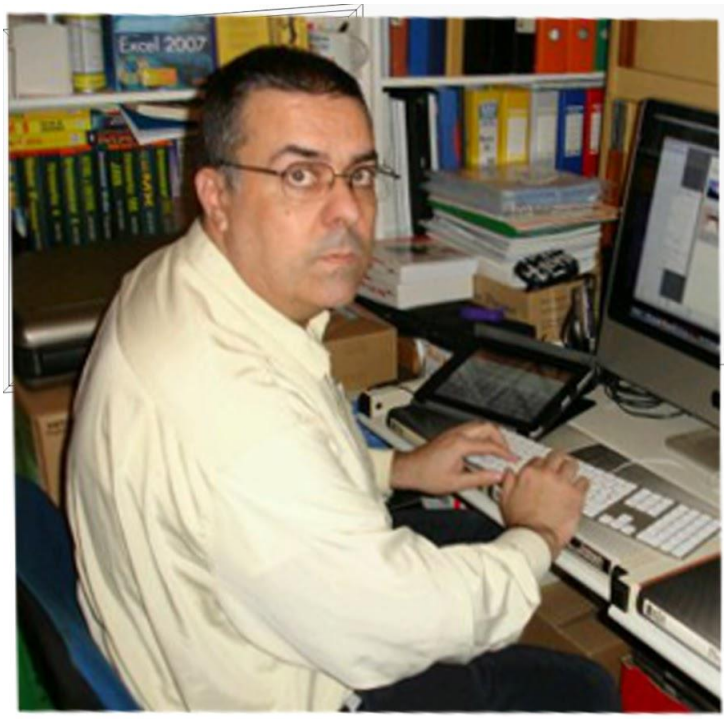


INTRODUCCIÓN A APP INVENTOR

App para móviles

DESCRIPCIÓN BREVE

Este tutorial de iniciación a App Inventor es un resumen de unos fantásticos vídeos recopilados en tutoriales de App Inverntor . Para personas que se quieran iniciar en este fantástico mundo de la programación para móviles Android.

Pere Manel Verdugo Zamora
pereverdugo@gmail.com

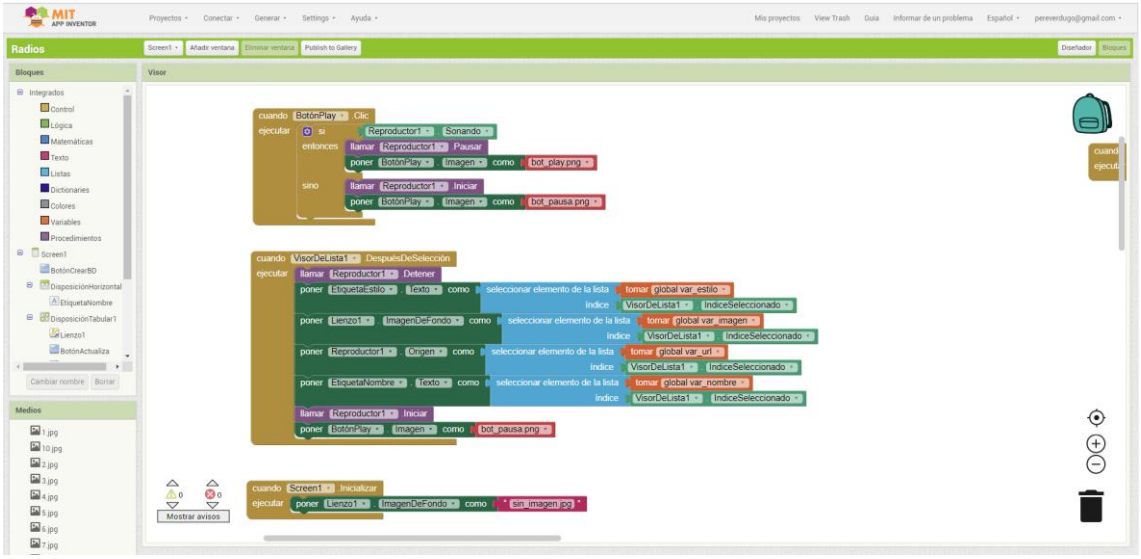
En http://www.peremanelv.com tendrás acceso a más tutoriales.

1.- ¿Cómo usar APP INVENTOR 2?

¿Qué es App Inventor?

App Inventor es un entorno de desarrollo de aplicaciones para dispositivos Android. Para desarrollar aplicaciones Con App Inventor sólo necesitas un navegador web y un teléfono o Tablet Android (si no lo tienes podrás probar tus aplicaciones en un emulador). App Inventor se basa en un servicio web que te permitirá almacenar tu trabajo y te ayudará a realizar un seguimiento de tus proyectos.

Se trata de una herramienta de desarrollo visual muy fácil de usar, con la que incluso los programadores podrán desarrollar sus aplicaciones.



Si tienes un teléfono con el sistema operativo iOS también podrás realizar el curso y ver los resultados en tu móvil, pero no podrás instalar las aplicaciones una vez las hayas terminado.

Lo ideal que tengas un móvil o Tablet con sistema operativo Android.

Solo necesitarás tener acceso a internet un navegador.

DISPOSITIVOS EN LOS QUE FUNCIONA

App Inventor es principio solo funciona para dispositivos móviles Android, pero si podrás utilizar tu dispositivo iOS para ver el resultado de tu aplicación, de forma que las apps que se generen con esta herramienta puedan ser usadas y probadas tanto en dispositivos Android como en iOS.

ALTERNATIVAS ACTUALES PARA iOS.

Si queremos desarrollar App para iOS lo podemos hacer de momento, con otra aplicación muy similar de programación por bloques denominada **Thunkable.**

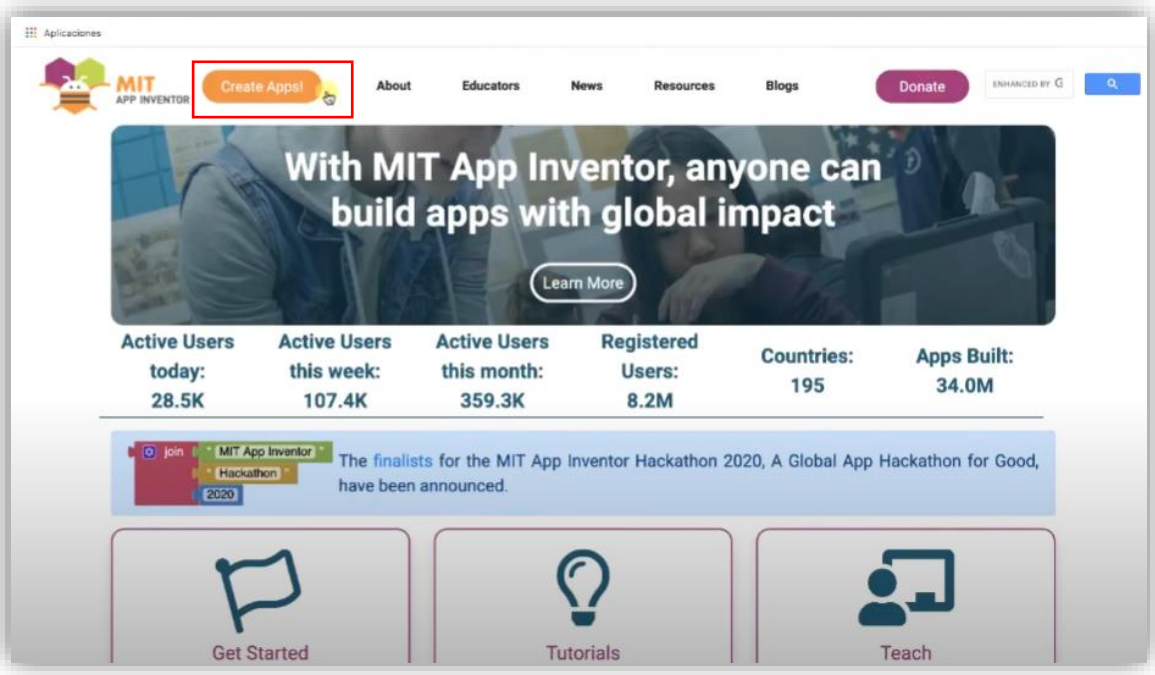
Acceder a App Inventor

App Inventor es un producto desarrollado por el Instituto Tecnológico de Massachusetts (MIT) junto con Google, de manera que como requisito se debe disponer de una cuenta de Google para acceder a la web de MIT App Inventor. Seguidamente podrás ver paso a paso como crear un perfil de App Inventor y así comenzar a programar.

- **Acceder a la web MIT App Inventor**. Accede a la web oficial mediante este enlace: http://appinventor.mit.edu
- **Crear un acceso a la web.** Una vez tengas abierta la web, verás a la barra superior un icono de color naranja donde dice Create app!, haz clic en él y asocia su cuenta de Google para acceder a la plataforma y comenzar a programar. Más adelante verás cómo cambiar el idioma. Si no tienes una cuenta de Google, puedes crearla en: https://accounts.google.com/signup/v2/webcreateaccount?cc=ES
- **Comenzar a trabajar.** Una vez hayas introducido tu cuenta de Google, te aparecerá un texto como el Acuerdo de Licencia de uso, y al pulsar aceptar ya tendrás acceso a la interfaz de programación como se muestra en la siguiente imagen.

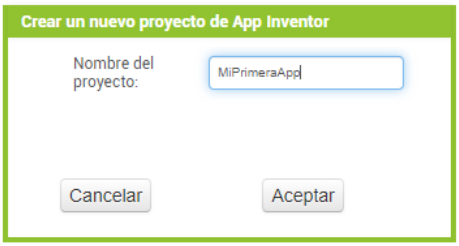
CAMBIAR EL IDIOMA

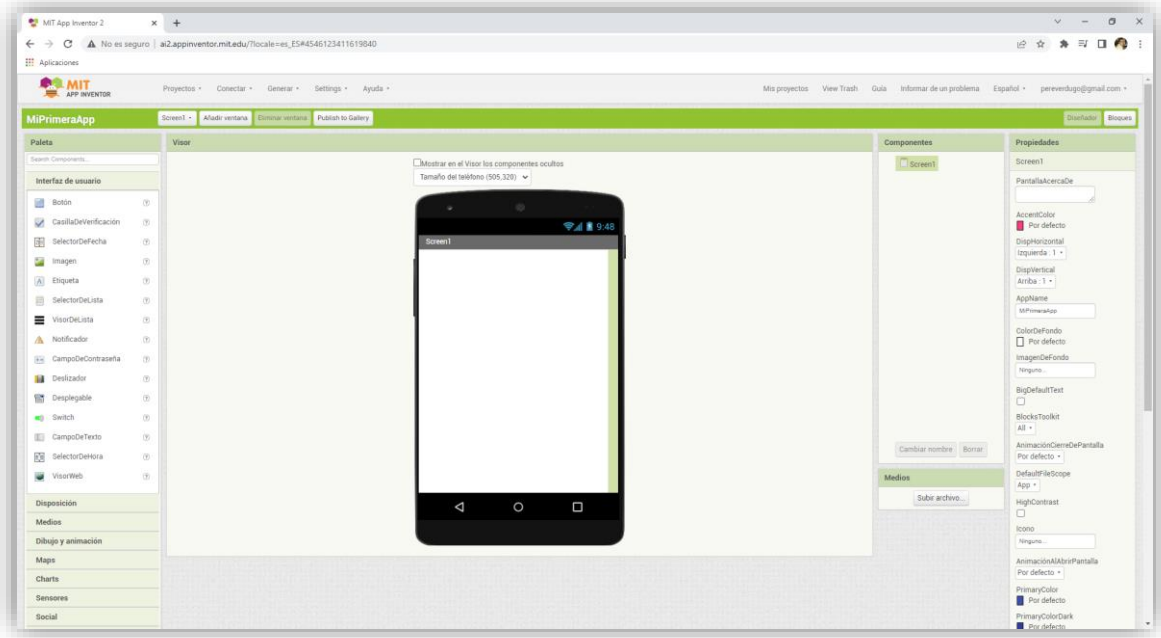
La primera vez que abres la web aparece un cuadro de texto en el centro informándote de que no tienes proyectos, animándote a crear nuevos. La web de App Inventor de manera predeterminada se muestra en inglés, pero en la barra de herramientas superior podrás ver la opción desplegable con todos los idiomas disponibles. Haz clic en la pestaña y selecciona Español.



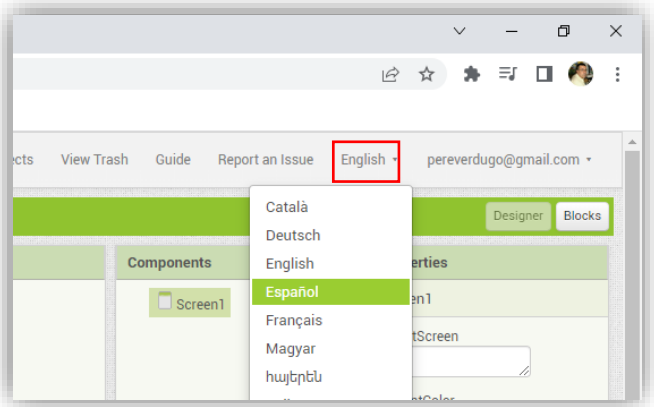
Seleccionaremos Create App!, a continuación nos pedirá la cuenta de Google.

Vamos a crear un nuevo proyecto.





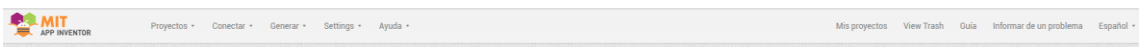
Lo primero será cambiar el idioma si aun no lo has hecho.



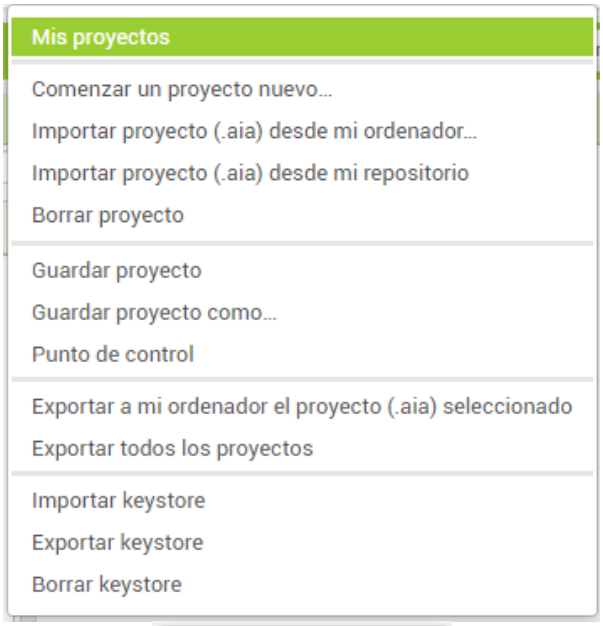
En el menú superior derecha donde pone English abre la lista y selecciona Español.

Barra de Herramientas

La barra de herramientas tiene diferentes opciones que te permitirán acceder a la creación y edición de Apps.

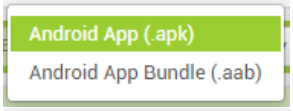


Proyectos:

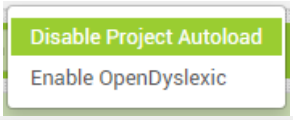


Conectar:

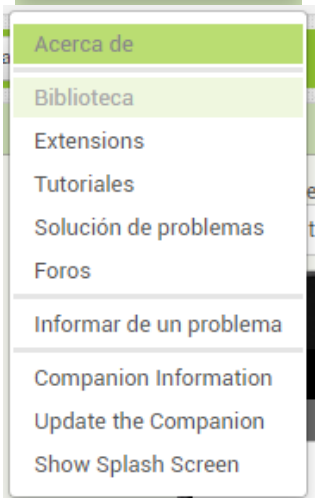
Generar:



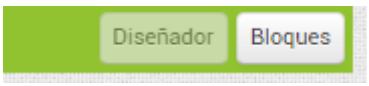
Settings:



Ayuda:



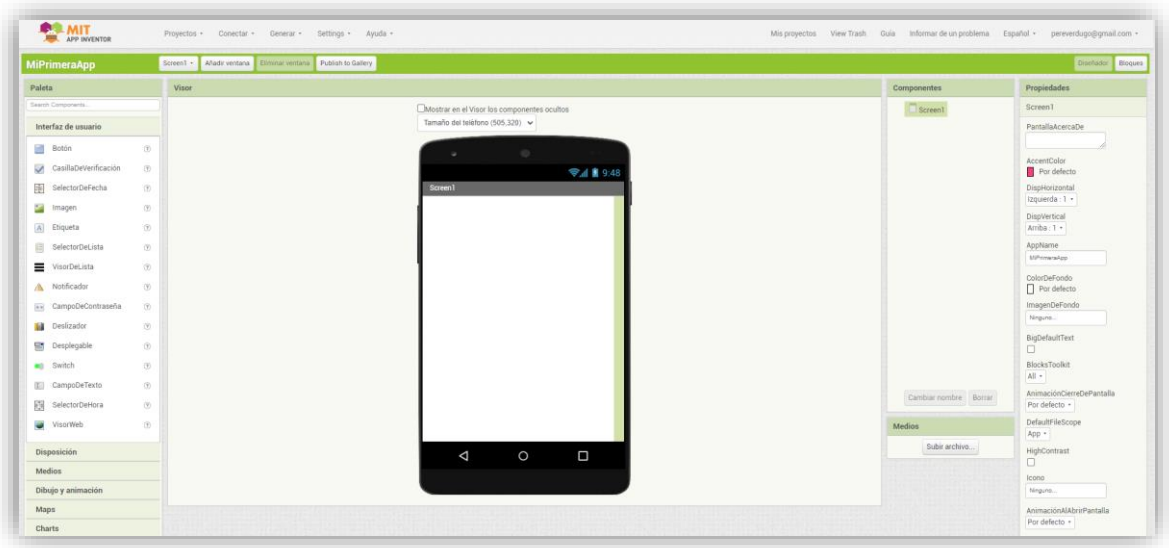
En la parte superior podemos seleccionar si trabajamos la parte de diseño o la parte de programación.



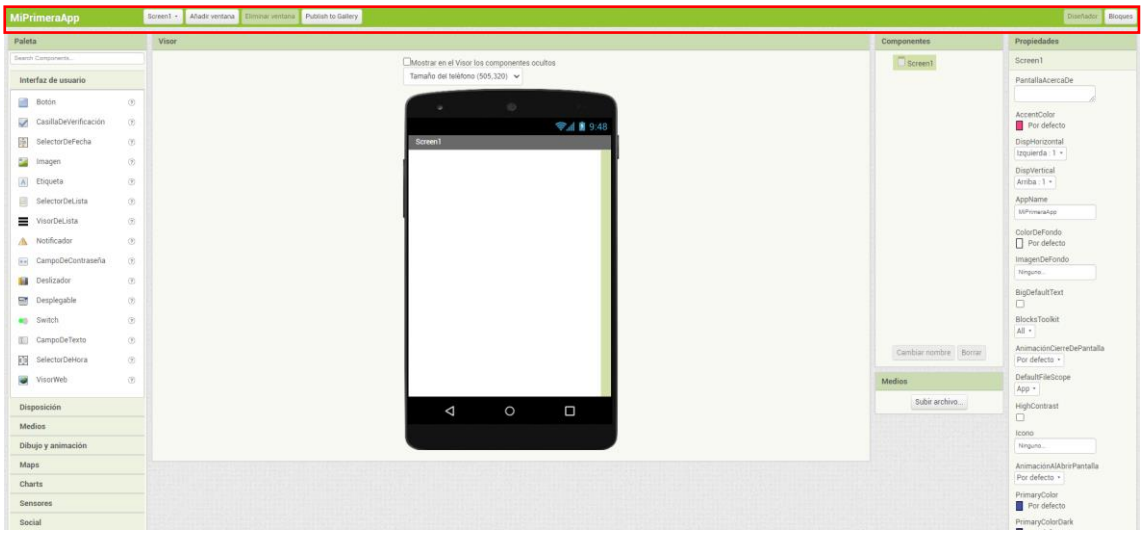
Primero empezaremos a trabajar en la pantalla de diseño.

