de él bajo el epígrafe “Non-visible components”
3. **Components** muestra la lista de los componentes que se han colocado en el proyecto. Cualquier componente que haya sido arrastrado y soltado desde la paleta al visor aparecerá ahí. Si se quiere borrar alguno es en la lista de componentes donde está el botón que permite borrarlo.

4. **Media** muestra las distintas imágenes y sonidos que estarán disponibles para el proyecto. Cualquier archivo de imagen o audio que se quiera usar en la aplicación hay que insertarlo usando este apartado para que esté disponible.
5. **Properties**: cada vez que en el Viewer se seleccione un componente, en Properties aparecerán todos los detalles que se puedan cambiar de ese componente. Por ejemplo, al hacer clic sobre un componente TextBox se podrá cambiar en Properties su color, texto, fuente, etc. Para aquellos que hayan usado antes entornos de desarrollo del tipo de Visual Studio en Windows o Gambas en Linux le será muy familiar esta forma de trabajar.

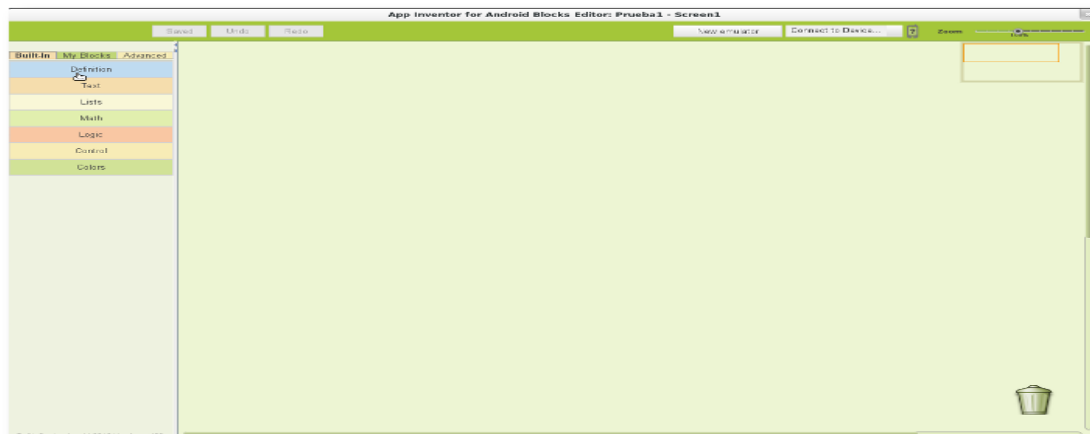
Este es el entorno en el que se diseña la aplicación. Para poder realizar la programación es necesario hacer clic en el botón de la esquina superior derecha con el texto “Open the blocks editor”. Eso hará que el navegador avise de que va a descargar una aplicación que arrancará con Java Web Start. Mientras se descarga la aplicación aparecerá en la parte inferior de la pantalla un texto indicativo de la carga:



Al terminar la carga, si no está instalada la aplicación que permite la ejecución o simulación del móvil aparecerá este mensaje:



Tras aceptar el mensaje aparecerá por fin el editor de bloques de programación:



Esta es la pantalla que usaremos para hacer los programas que interaccionan con el interfaz gráfico. Se puede apreciar cómo está dividida en dos partes muy claras: un enorme lienzo en “blanco” a la derecha donde se irán arrastrando los bloques de programación y a la izquierda los menús que dan acceso a esos bloques.

El menú está formado por tres pestañas: *Built-In*, *My Blocks* y *Advanced*, y bajo cada una de ellas aparecen a su vez distintos menús que permitirán seleccionar los bloques disponibles en cada caso.

1. Built-In contiene los bloques que componen el lenguaje de programación propiamente dicho, distribuidos en las secciones Definition, Text, Lists, Math, Logic, Control y Colors.
2. My Blocks ya se ha nombrado anteriormente y contendrá una lista de los elementos que hayan sido colocados en la pantalla Viewer del diseñador para poder acceder mediante el programa a sus propiedades y eventos
3. Advanced es una pestaña que permite acceder de forma global a algunas propiedades comunes a todos los elementos que hemos colocado en el Viewer y que sean de un mismo tipo. Por ejemplo, al colocar un par de botones aparecerá “Any Button” y se podrán cambiar ahí algunas propiedades que afecten simultáneamente a los todos los botones.

En la parte superior derecha del editor de bloques de programación se pueden ver también los botones que dan acceso a conectar con un teléfono Android que estuviera enchufado al ordenador o lanzar el emulador.

Programando

La mejor forma de ilustrar la forma de trabajar con App Inventor es mediante actividades hechas paso a paso, orientadas para Alumnos Universitarios en facultades de ciencias, o con pequeñas adaptaciones para disminuir la dificultad de los proyectos propuestos pueden ser usados en Bachillerato en la asignatura de TIC.