Realizar la interface de la App: DISEÑADIR

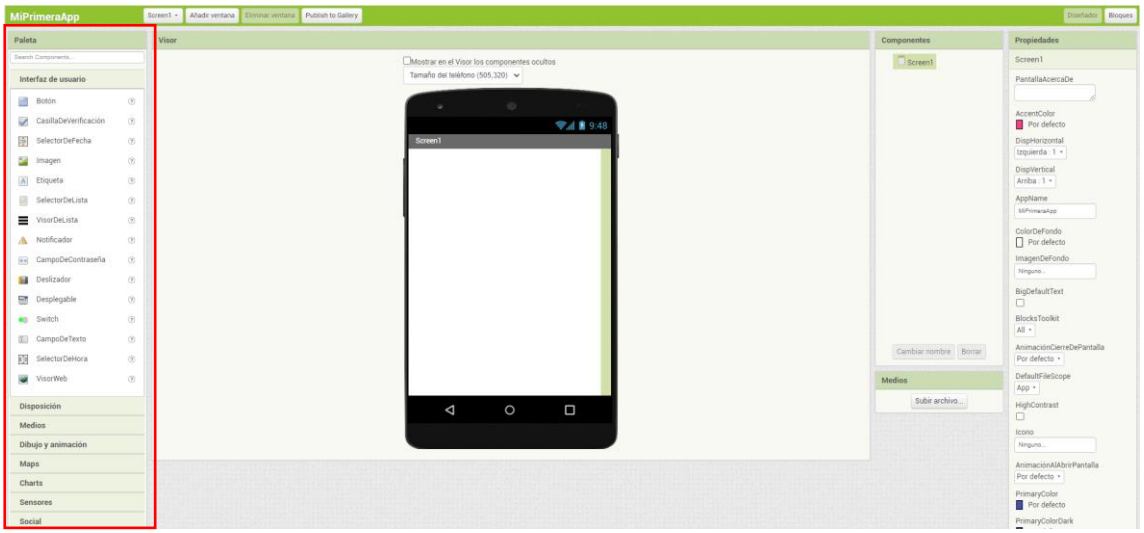
COMPONENTE DISEÑADOR



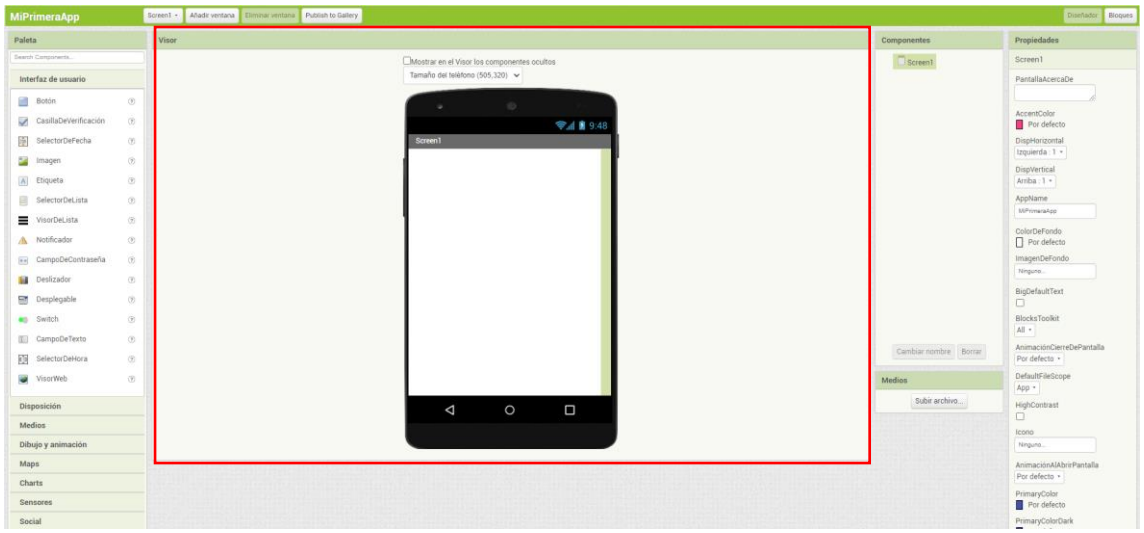
- **Barra de la app abierta.** Está barra de herramientas es la que ofrece las opciones que hacen referencia a la app que actualmente se encuentra abierta. Ventanas: permite cambiar la ventana del programa con la cual quieres trabajar en un momento concreto, ya que una app puede disponer de varias ventanas. Añadir ventanas: permite añadir nuevas ventanas a la app que estás desarrollando.



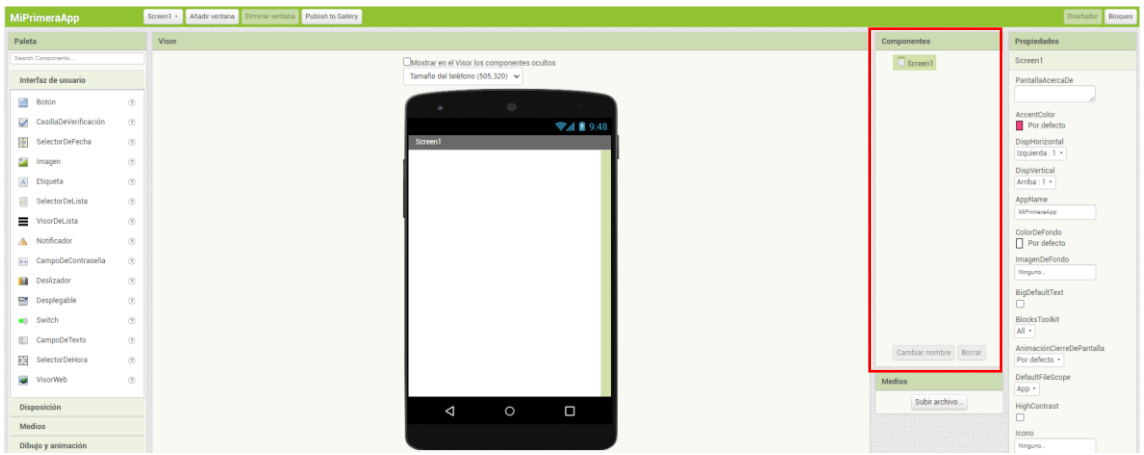
- **Paleta**. Esta sección del entorno situada en el lateral izquierdo de la interfaz de App Inventor, contiene todos los elementos visuales o de otro tipo, que pueden ser agregados y tratados en una App. Al pulsar en cada una de las opciones se despliega, abajo, el menú de elementos correspondientes a esa categoría.



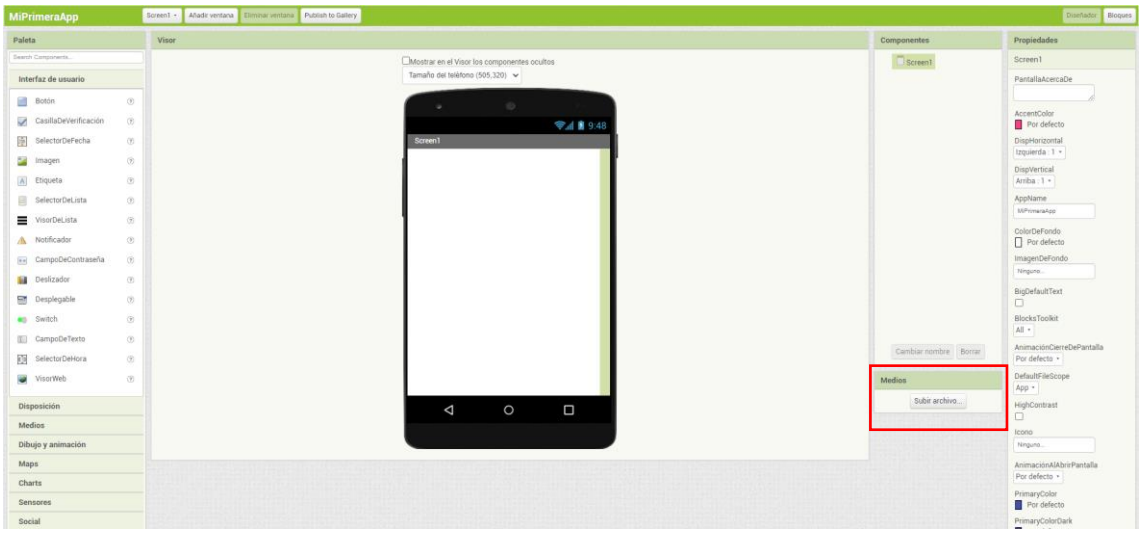
- **Visor**. Este apartado del entorno muestra lo que se verá a la App una vez la instales o emules en un dispositivo. Mediante los componentes de la sección Paleta que irás arrastrando al visor, se ira generando el aspecto visual de tu App.



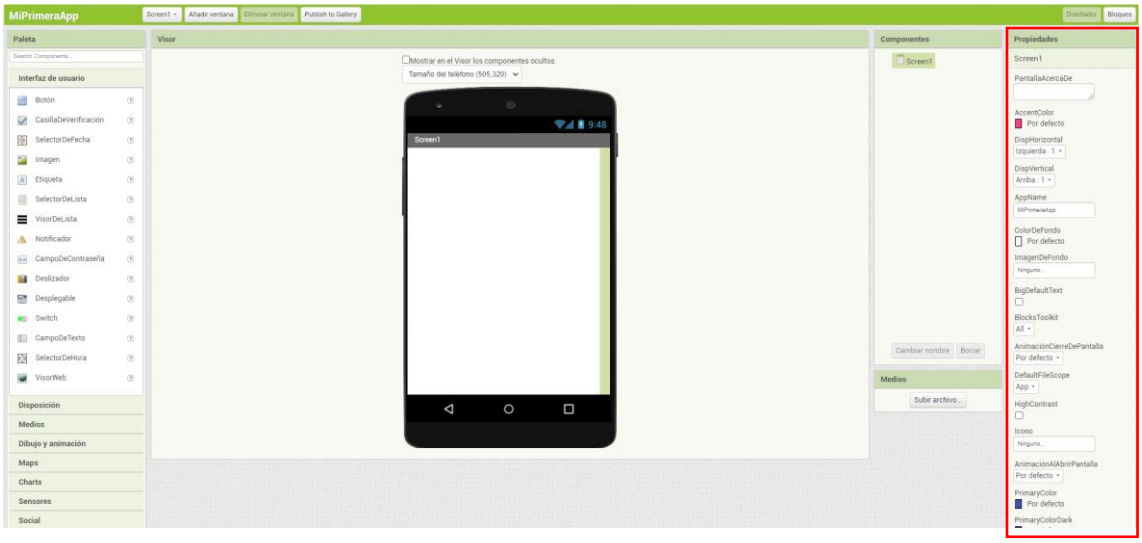
- **Componentes.** Cada vez que se añada un componente al visor de tu App, se irá generando una lista de éstos, de manera que podrás identificar rápidamente con qué objetos estás trabajando en todo momento. Para renombrar o eliminar un elemento no deseado, lo tienes que hacer mediante esta sección del entorno.



- **Medios.** Cuando quieras agregar archivos digitales desde el ordenador, tales como música, fotos o vídeos a la App, lo vas a tener que hacer desde este apartado. El botón Subir archivo permite acceder a los archivos que se encuentran almacenados dentro del ordenador desde el que se está ejecutando la herramienta web.



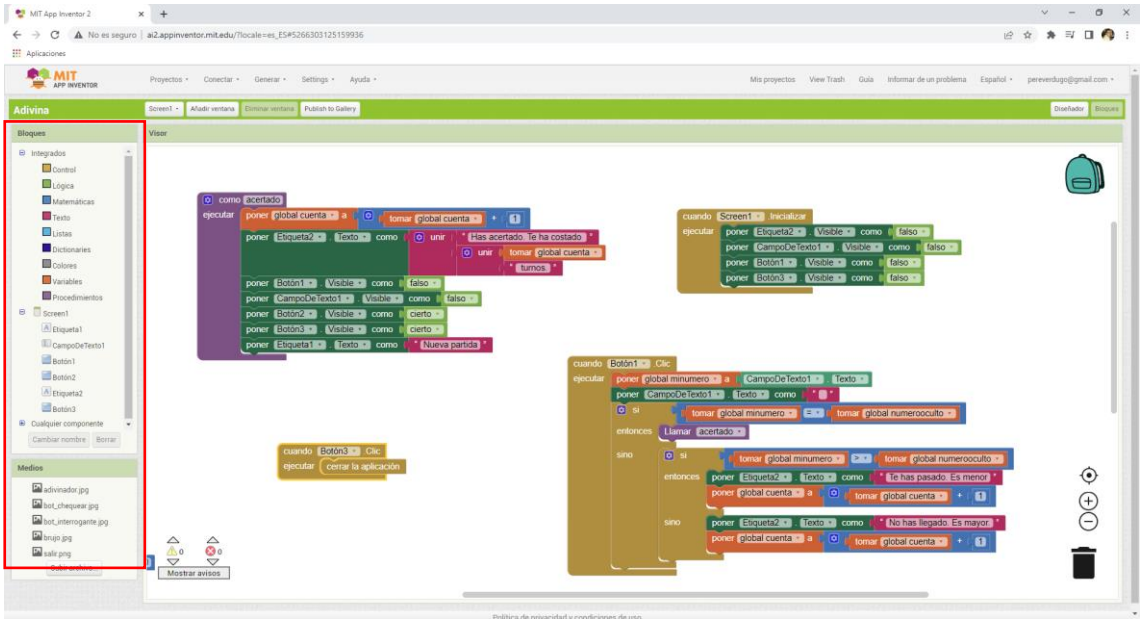
- **Propiedades.** Al igual que cada vez que se añade un elemento al Visor, éste se añada a la lista dentro del apartado Componentes de manera automática, también se va a generar la sección Propiedades donde se van a poder modificar ciertos parámetros del componente del visor seleccionado: el aspecto de un botón, de una imagen…



Programar la App: BLOQUES

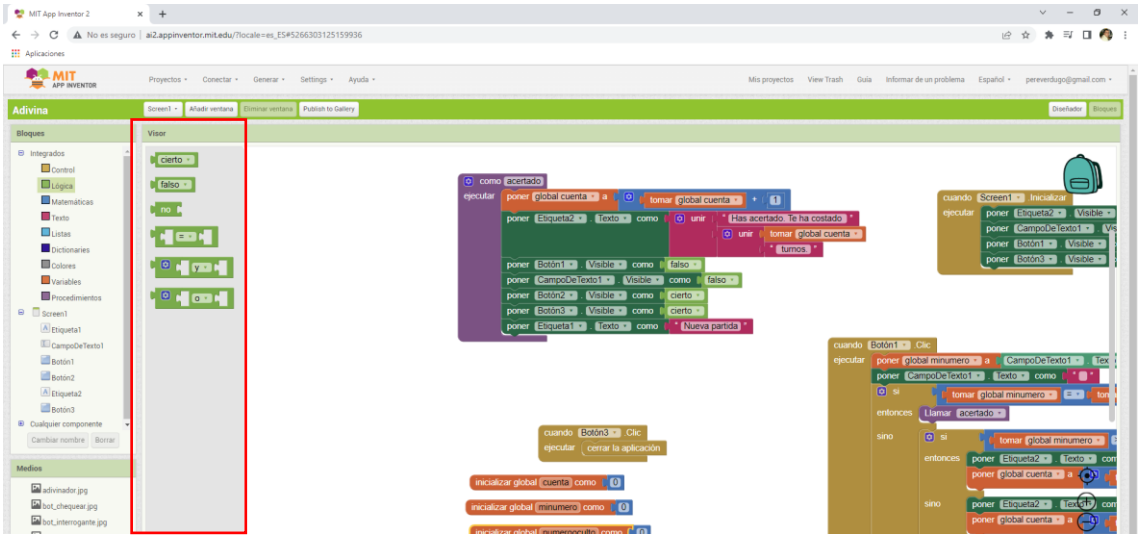
COMPONENTE BLOQUES

Bloques somo ya sabes, por un lado deberás desarrollar visualmente tu App agregando y distribuyendo los componentes dentro de la pantalla, pero también deberás programar con bloques el comportamiento de estos componentes. Para ello, como ya has visto antes, está la sección de Bloques. Cada objeto o componente tiene una serie de instrucciones organizadas por secciones organizadas por secciones de manera predeterminada.



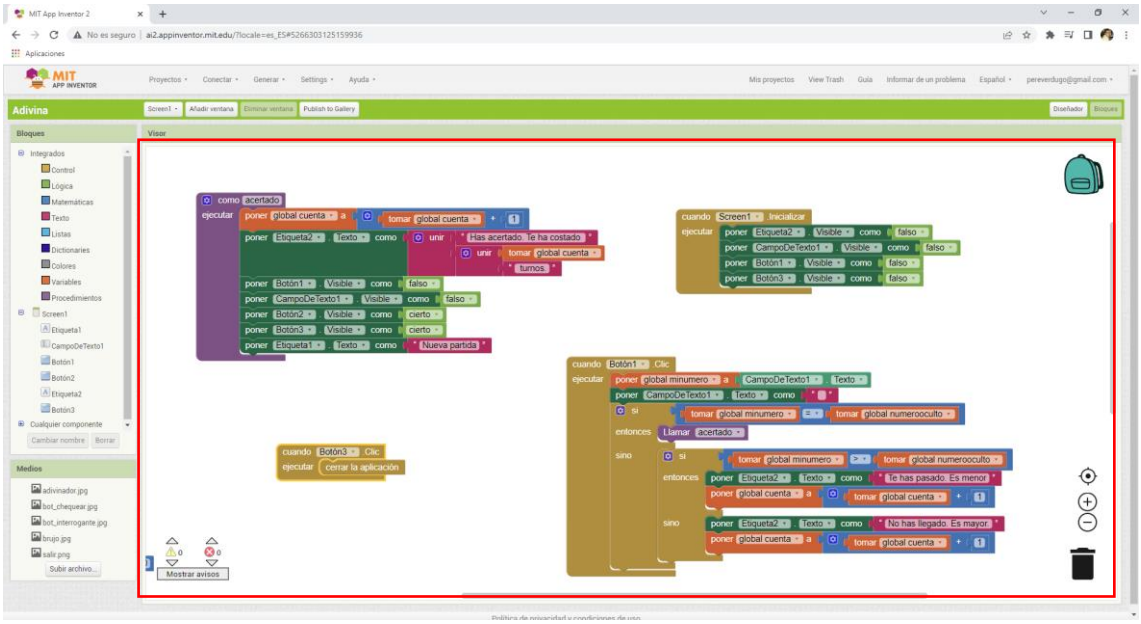
BLOQUES

- **La primera contiene las distintas secciones de bloques con instrucciones** generales que normalmente se pueden aplicar a cualquier componente. Digamos que son las instrucciones comunes.
- **En el segundo apartado encontrarás que aparece el nombre de los componentes añadidos en el visor.** Aquí se da la oportunidad de trabajar con instrucciones propias para este tipo de componentes.



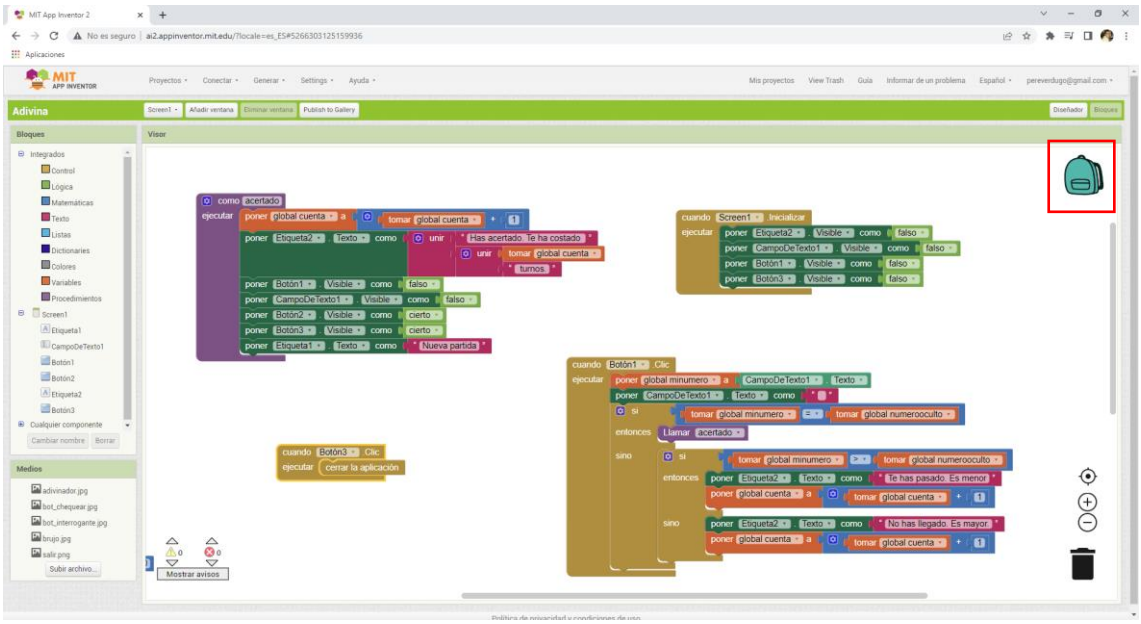
ZONA DE PROGRAMACIÓN

En esta zona es donde vamos construyendo nuestro programa poniendo las instrucciones en bloques unas debajo de otras.



MOCHILA

La mochila le permite transportar bloques a través de los repositorios de su proyectos, lo que permite transferir bloques entre proyectos y entre pantallas. El contenido de la mochila persiste durante una sesión de App Inventor. (NOTA: Cuando salga de la sesión de App Inventor o actualice la página de App Inventor en el navegador, la mochila se vaciará, es decir, se reiniciará).



2.- Programación de dispositivos móviles

Tipo de dispositivos móviles

- **TELÉFONO INTELIGENTES O SMARTPHONE**. Es un dispositivo digital que integra un ordenador en un teléfono móvil. Los sistemas operativos más comunes (Google: Android, Apple: iOS, Nokia: Symbian, Microsoft y BlackBerry: BlackBerry OS). Algunas de sus prestaciones son: pantalla táctil, Bluetooth, cámara de fotos y vídeo.

- **TABLET**. Una Tablet es un ordenador integrado en una pantalla táctil de gran precisión, cuyo tamaño suele oscilar entre 7" y 10". Se manejan mediante el uso de apps de todo tipo. Algunas características son: Utilizan el mismo SO que los teléfonos inteligentes (Android, iOS), memoria de varios GB, conexión a Internet (por redes móviles 4g/5G o a través de WIFI), conexión con otros dispositivos (teclado, altavoces, o mediante puestos USB, HDMI), así como Bluetooth para conexiones inalámbricas, cámara de vídeo y fotos.

DIFERENCIAS ENTRE UNA TABLE CON TECLADO Y UN PORTÁTIL

- El sistema operativo puede ser otro de los puntos que los diferencien. Generalmente, los portátiles tendrán instalado Windows mientras que las tabletas se consideran dispositivos móviles y llevan Android o iOS. Esto puede no ser un inconveniente pero sí será muy importante a la hora de elegir entre un dispositivo u otro dependiendo del tipo de uso que le vayamos a dar. Las manías existen y la interface de Windows no es igual que la de Android, por no citar la diferencia entre las aplicaciones de uno y otro sistema.
- Aunque si existen un aspecto que para muchos es muy importante, ese es el almacenamiento. Mientras que un iPad admite un máximo de 512GB o 64 GB una Tablet, los convertibles tienen un almacenamiento mínimo de 32 GB hasta 2TB, algunos incluso tienen SSD para mejorar su rendimiento y velocidad.

Importancia Programación Dispositivos móviles

¿Por qué desarrollar App?

- Porque hay más móviles en el mundo que personas.
- Porque se usan un 86% de apps, frente a un 14% de navegadores.
- Miramos nuestro teléfono más de 150 veces al día.
- No hay una forma más sencilla de llegar a la gente hoy en día

PLATAFORMAS

- ANDROID VS IOS (PLAY STORE VS APPSTORE)
- ANDROID: 80% IOS (15%)

En AppStore se puede conseguir más ingresos pero inicialmente requiere una mayor complejidad por lo que recomiendan comenzar por PlayStore.

ELEMENTOS IMPORTANTES PARA EL ÉXITO DE UNA APP

1. Posicionamiento en el mercado de aplicaciones (ASO)
2. Cómo MONETIZARLA.
3. Cómo expandir y hacer crecer el número de descargas.

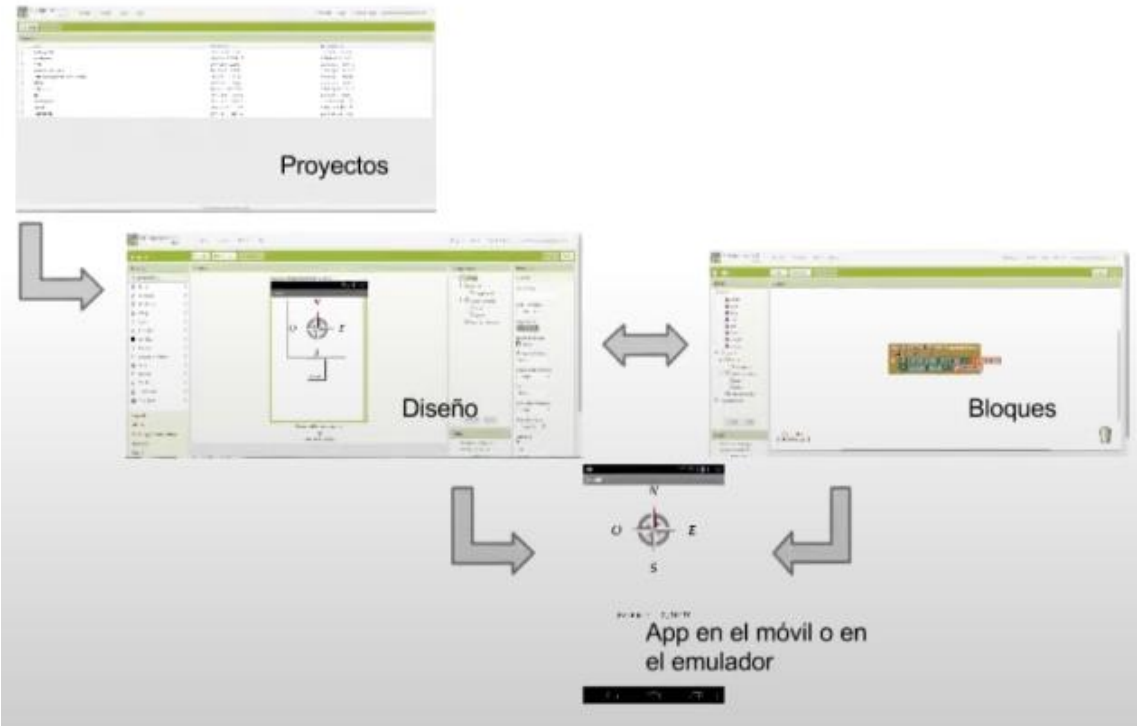
Etapas en el desarrollo de aplicaciones móviles

El desarrollo de una Aplicación móvil engloba dos grandes procesos/etapas:

1º DEFINICIÓN DE LA INTERFAZ. En primer lugar deberemos de definir la interfaz de nuestra aplicación, seleccionando y colocando los elementos que queremos que aparezcan en la pantalla de nuestra App.

2º PROGRAMACIÓN POR BLOQUES. Ahora realizamos la programación de los componentes que hemos dispuesto en la pantalla de nuestra aplicación para indicar como se debe de comportar cada uno de ellos.

ESQUEMA DESARRROLLO DE APLICACIÓN MÓVIL



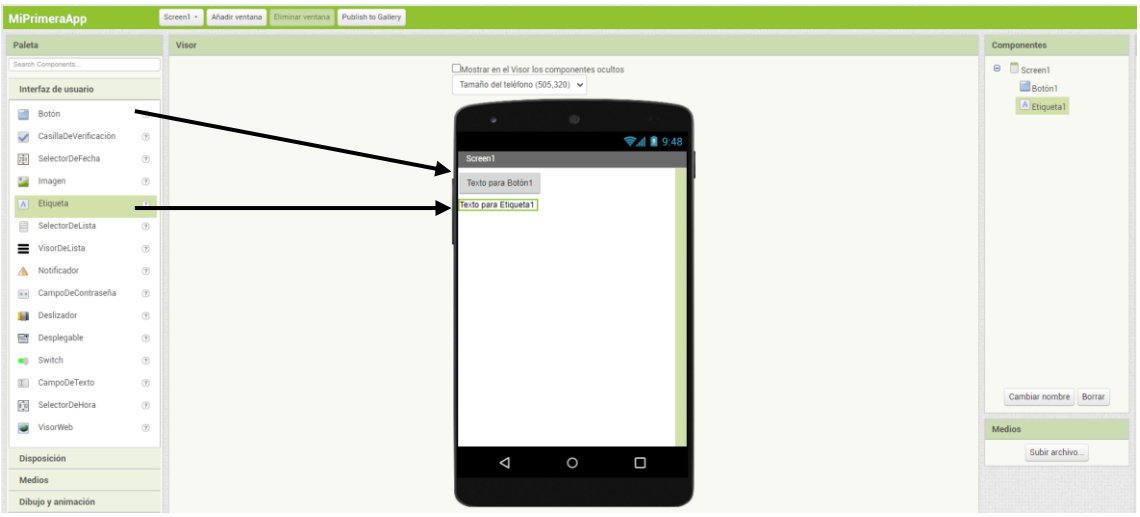
3.-Mi primera App con App Inventor

Desarrollo App: “Hola Mundo”

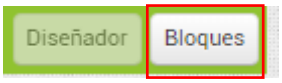
Vamos a crear nuestra primera App de App Inventor. Vamos a crear la típica primera aplicación que se suele realizar en cualquier lenguaje de programación: “Hola Mundo”. Crearemos una Aplicación donde aparezca un botón que cuando se pulse muestre en pantalla el texto “Hola Mundo”.

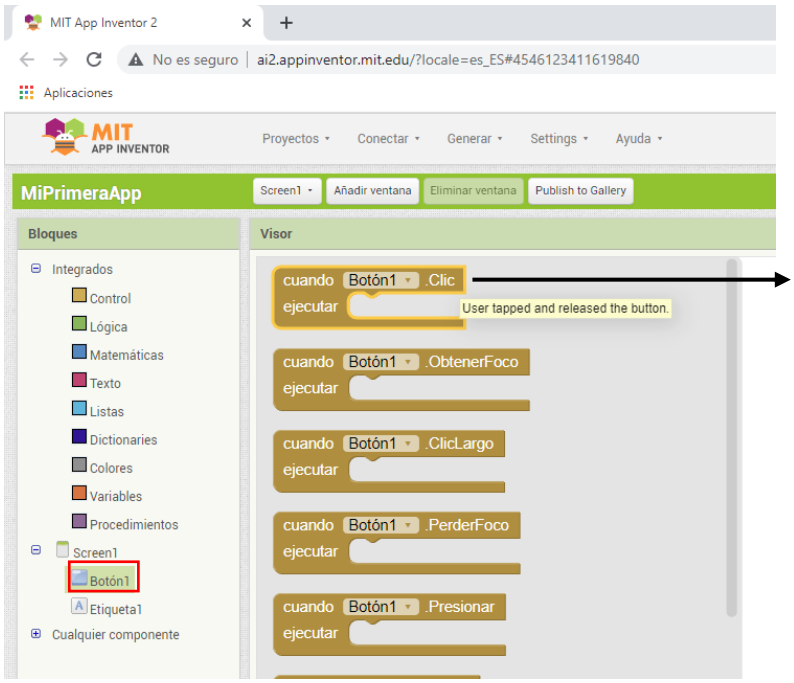
1º DEFINICIÓN DE LA INTERFAZ. En primer lugar deberemos de definir la interfaz de nuestra aplicación, seleccionando y colocando los elementos que queremos que aparezcan en la pantalla de nuestra App.

- Necesitamos un componente **Botón**, para que al pulsarlo muestre el texto “Hola mundo” por pantalla.
- Necesitamos un componente **Etiqueta**, para mostrar el texto “Hola mundo”

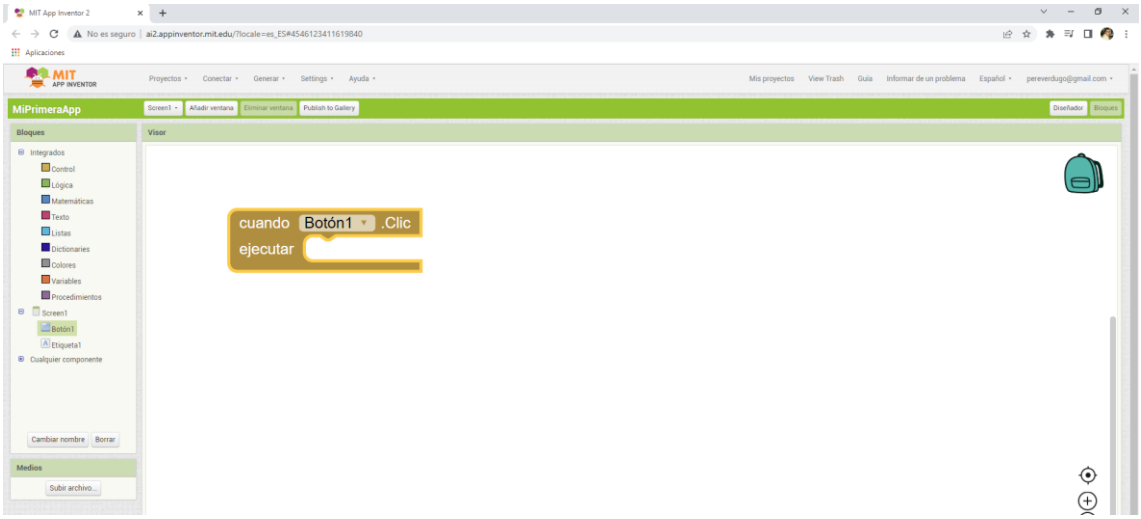


2º PROGRAMACIÓN POR BLOQUES. Ahora realizamos la programación de los componentes que hemos dispuesto en la pantalla de nuestra aplicación para indicar cómo se debe comportar cada uno de ellos.



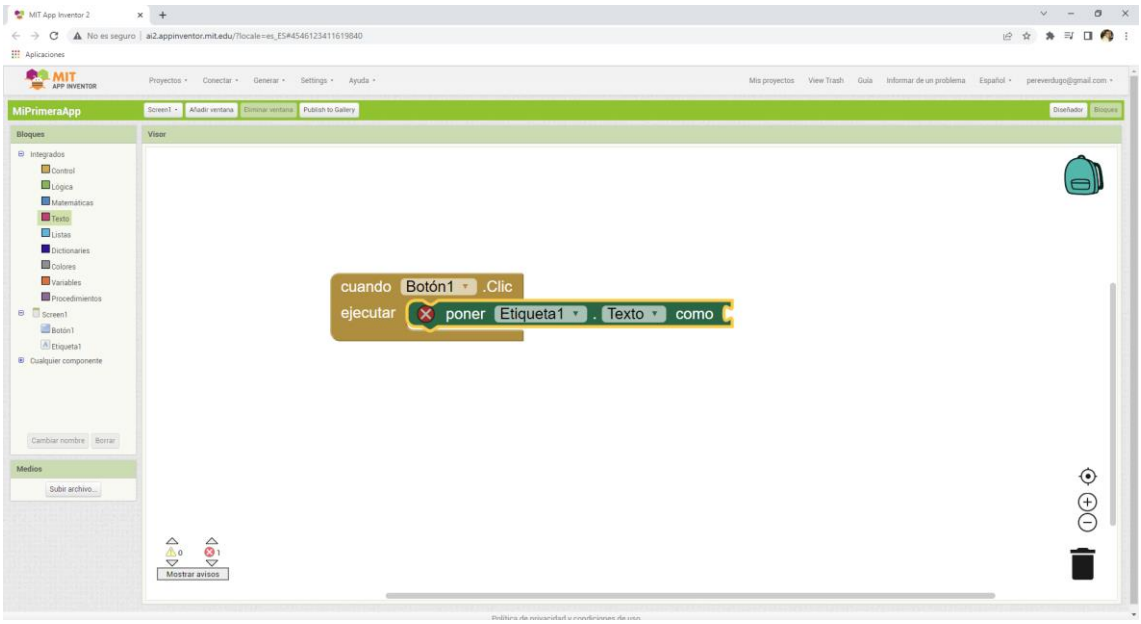


Seleccionamos el componente botón y seleccionamos el bloque Cuando Botón1 Click y lo arrastramos a la ventana de programación.

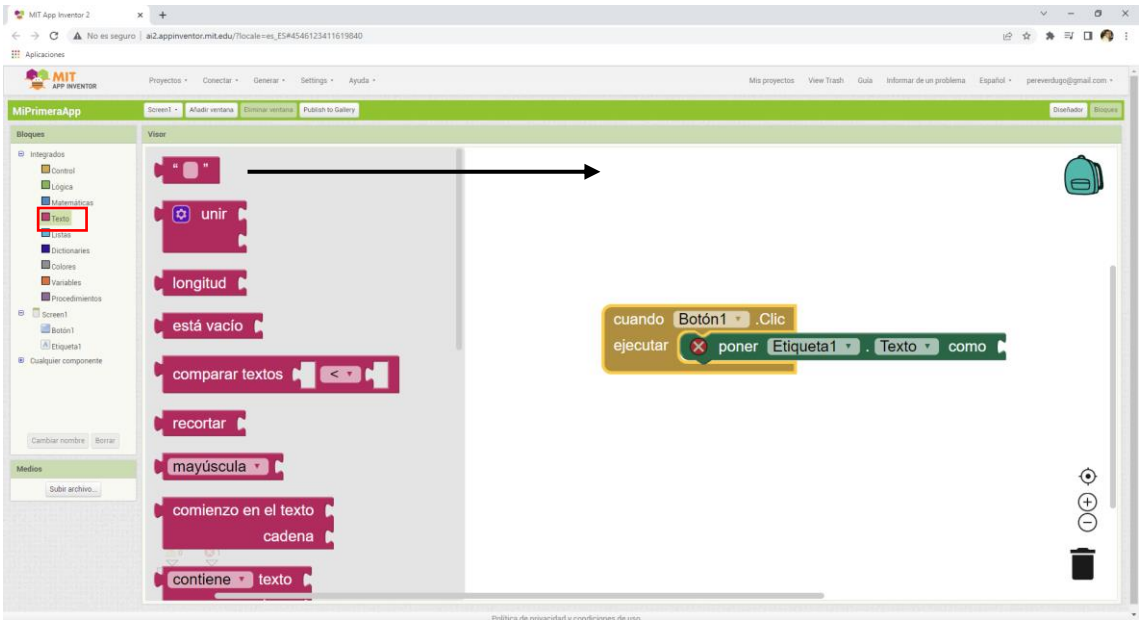


A continuación seleccionamos el componente etiqueta.

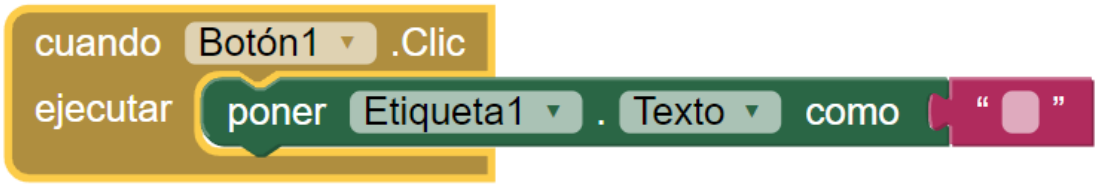
Arrastramos el bloque poner Etiqueta1.Texto como.



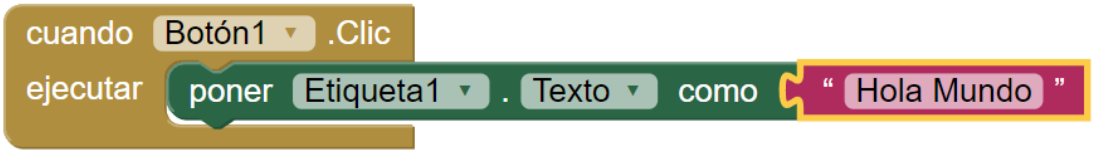
Ahora seleccionaremos el grupo de texto.



Arrastramos el primer bloque que tendremos que encajar con el bloque poner Etiqueta1.



En el último bloque escribiremos Hola Mundo.



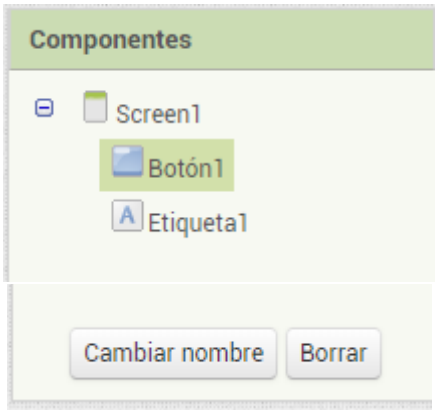
Nombrado significativo de Objetos

Es muy importante que para tener un mejor conocimiento y control sobre el desarrollo de nuestras aplicaciones nombremos de forma significativa a los diferentes componentes que vamos a utilizar en nuestra aplicación para luego poderlos identificar de una forma más rápida y clara.

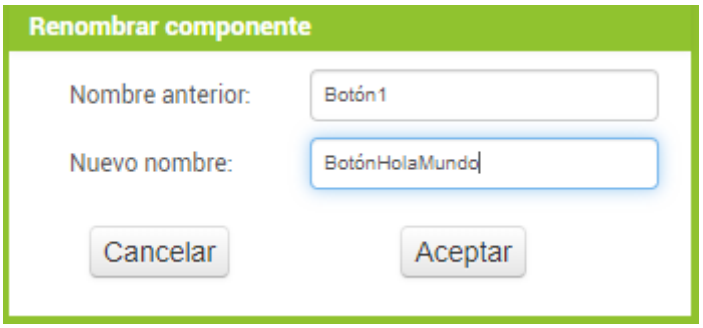
De esta forma nuestra primera App podría quedar de la siguiente manera:

Nos vamos de nuevo a modo de diseño.

En la ventana del visor seleccionamos el Botón1.



En el apartado de Componentes seleccionaremos el botón Cambiar nombre.

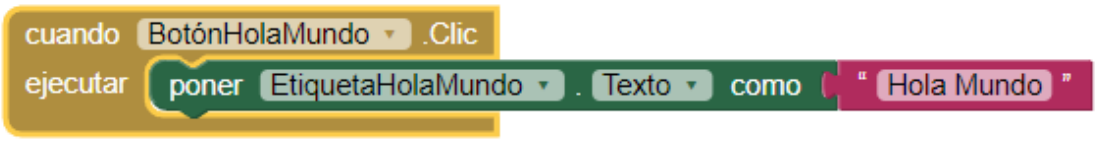


Seguido del botón Aceptar.

Repetiremos lo mismo con la etiqueta.

Seguido del botón Aceptar.

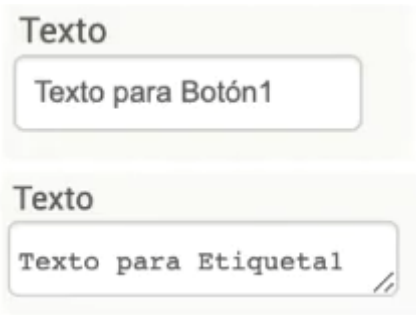
Si ahora vamos a la programación en bloques.



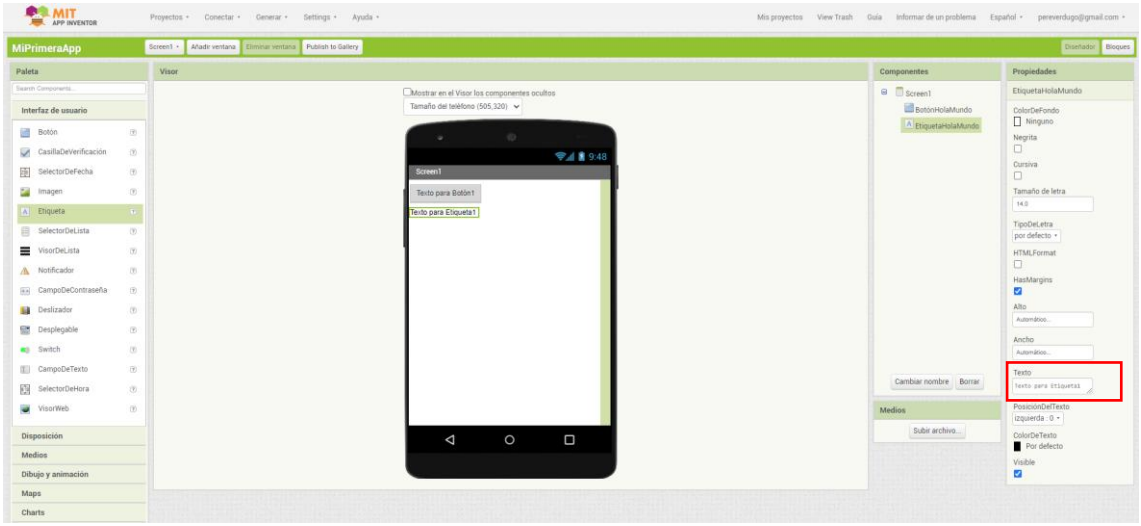
Ha cambiado automáticamente los nombres de los objetos.

DIFERENCIA ENTRE EL NOMBRE DE LOS COMPONENTES Y EL TEXTO QUÉ SE VA A MOSTRAR EN LOS MISMOS

Una cosa en el nombre del componente, que será el que utilicemos en programación de bloques para referirnos a el mismo y otra el texto que queremos que muestre por defecto estos componentes en pantalla cuando se muestren, esto segundo lo modificamos en el campo de texto que aparece en la columna de propiedades.

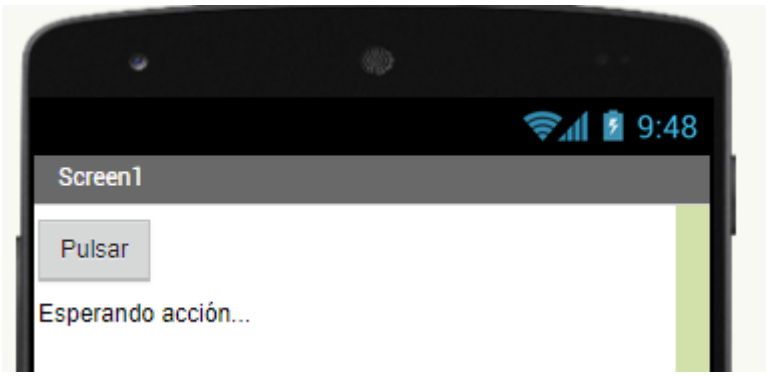


También se podrá modificar lo indicado anteriormente cuando se quiera por programación.



Vamos a cambiar el texto del botón y el texto de la etiqueta que mostrará por defecto.

Finalmente nuestra aplicación quedará con el siguiente aspecto:



Si queremos que la etiqueta de texto no muestre nada lo podemos dejar vacía.

La etiqueta no se muestra pero sigue estado.

Probar nuestra App

Hay 3 posibilidades para realizar pruebas en vivo. Los cambios que realicemos en nuestro programa directamente se actualizarán en nuestras pruebas. Estas son:

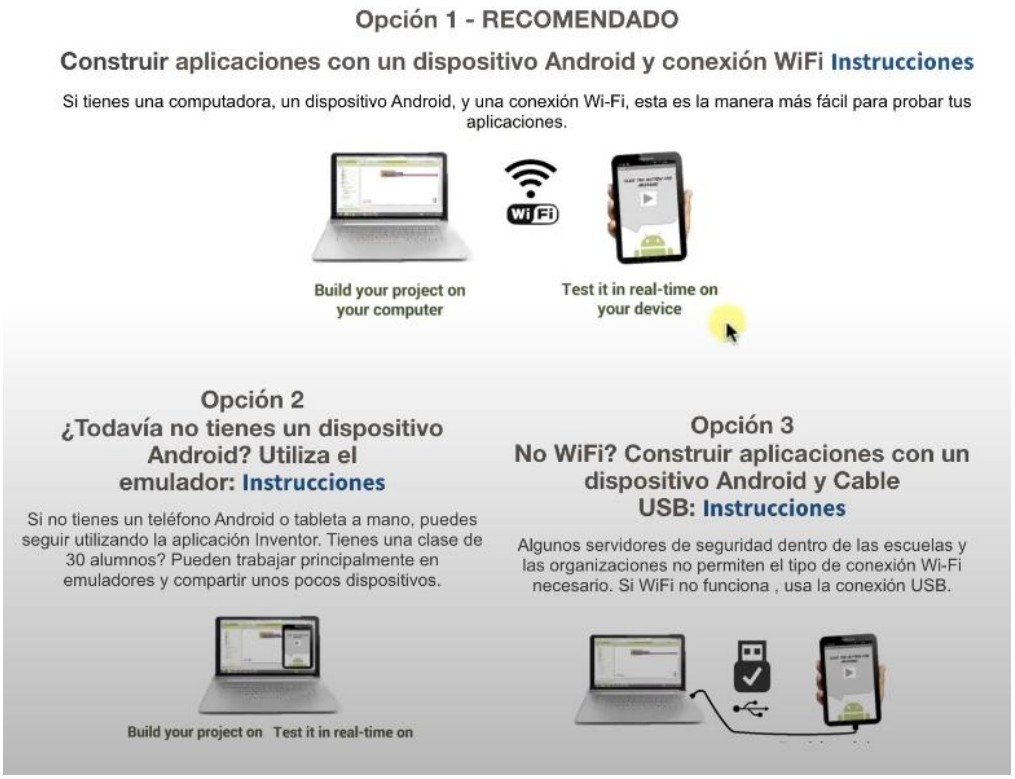
1.- Si estás utilizando un dispositivo Android y tienes una conexión inalámbrica a Internet (WIFI), puedes comenzar la creación de aplicaciones sin descargar ningún software en tu ordenador. Eso sí, tendrás que instalar la aplicación Companion App Inventor en tu dispositivo.

Una vez la tengas instalada aparecerá este icono.

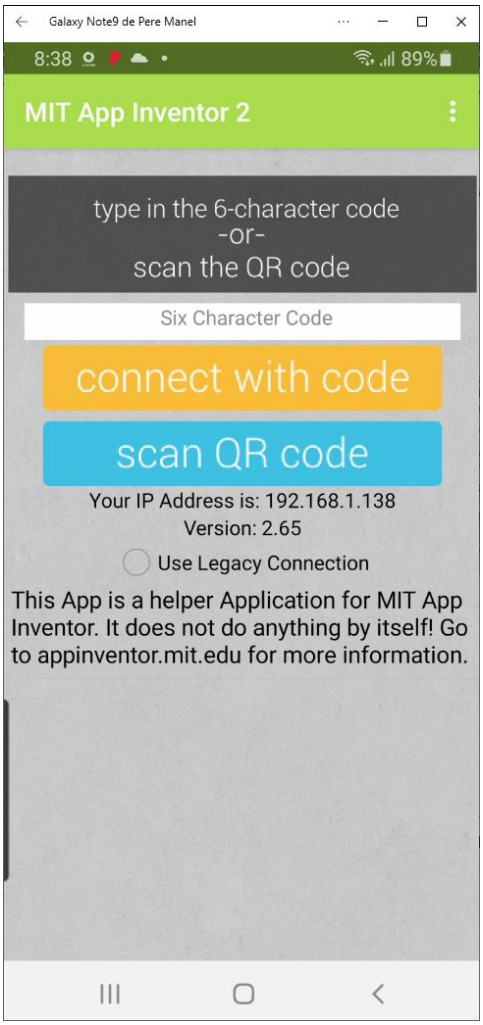
Elige la opción uno. Esta opción se recomienda encarecidamente.

2.- Si no tienes un dispositivo Android, tendrás que instalar el software en tu ordenador para que pueda utilizar el emulador de Android en la pantalla del mismo. Elige la opción dos.

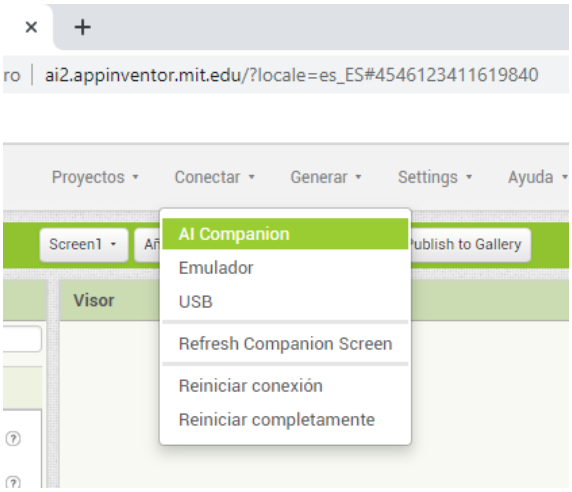
3.- Si no tienes una conexión inalámbrica a Internet (WIFI), tendrás que instalar el software en tu ordenador de modo que puedas conectar a tu dispositivo Android a través de USB. Elige la opción tres. La opción de conexión USB puede ser complicada, especialmente en Windows. Usa esto como un último recurso.



Ahora vamos a ejecutar la aplicación que anteriormente instalamos en nuestro móvil.

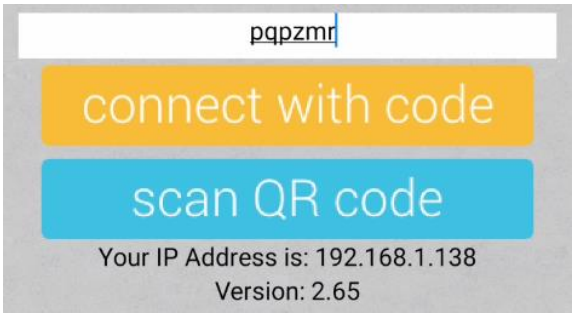


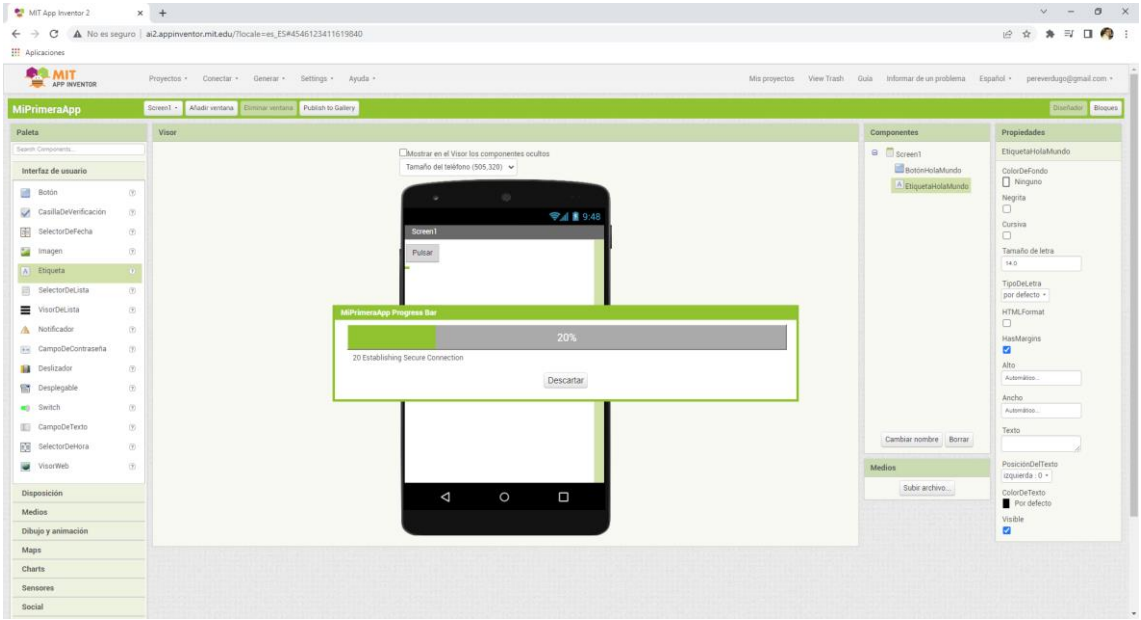
Ahora para poder ver nuestra aplicación en el móvil desde App Inventor.



Del menú Conectar seleccionaremos Al Companion.

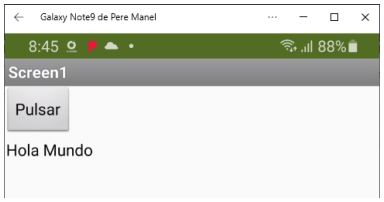
Nos muestra un código y además un código QR, podemos conectar de las dos formas.





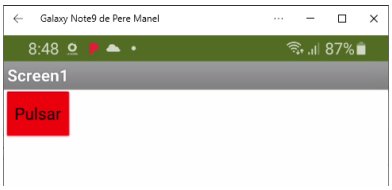
Al finalizar este proceso se podrá probar en nuestro móvil.

Ahora vamos a pulsar el botón.



Ya se muestra el mensaje Hola Mundo.

Puedes realizar cambios y esto se verán automáticamente en el móvil, vamos a cambiar el color del botón.

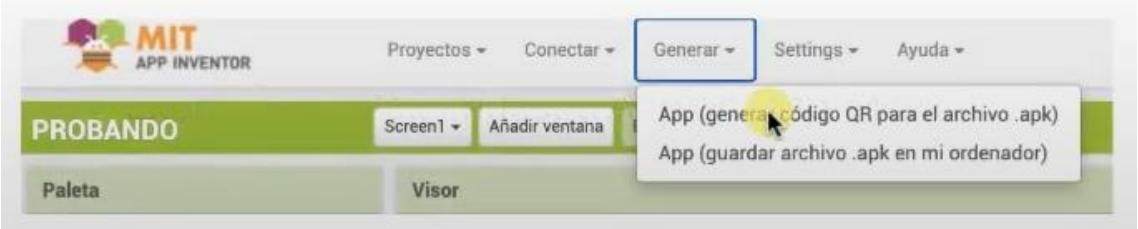


Exportar e instalar nuestra App en Dispositivo móvil

¿QUE ES UN APK?

Un archivo APK es el formato utilizado para la instalación de software en Android. Éste es una variante del JAR de Java, y se utiliza para distribuir e instalar componentes empaquetados a smartphones o tabletas.

¿CÓMO CREAR O GENERAR UN ARCHIVO APK EN APP INVENTOR?



Crear o construir una aplicación a partir de los bloques de código lógicos es muy fácil.

Este archivo lo podemos compartir con nuestros compañeros de clase y amigos, para que lo puedan instalar en su móvil.

Cuando instaléis la aplicación el móvil o tableta nos pedirá permiso para su instalación ya que esta aplicación no la descargamos de la tienda de AppStore, pero como la hemos realizado nosotros la podemos instalar sin ningún problema.

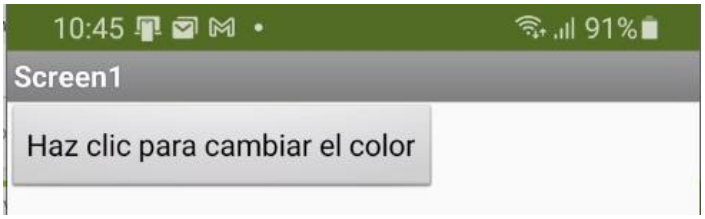
4.-Interfaz de usuario

Elementos típicos interface Usuario

- Botones
- Casilla de verificación
- Selector de Fecha
- Selector de Hora
- Imagen
- Etiquetas para poner textos en la pantalla
- Selector de lista (botón que al pulsar sale una lista para elementos)
- Desplegable
- Visor de listas
- Notificador. Lo veremos más adelante
- Campo de texto
- Campo de contraseña: Es una campo de texto pero oculto (para contraseñas)
- Deslizador. Barra horizontal sobre la cual me puedo desplazar, por ejemplo un volumen.
- Cambiar. Desencadena un evento.
- VisorWeb. Para ver una página web, ojo vaciar el cache de los visores web para que se vaya actualizando la información si está información cambiar rápidamente y queremos ver el refresco automáticamente.

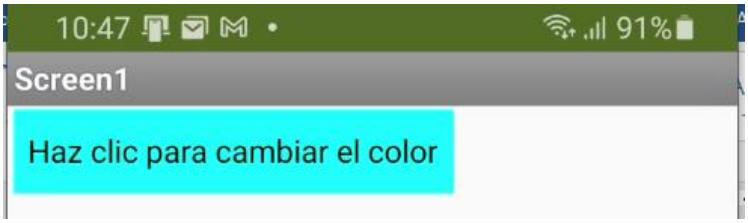
Botón que detecta cuándo se hace clic sobre él. Se puede cambiar su apariencia, y también se puede deshabilitar.

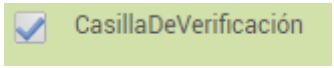
Vamos a agregar un botón, como nombre BotonCambiaColor y como texto haz clic para cambiar el color.



Este será el código por bloques.

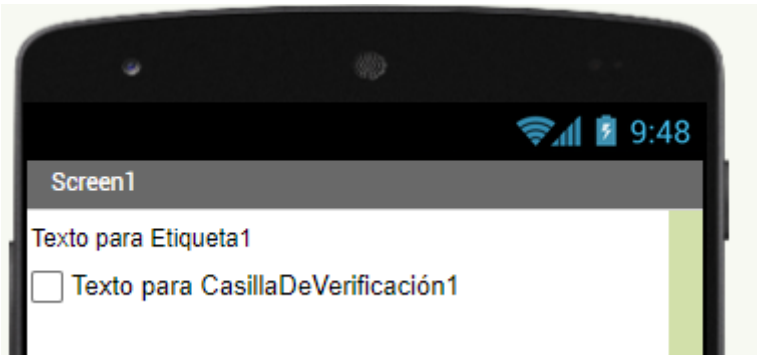
Cuando presionemos el botón el color de fondo será de color azul.



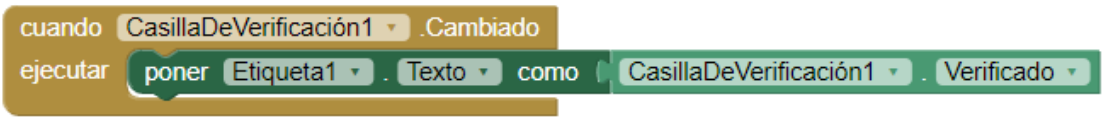


Es una casilla de verificación que activa un evento cuando se hace clic en ella. Existen muchas propiedades para definir su aspecto, que pueden gestionarse desde el Diseñador o el Editor de Bloques.

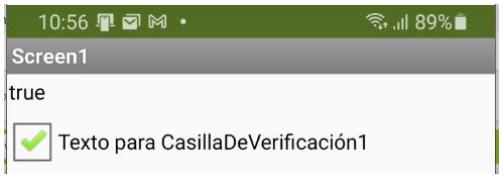
Vamos a agregar un etiqueta de texto en la parte superior .

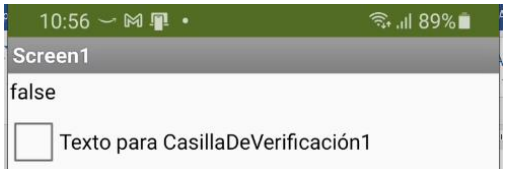


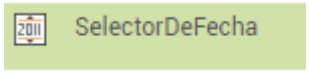
Hemos agregado una etiqueta y en la parte inferior una casilla de verificación, ahora vamos a los bloques.



Cuando activemos la casilla de verificación y luego la desactivemos este será el resultado desde nuestro móvil.

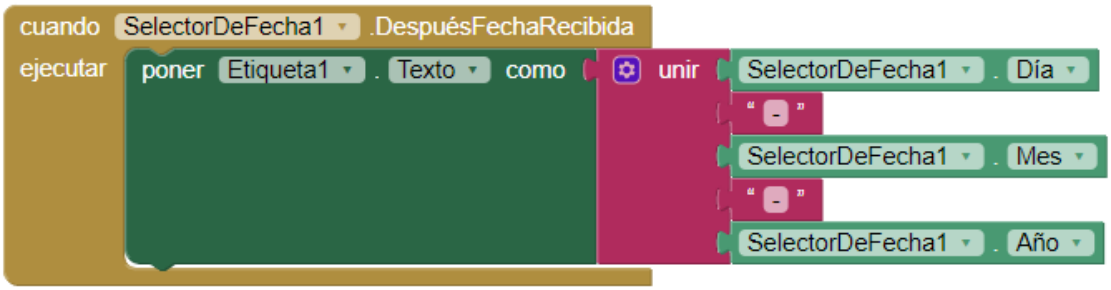






Un botón que, al pulsarlo, abre una ventana de diálogo que permite al usuario seleccionar una fecha.

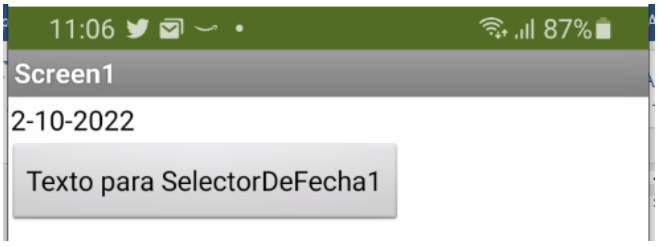
Este será el diseño, ahora vamos a bloques.



Después de fecha recibida ponemos poner Etiqueta1.texto como, en el grupo de texto encontramos el bloque unir que nos concatenará varias cadenas de texto y además le pasamos los valores del día, mes y año separado por guiones.

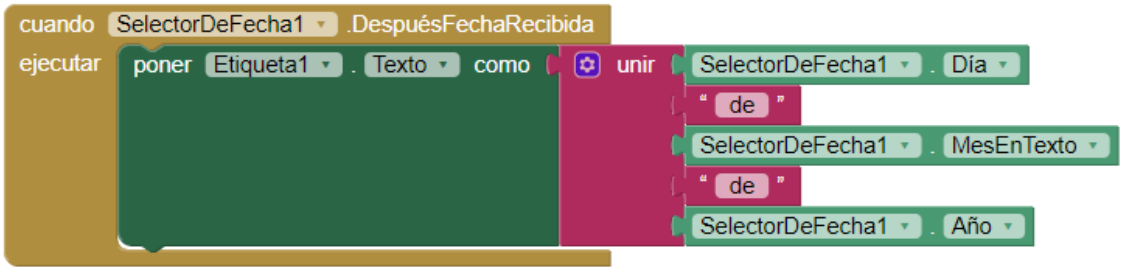
Presionamos el selector de fecha.

Seleccionamos la fecha deseada, seguido del botón Aceptar.



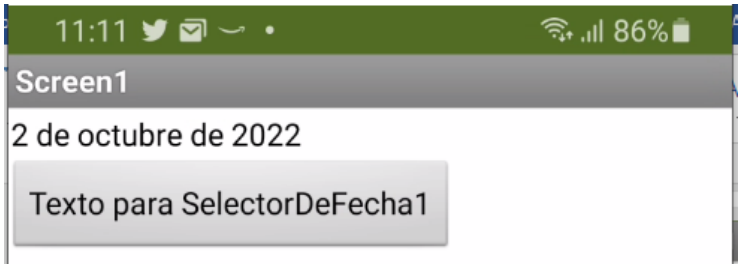
La etiqueta de texto ya muestra la fecha que hemos seleccionado en el SelectorDeFecha1.

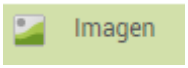
Ahora vamos a realizar los siguiente cambios.



En las cadenas de texto donde está la palabra de hemos puesto un espacio antes y después de la palabra de.

Este será el resultado cuando seleccionemos una fecha.



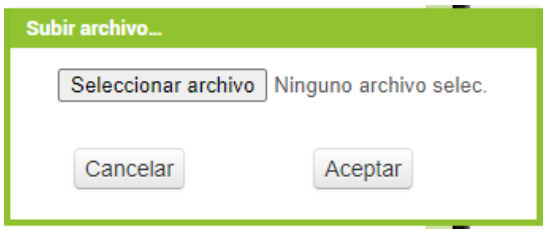


Componente para mostrar fotos. Mediante el Diseñador o el Editor de Bloques se pueden definir tanto la foto que hay que mostrar, como otras características que afectan a su apariencia.

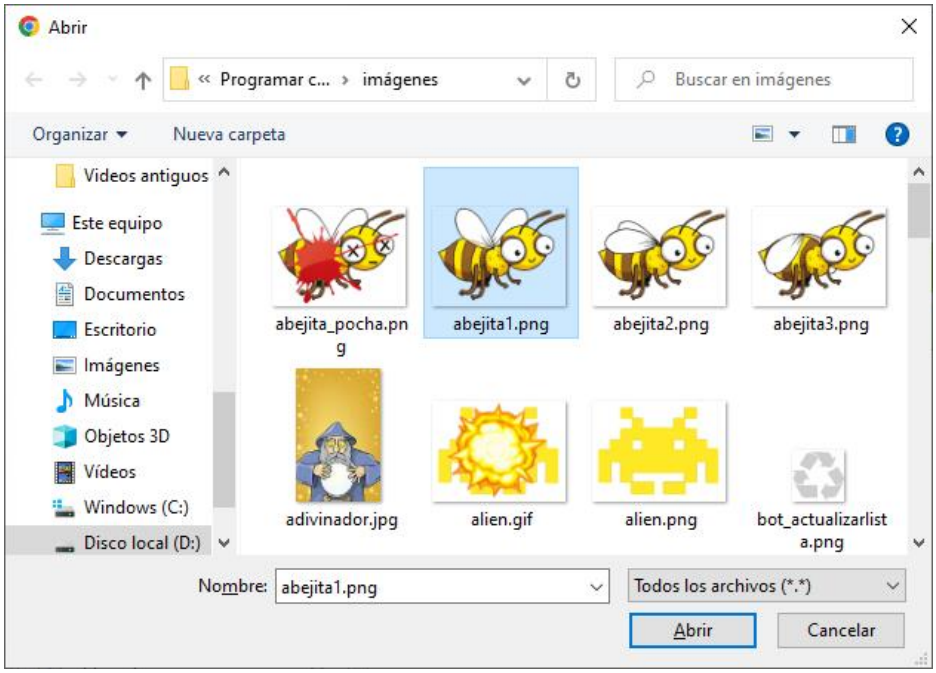
Teniendo la imagen seleccionada en propiedades.

Seleccionamos Foto.

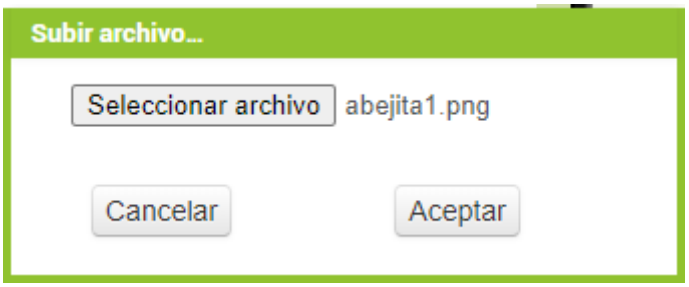
Seccionamos subir archivo.



Seleccionamos Seleccionar archivo.



Seleccionamos la imagen seguido del botón Abrir.

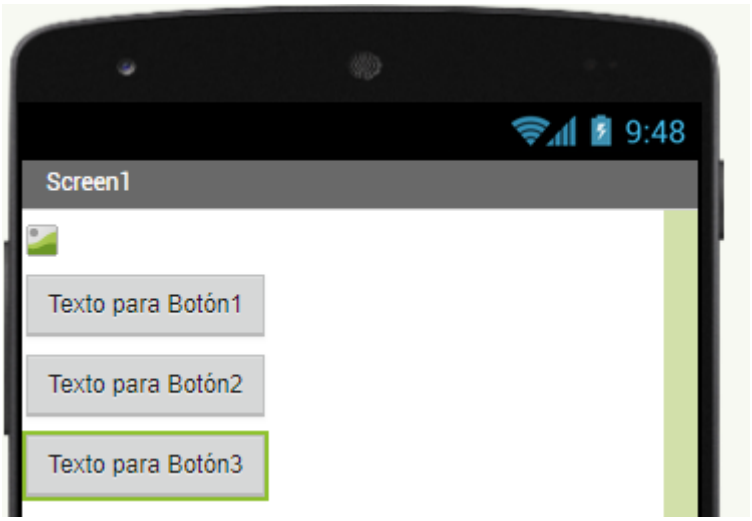


A continuación le damos al botón Aceptar.

Luego en la ventana propiedades podremos cambiar el tamaño del alto y ancho, así como la rotación y poder hacer de esta imagen clickable.

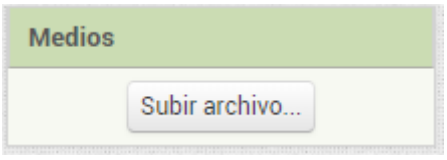
Ahora vamos a realizar la siguiente práctica:

Crearemos el siguiente diseño:



Según el botón que presionemos tiene que mostrar una determinada imagen.

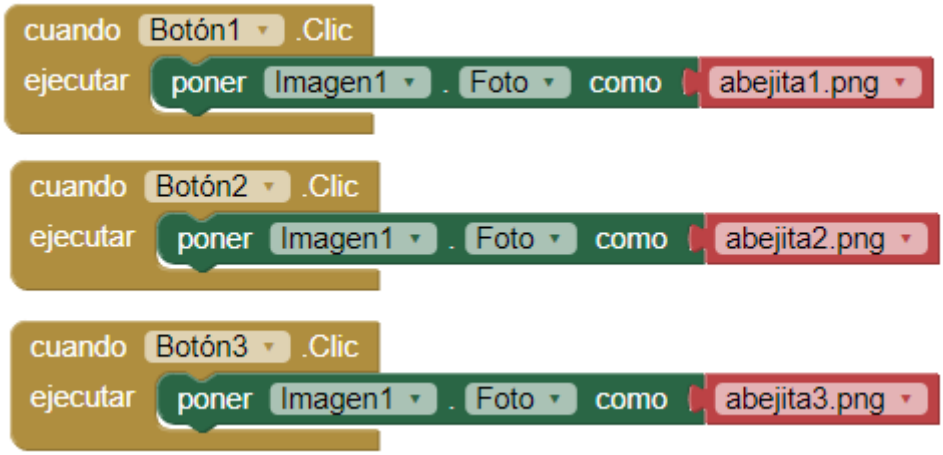
El segundo paso será subir las imágenes.



En la ventana Medios vamos a subir las tres imágenes.

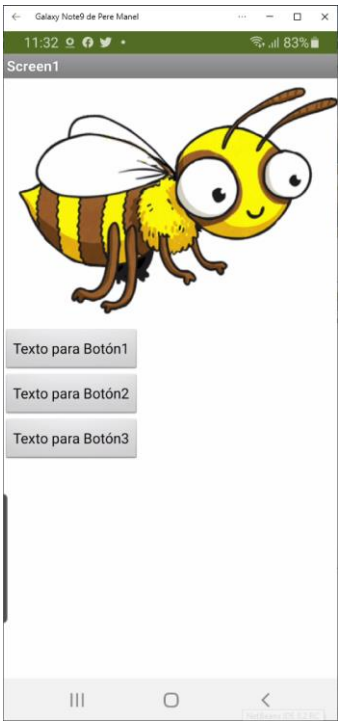
Hemos seleccionado una abejita pero cada imagen tiene una posición distinta de sus alas.

Ahora nos vamos a la parte de bloques.

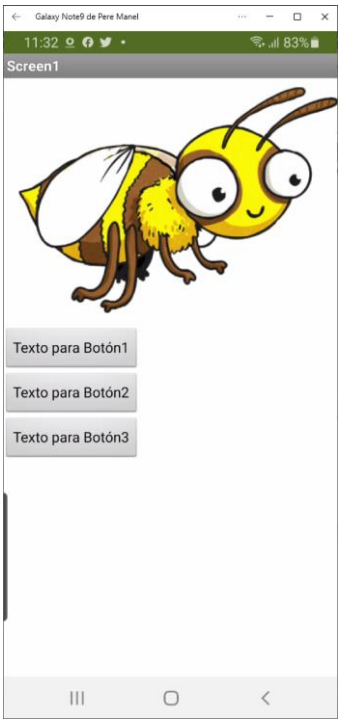


Al hacer clic en cada botón muestra las siguientes imágenes:

Botón1



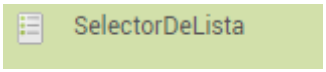
Botón2



Botón3

Una etiqueta muestra un texto, definido en su propiedad `Texto`. Utilizando otras propiedades, que pueden manejarse en el Diseñador o en el Editor de Bloques, es posible modificar la apariencia y la ubicación del texto.

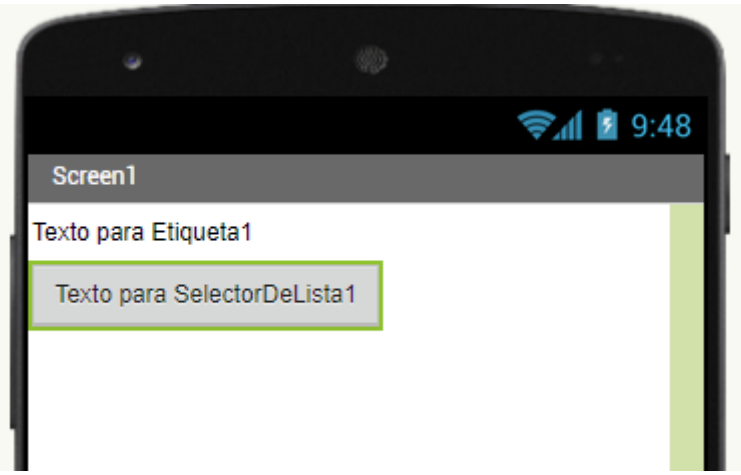
Ya la hemos utilizado en varios ejemplos.



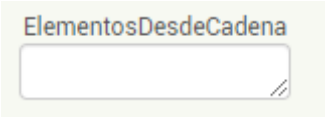
Un botón que, al pulsar sobre él, muestra una lista de textos entre los cuales el usuario puede elegir. Se pueden definir los textos a través del Diseñador o del Editor de Bloques, especificando en la propiedad `ElementosDesdeCadena` la lista de textos delimitados (por ejemplo, *opción 1, opción 2, opción 3*), o en el Editor de Bloques, poniendo en la propiedad `Elementos` el nombre de una Lista.

Indicando el valor `Verdadero` en la propiedad `MostrarBarraDeFiltrado` se permite al usuario buscar en la lista. Hay otras propiedades que pueden afectar a la apariencia del botón (`PosiciónDelTexto`, `ColorDeFondo`, etc.) y que determinan si el botón se puede pulsar.

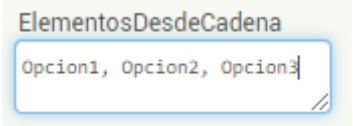
Vamos a realizar el siguiente diseño:



En la ventana de Propiedades nos vamos a elementos.

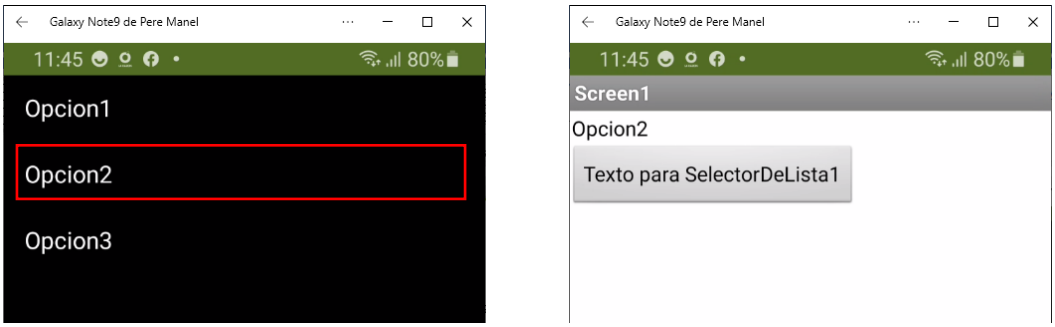


Hemos de escribir sus valores separados por comas.



Vamos a programar con bloques.

Cuando presionamos el SelectorDeLista, muestra los valores de la lista y una vez seleccionado en la etiqueta1 muestra el valor que hemos seleccionado.



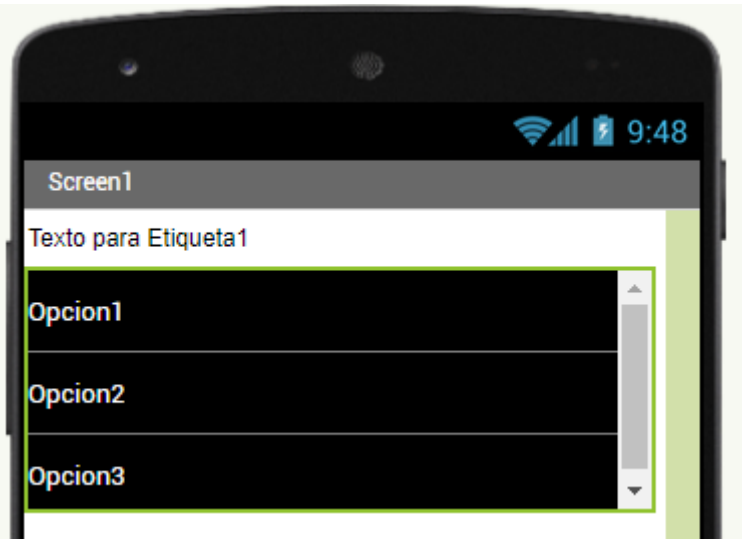
Hemos seleccionado la Opción2.

Es un componente visible que permite definir una lista de textos para que se muestren en la pantalla. La lista se puede definir utilizando la propiedad `ElementosDesdeCadena`, o desde el Editor de Bloques, por medio del bloque `Elementos`.

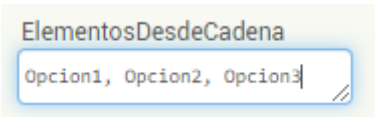
Atención: Este componente no funciona correctamente en pantallas enrollables.

Es muy parecido al SelectorDeLista.

Vamos a realizar el siguiente diseño.

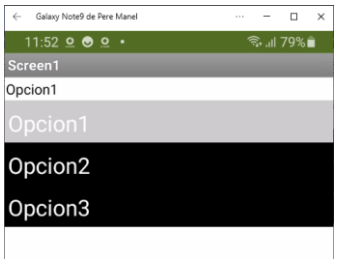


Agregamos los elementos.



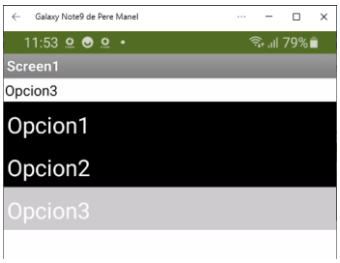
Ahora vamos al bloque de programación.

Este será el resultado en nuestro móvil.

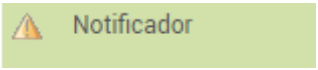


Seleccionamos Opcion1

Seleccionamos Opcion2



Seleccionamos Opcion3



El componente Notificador muestra cuadros con alertas, mensajes y alertas temporales, y hace anotaciones en el registro de Android utilizando los siguientes métodos:

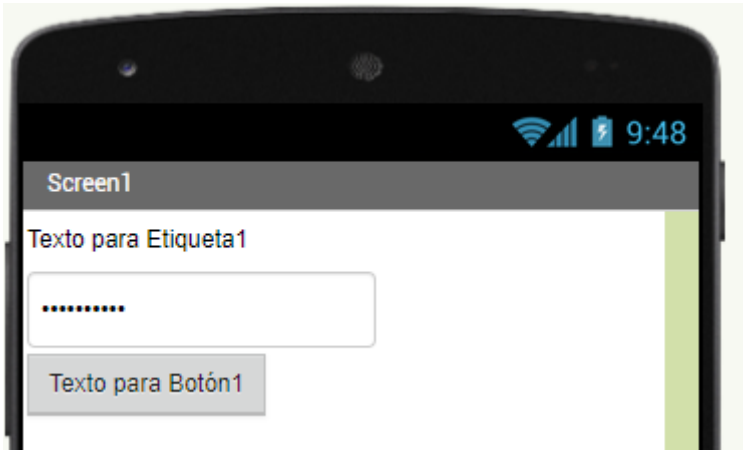
- MostrarDiálogoMensaje: presenta un mensaje que el usuario puede descartar pulsando un botón.
- MostrarDiálogoElección: muestra un mensaje y dos botones, para que el usuario pueda elegir entre dos respuestas, por ejemplo si o no, después de lo cual si activa el evento DespuésDeSelección.
- MostrarDiálogoTexto: permite al usuario escribir una respuesta al mensaje, después de lo cual se activa el evento DespuésDeEntradaDeTexto.
- MostrarAlerta: presenta una alerta que desaparece después de un breve periodo de tiempo.
- RegistrarError: anota un mensaje de error en el registro de Android.
- RegistrarInfo: escribe un mensaje de información en el registro de Android.
- RegistrarAviso: anota un mensaje de aviso o advertencia en el registro de Android.
- Se puede poner formato a los mensajes que aparecen en los cuadros de diálogo (pero no a las alertas) usando las siguientes etiquetas HTML: <b>, <big>, <blockquote>,
, <cite>, <dfn>, , , <small>, , <sub>, <sup>, <tt>. <u>
- También se puede utilizar la etiqueta font para definir, por ejemplo, el color con <font color="blue">. Entre los nombres de colores disponibles están aqua, black, blue, fuchsia, green, grey, lime, maroon, navy, olive, purple, red, silver, teal, white, y yellow

Este componente lo veremos en próximos capítulos.

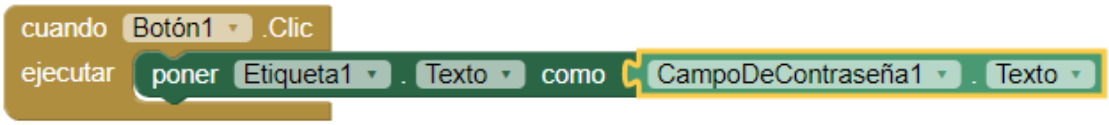
Es un cuadro para escribir contraseñas. Es como un componente `CampoDeTexto` normal, pero no se muestran los caracteres mientras el usuario los está escribiendo.

El texto que escribe el usuario queda almacenado en la propiedad `Texto`. Si está vacío, se puede utilizar la propiedad `Pista`, que aparece como texto atenuado, para ofrecer al usuario un ayuda sobre lo que debe escribir.

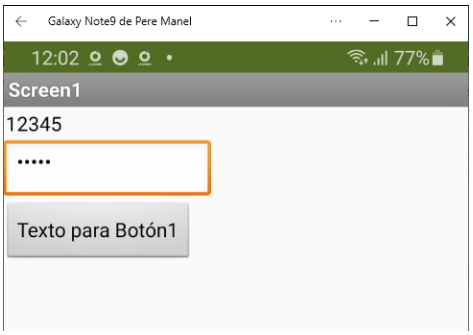
Los campos de texto se utilizan normalmente junto a componentes `Botón`, de tal manera que el usuario pulsa el botón una vez que ha terminado de introducir el texto.

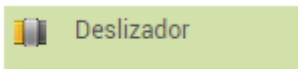


Realizamos este diseño y esta será la programación en bloques.



Cuando introduzcamos una contraseña y presionemos el botón este será el resultado:

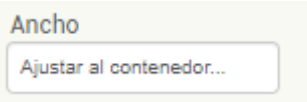




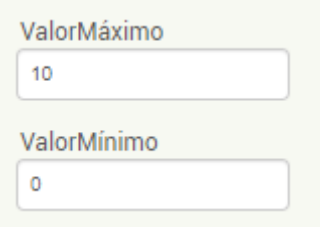
Un Deslizador es una barra de progreso con un marcador que puede desplazarse. Al tirar del marcador y arrastralo a la izquierda o a la derecha éste cambia de posición. Según se vaya arrastrando irá activándose el evento PosiciónCambiada, para indicar cuál es la nueva posición del marcador dentro del componente Deslizador. Se puede usar la posición del Deslizador en cualquier momento para cambiar dinámicamente otro atributo de un componente, como el tamaño de letra de un CampoDeTexto o el radio de una Pelota.

Este será el diseño:

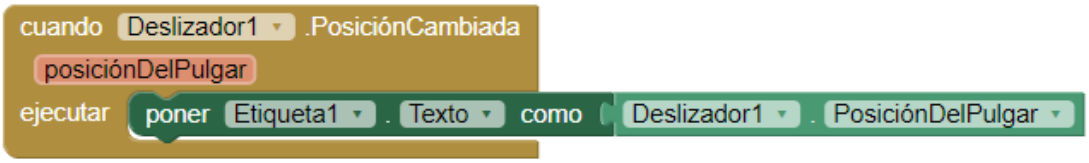
Vamos a modificar las siguientes propiedades.



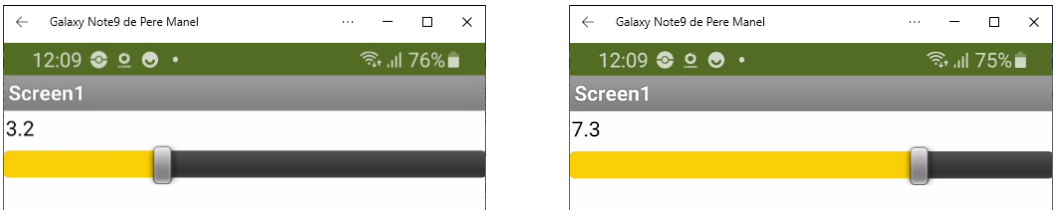
Valores mínimo y máximo.

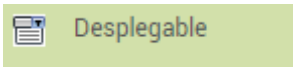


Ahora vamos a agregar el bloque de programación para que en la etiqueta se vaya viendo su valor.



Vamos a ver en nuestro móvil moviendo el deslizador.

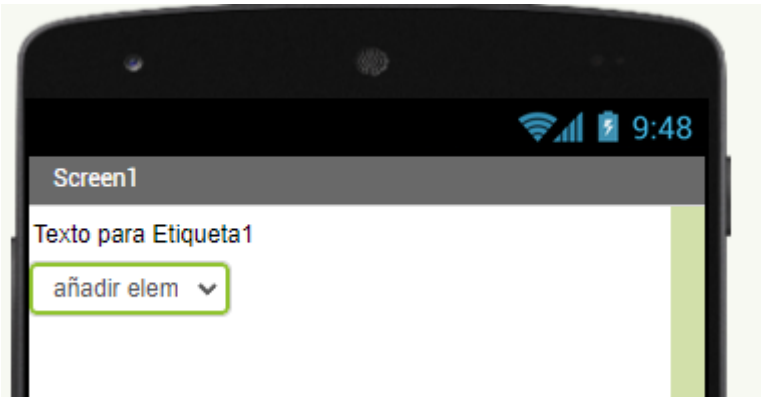




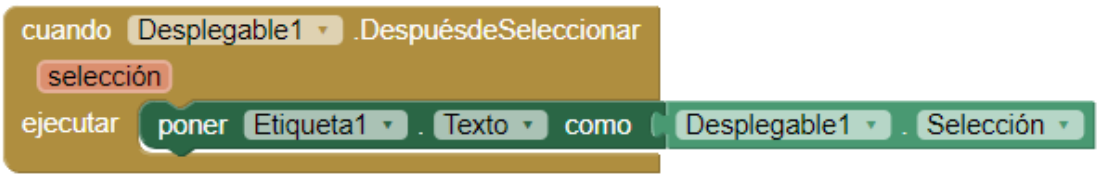
Es un componente desplegable que muestra una ventana emergente que contiene una lista de elementos. Dichos elementos se definen en el Diseñador o en el Editor de Bloques poniendo en la propiedad `ElementosDesdeCadena` una sucesión de elementos separados por comas (por ejemplo, *opción 1, opción 2, opción 3*), o en el Editor de Bloques, indicando en la propiedad `Elementos` el nombre de una Lista.

Son muy parecidos a SelectorDeLista y VisorDeLisata

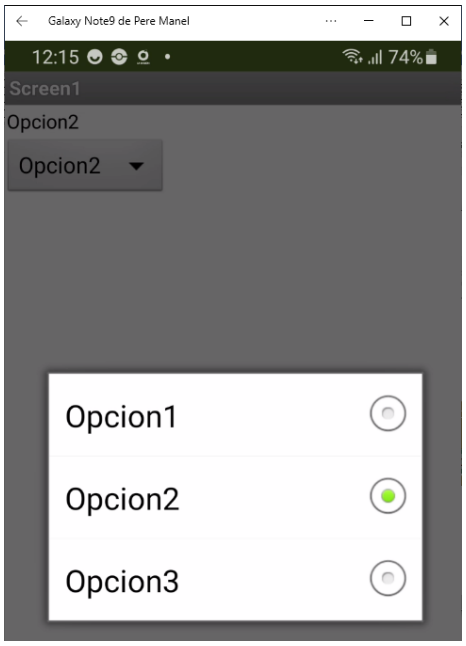
Como ya sabes como van los anteriores realiza el siguiente diseño:

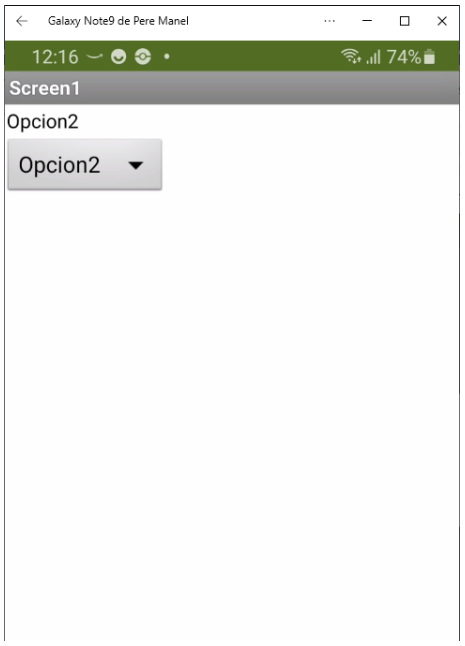


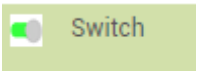
Agregamos el siguiente bloque de programación:



Este será el resultado desde nuestro móvil.





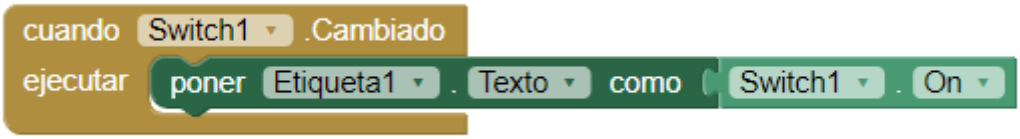


Interruptor de palanca que genera un evento cuando el usuario hace clic en él. Hay muchas propiedades que afectan su apariencia que se pueden configurar en el Diseñador o el Editor de bloques.

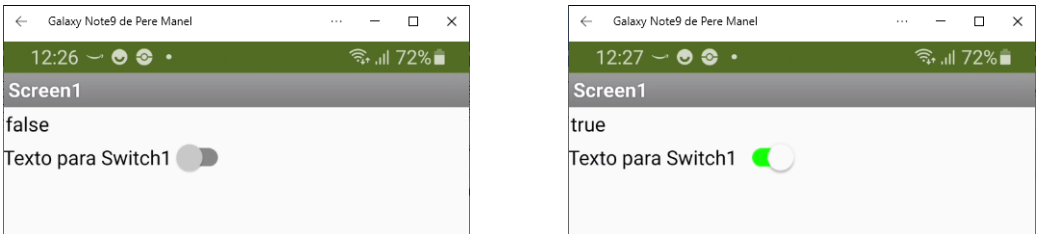
Es muy similar a CasillaDeVerificación.

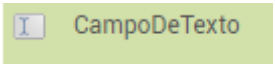
Vamos a realizar el siguiente diseño:

Ahora vamos a programar el siguiente bloque:



Ahora veremos el resultado en nuestro móvil cuando está activo y desactivado.





Es un campo que permite al usuario introducir texto. En la propiedad `Texto` se almacena el texto que ha introducido el usuario. Si el campo está vacío, la propiedad `Pista`, que aparece como un texto atenuado dentro del campo de texto, puede proporcionar alguna pista al usuario sobre lo que tiene que escribir.

La propiedad `Multilínea` determina si el texto debe estar compuesto por más de una línea. En el caso de un campo de texto de una sola línea, el teclado se ocultará automáticamente cuando el usuario pulse la tecla Hecho. Cuando se trate de un campo Multilínea, la aplicación tendrá que utilizar el método OcultarTeclado, o confiar en que el usuario pulse la tecla Atrás para ocultarlo.

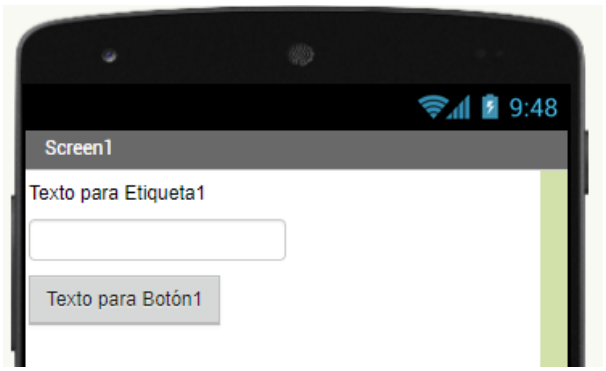
La propiedad `SóloNúmeros` obliga a que únicamente puedan introducirse números.

Otras propiedades afectan a la apariencia de un campo de texto (`PosiciónDelTexto`, `ColorDeFondo`, etc.), y determinan si el campo es editable o no (`Habilitado`).

Es habitual utilizar los campos de texto junto a componentes `Botón`, de tal manera que el usuario pulsa el botón cuando ha terminado de introducir el texto en el campo.

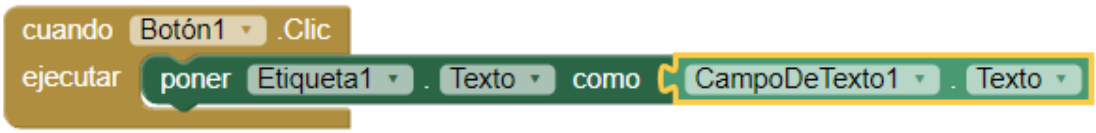
Cuando no se quiere que sea visible el texto que está escribiendo el usuario, es más apropiado utilizar el componente `CampoDeContraseña`.

Vamos a realizar el siguiente diseño:

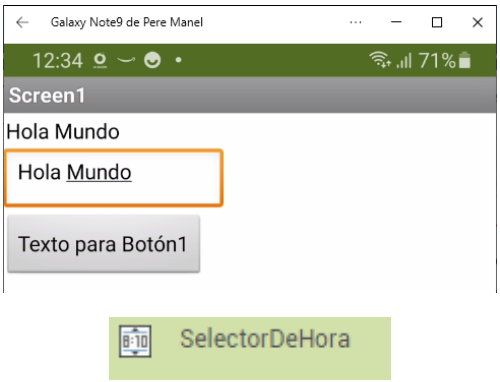


Vamos a escribir un texto y cuando presionemos el botón se tienen que mostrar en la etiqueta.

Vamos a programar el bloque.



Este será el resultado escribiendo Hola Mundo.

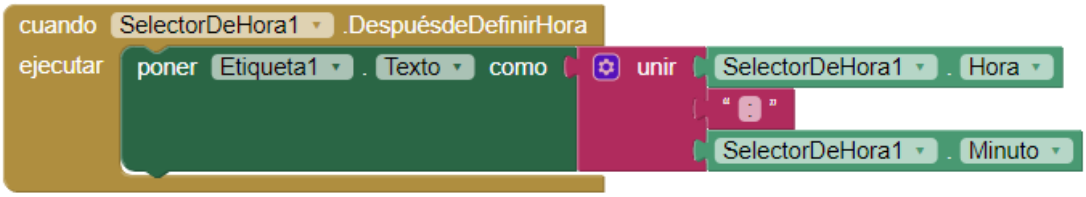


Es un botón que cuando se pulsa muestra una ventana que permite al usuario definir una hora.

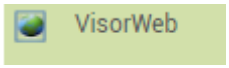
El funcionamiento es muy similar al SelectorDeFecha.

Vamos a realizar el siguiente diseño:

Vamos a programar los siguientes bloques:



Cuando ejecutemos la aplicación este será el resultado desde nuestro móvil.



Es un componente para ver páginas Web. Se puede especificar la página de inicio desde el Diseñador, o en el Editor de Bloques. Se puede definir el visor para seguir los enlaces al tocar en ellos, y para que los usuarios puedan rellenar formularios. Atención: No es un navegador con funcionalidad completa. Por ejemplo, al pulsar la tecla Atrás del teléfono se saldrá de la aplicación, es decir, no nos llevará a la página anterior.

Vamos a realizar una aplicación para poder navegar por diferentes ventanas.

Este será el diseño:

Queremos que al introducir una dirección url en CampoDeTexto y presionar el botón nos muestra la página web.

Vamos a programar los bloques:

Este será el resultado desde mi móvil.

5.-Disposición elementos en pantalla

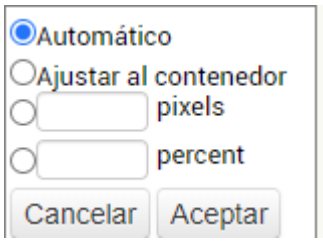
Disposición y tamaños Elementos Pantalla

Creando la interfaz de usuario. Para crear la interfaz de usuario, el desarrollador puede hacer uso de distintos elementos, desde botones y cuadros de texto, a elementos que modifican la manera en que los elementos visuales se colocan en pantalla.

Para incluir elementos en App Inventor solo tendremos que buscar el que queremos incluir en la aplicación y arrastrarlo en la pantalla Visor, pero hay cierto elemento que nos facilita crear una interfaz mejor, con los elementos Disposición.

TAMAÑO DE ELEMENTOS

Podemos indicar el tamaño de los elementos de diversas formas:



Así podemos modificar el alto y ancho de nuestros elementos.

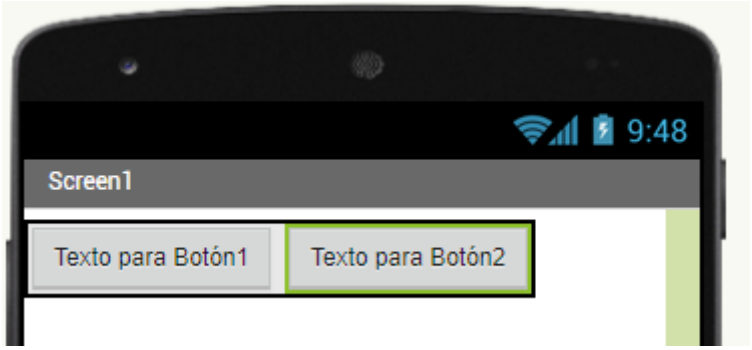
- Automático: es el tamaño que deja por defecto.
- Ajustar al contenedor: Pondrá el tamaño máximo, ajustado el elemento donde esté insertado.
- Pixels: Podemos dar las dimensiones en pixeles, esto tiene un inconveniente y es que cuando lo instalemos a distintos móviles con distintas resoluciones el resultado optimo no será el mismo en todos los móviles.
- Porcentaje: Si trabajamos con porcentaje es resultado será el mismo en los móviles aunque tengan distintas resoluciones.

Nota: Lo aconsejable es trabajar en porcentaje para que se pueda ajustar a todos los móviles.

Disposición Horizontal

Ordena los elementos colocados uno después de otro en orientación horizontal.

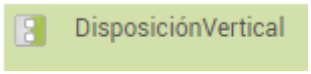
Para ello tenemos que arrastras una DisposicionHorizontal y dentro de esta arrastraremos los elementos que sean necesario, en este ejemplo agregamos dos botones.



En la ventana de componentes observamos la siguiente estructura:

Ten mucho cuidad a la hora de borrar disposición, ya que con ello vamos a eliminar todos los elementos que hay dentro de esta.

Disposición Vertical



Un elemento de formato en el que se pueden colocar otros elementos que deben aparecer uno debajo de otro (el primero estará arriba, el siguiente debajo, etc). Si deseas colocar elementos en horizontal, el elemento DisposiciónHorizontal.

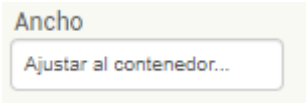
Disposición tabular

Ordena los elementos en forma de tabla.

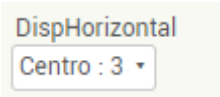
En propiedades de la DisposiciónTabular podemos modificar las columnas y los registros.

Disposiciones dentro de disposiciones

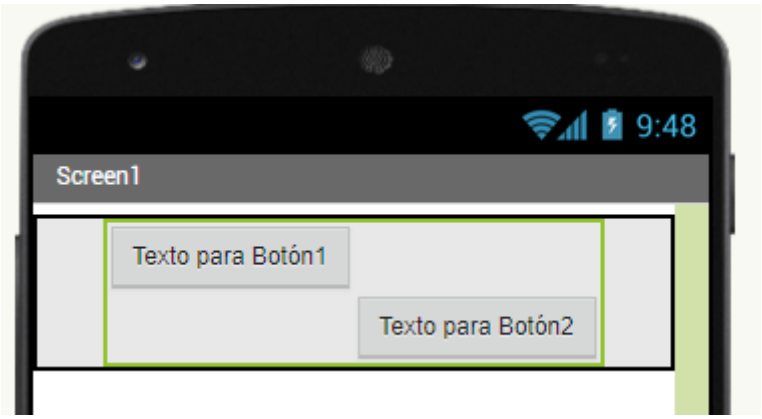
Como podrás observar con la DisposiciónTabular no podemos centrar los botones, para esto lo que tenemos que hace es primero insertar una DisposicionHorizontal, que ocupe todo el acho de la pantalla.



Alinearlo al centro.

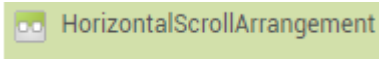


Dentro agregaremos una DisposicionTabular y podremos conseguir el siguiente resultado.



Esta será la estructura:

Scroll Horizontal y Vertical



En el caso de que agreguemos más componentes de lo que soporta el ancho o alto de nuestra pantalla de este modo todos los elementos se podrán mover de izquierda a derecha con el HorizontalScrollArrangement y de arriba abajo con el VerticalScrollArrangement.

Si queremos que todas la pantalla tenga un scroll vertical seleccionaremos el componente Screen1 y en la ventana de propiedades activaremos Enrollable.

Esto es útil se en el visor no accedemos a los componentes que se encuentra en la parte inferior, ya que la resolución del visor es menor a algunos móviles del mercado.

Cuando finalices vuelve a desactivar la casilla Enrollable.

A partir de ahora vas a mejorar el diseño de tu App para que los elementos los puedas situar en el lugar más acertado.

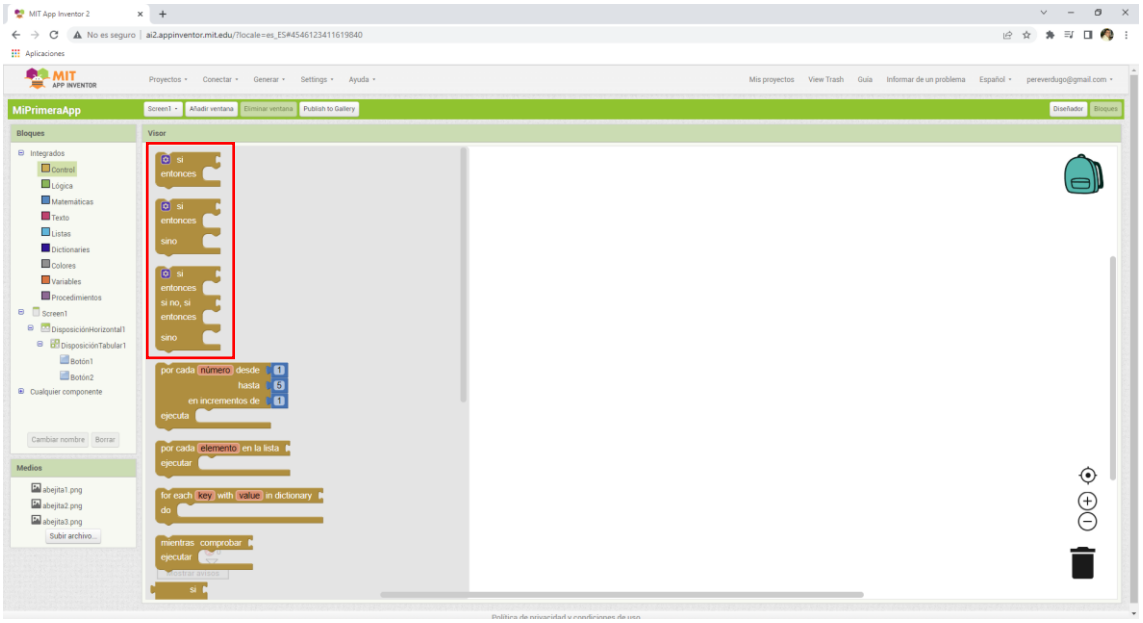
6.- Programación Bloques Básicos

Condiciones

Los condicionales hacen parte importante de la aplicación en App Inventor, pues hace parte de la lógica de nuestra App.

¿Cómo se crea un condicional en App Inventor?

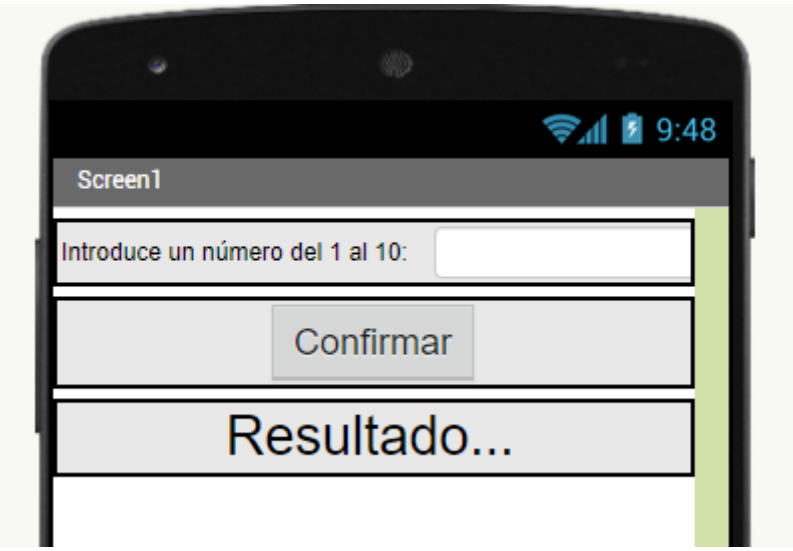
En la sección de bloques, control, podemos ver los condicionales if, else y else if, para añadir la opción del else tenemos que pinchar en la rueda de configuración e incorporar esta nueva opción.



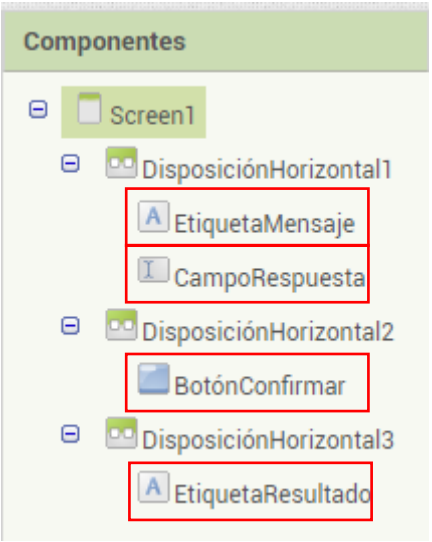
Si la condición se cumple es un True y si no se cumple es un False.

Vamos a crear una aplicación donde nos va a preguntar un número del 1 al 10, el programa dará por bueno el valor 5 en cambio al resto los dará por malo.

Vamos al diseño:

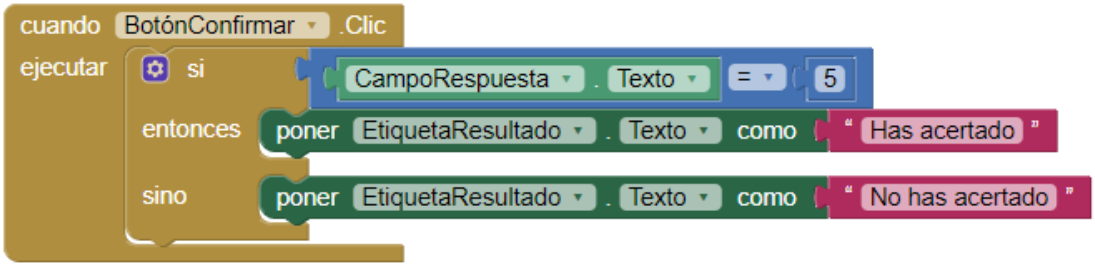


Esta es la estructura:



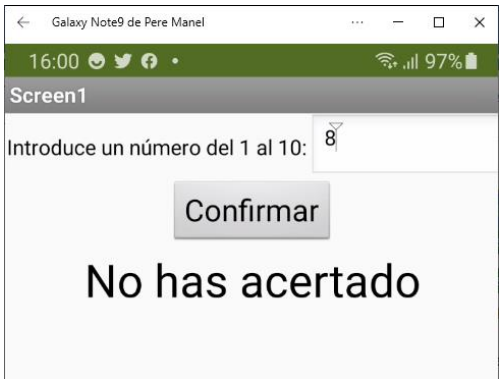
Así podrás cambiar el nombre a los elementos.

Ahora vamos al bloque de programación:

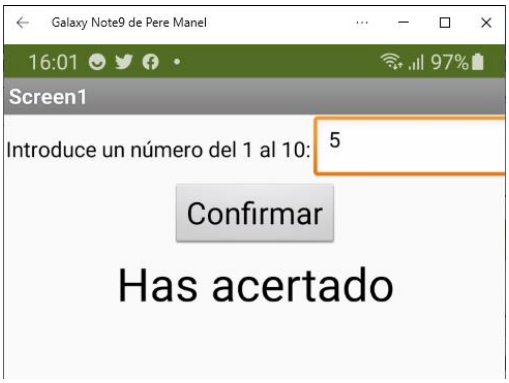


Cuando hacemos clic en el BotónConfirmar hacemos una comparación, si CampoRespuesta.texto es igual a 5 muestra el mensaje “Has acertado” de lo contrario el mensaje “No has acertado”.

Este será el resultado desde el móvil.



Respondiendo con el valor 8



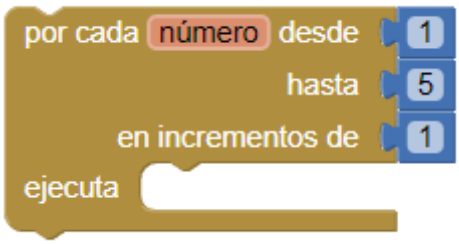
Respondiendo con el valor 5.

Con la rueda de configuración podemos modificar la estructura condicional.

Repeticiones

Ejecuta los bloques que engloban para cada valor numérico en la gama de partir del valor desde y termina hasta, incremento el número por el valor determinado.

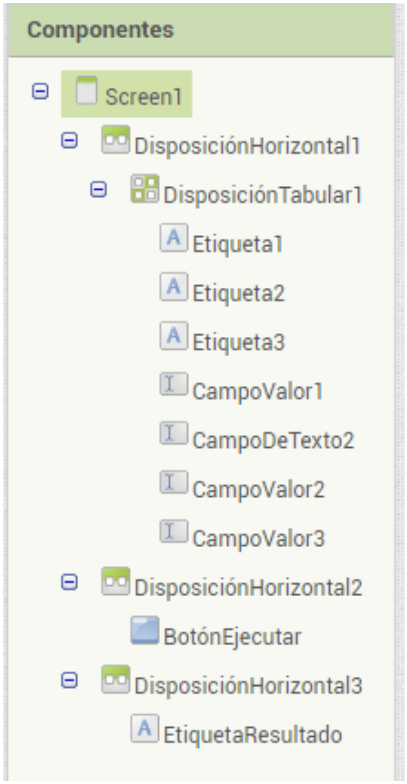
Se puede cambiar el nombre de la variable numero por otro si se desea.



Vamos a realizar un ejercicio donde nos preguntará por el valor inicial, el valor final y el incremento, cuando pulsemos en un botón tiene que imprimir todos los valores.

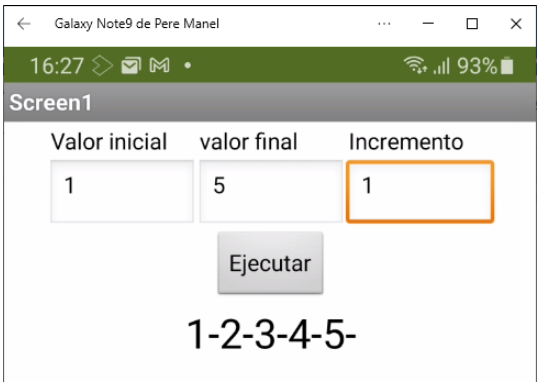
Este es el diseño:

Esta es la estructura:



Ahora vamos al bloque de programación.

Este será el resultado desde nuestro móvil.

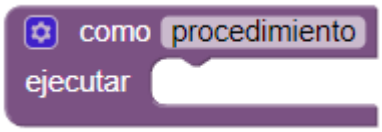


Mientras comprobar… ejecutar

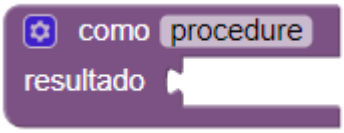
Comprueba el valor “comprobar”. De ser cierto, lleva a cabo la acción indicada en -ejecutar, luego prueba de nuevo. Cuando ya no se cumple la condición se pasa al siguiente bloque.

Mientras se cumpla la condición este será un bucle, cuando deja de cumplir la condición termina el bucle.

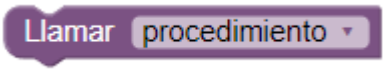
Crear Procedimientos



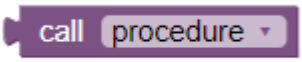
Como procedimiento ejecutar…
Recoge una secuencia de bloques juntos en un grupo. A continuación, se puede utilizar la secuencia de bloques establecida mediante una llamada al procedimiento.
Si el procedimiento tiene argumentos, especifique los argumentos con el botón mutador del bloque. Si haces clic en el icono más azul, puedes arrastrar argumentos adicionales en el procedimiento.



Como procedimiento resultado…
Igual que en el anterior bloque, pero al llamar a este procedimiento se devuelve un resultado.



Llamar procedimiento
Llama a un procedimiento que no devuelve valor.



Llamar procedimiento
Hacer una llamada a un procedimiento que devuelve un valor.

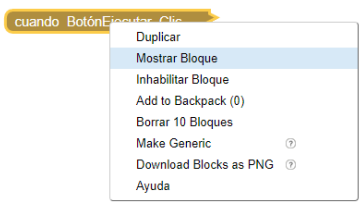
El objetivo de los procedimiento es para poder ejecutar las mismas instrucciones sin tener que repetir los mismos bloques.

Ocultar / Mostrar Bloques Instrucciones

Puede llegar un momento que al tener muchos bloques en la ventana de programación se haga más complicado el seguir programando, una forma de ocultar bloques consiste en hacerle doble clic en él.

Este se contrae, si hacemos de nuevo doble clic este volverá a expandirse.

También se puede realizar con el botón derecho del ratón.



7.- Datos: Variables, Listas, Textos

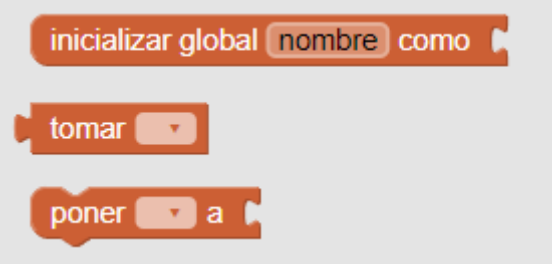
Variables

Las variables pueden entenderse en un lenguaje de programación como almacenes para guardar datos de manera temporal.

En App Inventor existen variables de dos tipos: **globales y locales**. La elección entre el uso de un tipo de variable y otro depende de la aplicación en particular.

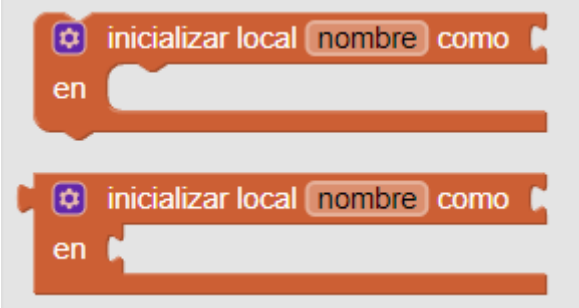
VARIABLES GLOBALES

Las variables globales son aquellas que están disponibles para acceder a su contenido desde cualquier lugar de la aplicación del móvil desarrollada. Por tanto, el valor de una variable global puede ser empleado por cualquier componente o bloque. Las variables globales se crean los siguientes bloques:

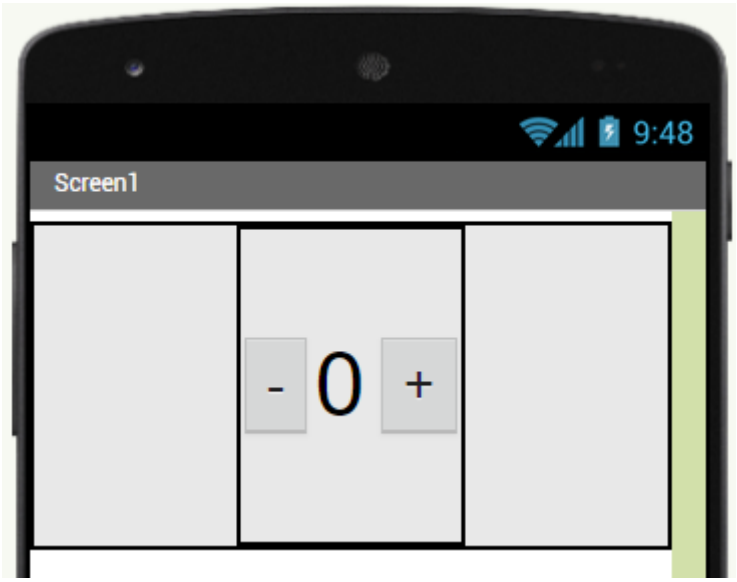


VARIABLES LOCALES

Las variables locales son aquellas que están declaradas dentro del ámbito de un bloque particular. También pueden ser parámetros pasados a función definida por el usuario. El acceso a variables locales solo se pueden realizar desde el bloque en que se define o desde la función que la recibe como parámetro.

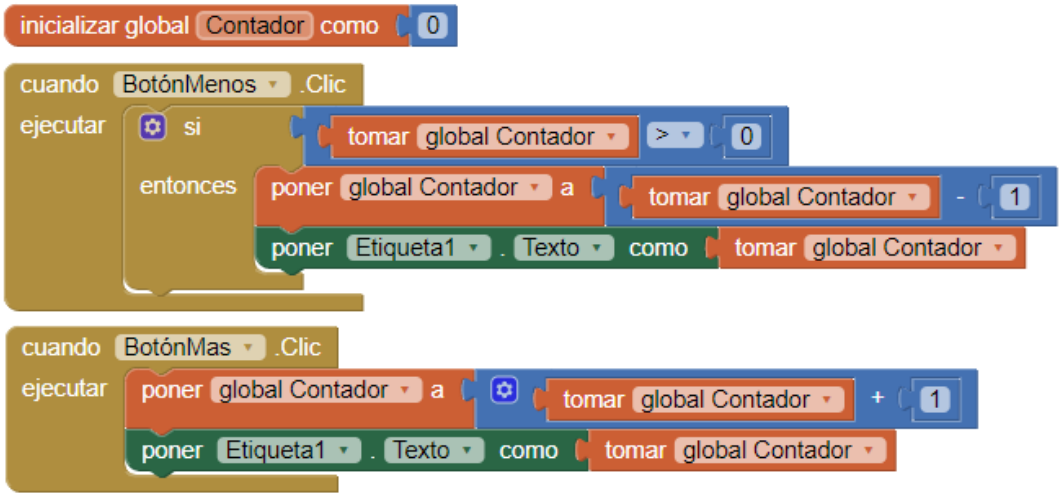


Vamos a realizar una práctica con dos botones uno para incrementar de uno en uno y el otro para decrementar de uno en uno. Podemos controlar las personas que entran y salen de una que tiene limitado el aforo.

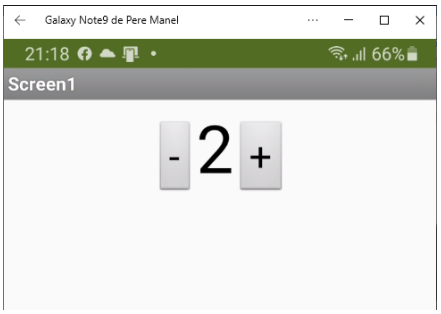


La estructura:

Este será el código por bloques:



Este será el resultado en el móvil.



Con el botón menos decrementa 1 y con el botón más se incrementa 1.

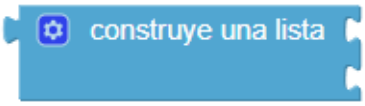
Hemos puesto una condición de que si el contador llega a 0 no reste más.

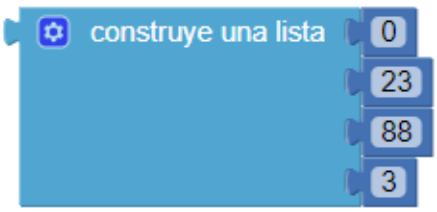
Lista de Elementos

Hay tanta información en el mundo, que puede volverse muy confuso. Por suerte, tenemos manera de organizar la información para así encontrarla y usarla fácilmente. ¿Puedes pensar en algún ejemplo de cómo tú organizas la información y tus objetos en la vida? Aquí hay algunos ejemplos de cosas que quizás haces para organizar cosas:

- Poner el número telefónico de un amigo en una lista de contactos
- Poner lápices en un lapicero
- Poner ropa en un closet

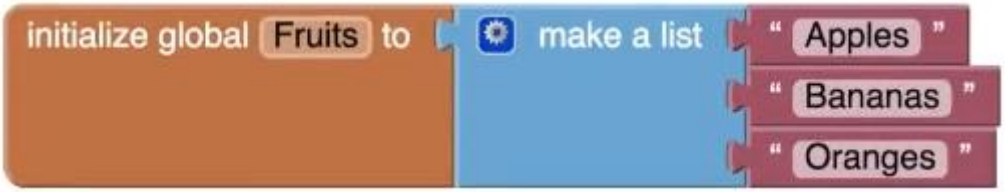
Los programadores usan listas para organizar la información en sus programas. Las listas pueden albergar múltiples porciones de datos y es muy fácil obtener datos de las mismas. Es muy probable que ya hayas hecho alguna lista antes, como las que uno hace antes de ir a la verdulería. Las listas en programación son muy similares. En App Inventor, puedes ir a Bloques y seleccionar “listas”. Las listas lucen más o menos así:





Ejemplo:

Vamos a inicializar la variable Fruit con una lista de tres frutas.

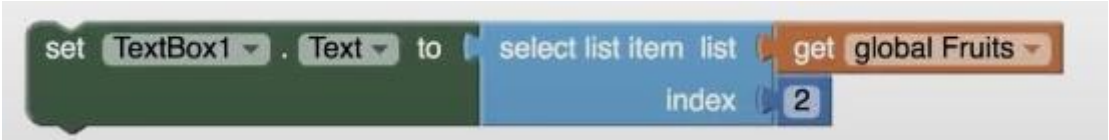


Así es como tu aplicación ve esta información.

Nombre de la lista: Fruits

- Apples (Manzanas) (Index = 1)
- Bananas (Index = 2)
- Oranges (Naranjas) (Index = 3)

Puedes obtener una cosa de una lista en ver de usarla completamente. Si quisieras agregar la cadena “bananas” en un cuadro de texto en vez de toda la lista, deberás decirle a la aplicación que busque el index 2 en tu lista. Así es cómo ubicarías la palabra “bananas” en un cuadro de texto usando App Inventor.

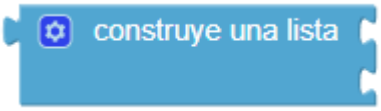


También puedes agregar, remover o reemplazar cosas en la lista. Digamos que olvidaste agregar “kiwis” y “grapes (uvas)” a tu lista de frutas. Podrías agregarlas de este manera:

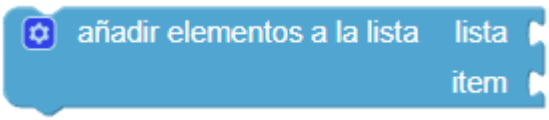
Ahora tu aplicación verá tu lista de la siguiente manera:

- Apples (Manzanas) (Index = 1)
- Bananas (Index = 2)
- Oranges (Naranjas) (Index = 3)
- Kiwis (Index = 4)
- Grapes (Uvas) (Index = 5)

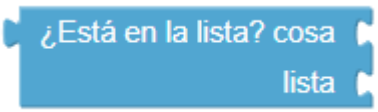
Crea una lista vacía



Crea una lista con un número de elementos.



Añade elementos al final de la lista.

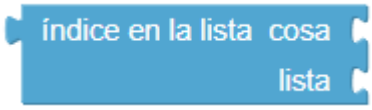


Devuelve cierto si la cosa es un elemento de la lista y falso si no lo es.

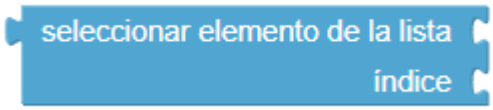
Contar el número de elementos que hay en la lista.

Devuelve cierto si la lista está vacía.

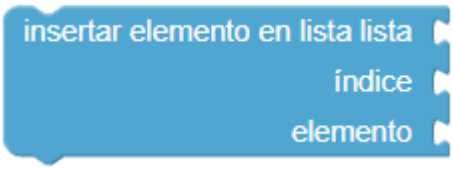
Toma de la lista un elemento al azar.



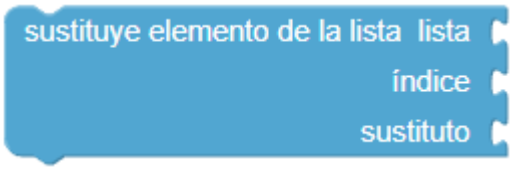
Encuentra la posición de algo en concreto en la lista. Si no está en la lista devuelve 0.



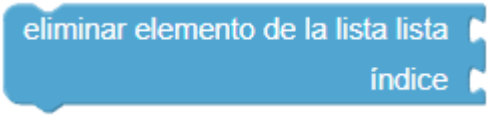
Devuelve el elemento de la lista ubicado en la posición indicada.



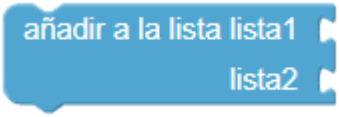
Inserta un elemento en una lista en una posición específica.



Sustituye el elemento n de una lista.



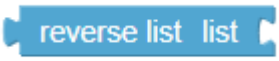
Elimina de la lista el elemento que ocupa una posición específica.



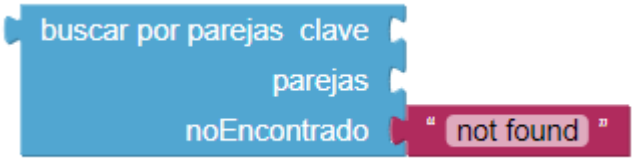
Añade al final de la lista1 todos los elementos que hay en la lista2. Una vez añadidos, lista1 contendrá todos los elementos y lista2 permanecerá inalterada.

Hace una copia de una lista, incluyendo una copia de toda las sablistas.

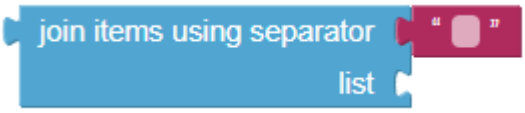
Comprueba si algo está en la lista.



Invierte el orden de la lista de entrada y devuelve como una nueva lista.



Devuelve el valor asociado con la clave en la lista de parejas.

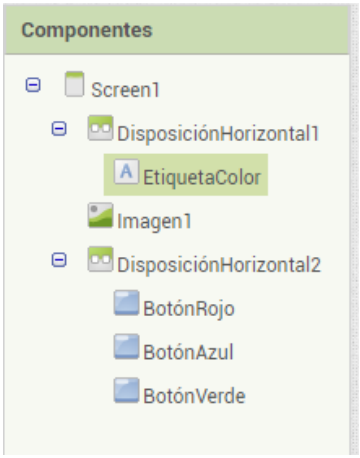


Devuelve texto con elementos de la lista unidos con separador.

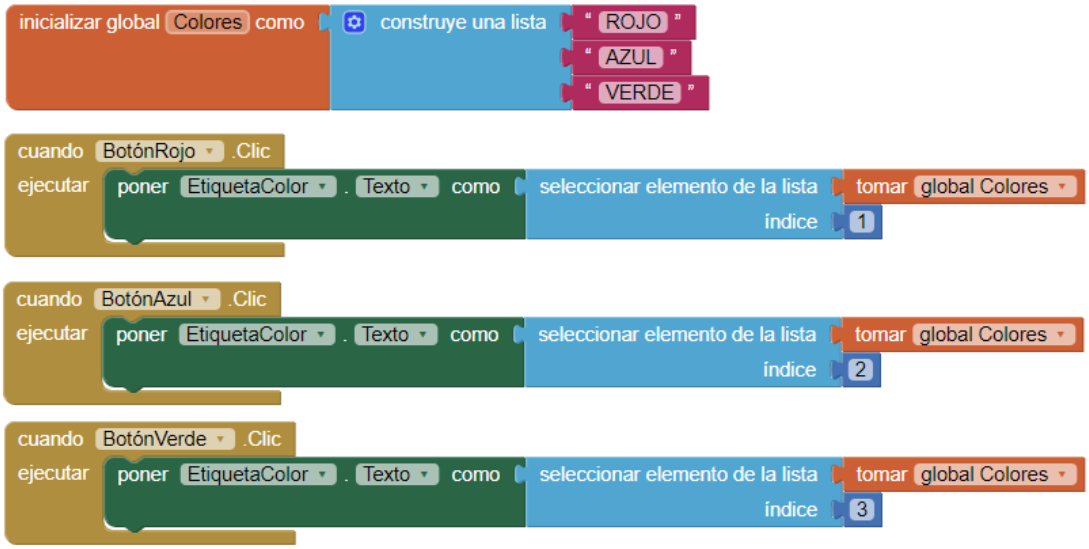
Como ejemplo vamos a crear una lista con tres colores y además tres botones cada uno de ellos tiene que seleccionar un valor de la lista.

Este será el diseño.

Esta es su estructura:



Vamos a programar los bloques:



Este será el resultado en mí móvil:

Presionado botón rojo

Presionado botón azul

Presionado botón verde

Instrucciones de Texto

"…" Contiene una cadena de texto.
Esta cadena puede contener caracteres (letras, números u otros caracteres especiales). En App Inventor se considerará un objeto de texto.

Unir
Anexa todas las entradas para hacer una sola cadena. Si no hay entradas, devuelve una cadena vacía.

Longitud
Devuelve el número de caracteres, incluidos los espacios en la cadena. Esta es la longitud de la cadena de texto dada.

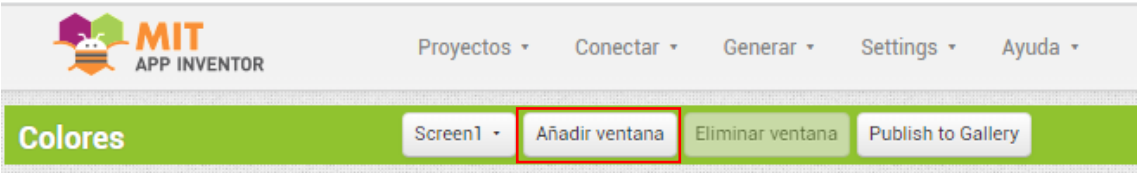
Está vacío
Devuelve si la cadena contiene caracteres (incluyendo espacios) o no. Cuando la longitud de la cadena es 0, devuelve verdadero de lo contrario, devuelve falso.

Compara textos… (Comparador)…
Compara si la primera cadena es lexicográficamente con <, > o = a la segunda.

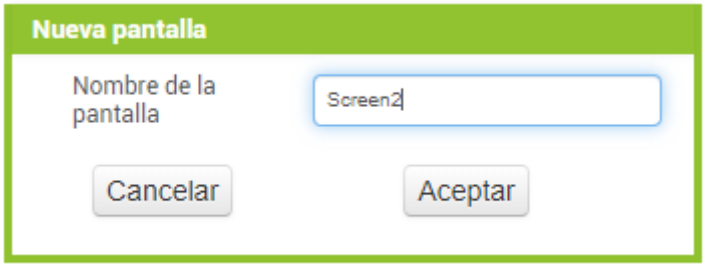
8.- Configurar nuestra App

Aplicaciones Multipantalla

Podemos crear aplicaciones con varias pantallas, las podemos añadir y eliminar a través del siguiente menú:

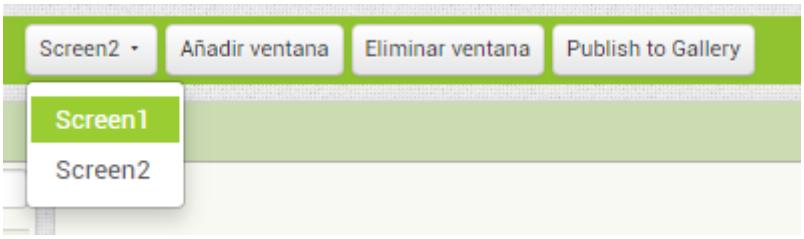


Una vez creada una nueva ventana no podrás modificar el nombre.



Por defecto pone un nombre que tu podrás cambiar.

Como nos podemos desplazar por las ventanas.

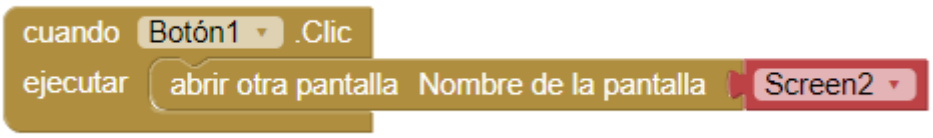


Puedes desplegar el primer botón donde verás todas las ventanas que tú has creado.

Para modificar el titulo de la pantalla, no su nombre en propiedades:

Ahora vamos a ver como podemos programar un botón para que vaya a una determinada ventana.

Esta será la programación en bloques:



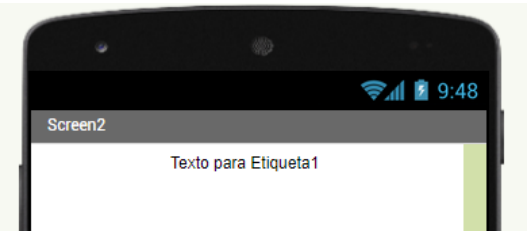
Ten en cuenta que cuando pasamos a otra ventana los valores de las variables que atenemos almacenadas en la primera pantalla, no podremos acceder a ella en la segunda pantalla.

Normalmente se utilizan base de datos para escribir los datos en la primera pantalla y poder leerlos en la segunda pantalla.

No obstante lo que si permite es poder pasar un palor de la primer pantalla a la segunda.

Vamos a diseñar la primera se segunda pantalla:

Primera pantalla

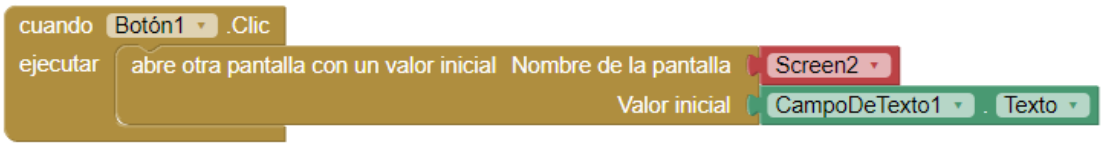


Segunda pantalla

En la primera pantalla tenemos un campo de texto para escribir el texto que nosotros deseemos y un botón para confirmar.

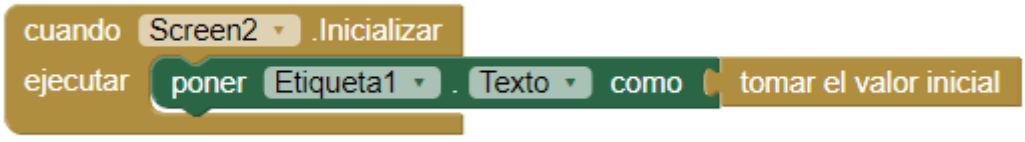
En la segunda pantalla solo tenemos una etiqueta donde queremos que aparezca impreso el texto que escribimos en la primera pantalla.

Programación por bloques de la primera pantalla:



Cuando presionamos el Botón1 vamos a la pantalla2 y le pasamos como valor inicial el contenido que tiene la caja de texto.

Vamos ahora a la segunda página.



Al cargarse la segunda página la Etiqueta1 se le asigne el texto que pasamos de la pantalla1.

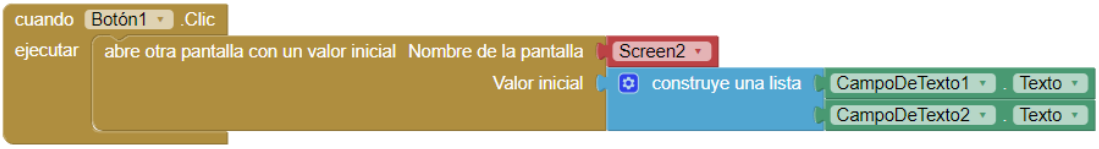
Ahora ya lo puedes probar desde tu móvil.

Si queremos pasar más de un valor este se puede hacer con listas.

Vamos a modificar la pantalla1.

Tiene dos cajas de texto, queremos pasar dos valores a la página2.

Este es la programación en bloques:



La página2 lo que es el diseño continua igual, vamos a ver la programación en bloques:

Pasamos los dos valores de la lista a la Etiqueta1 separadas por un guion.

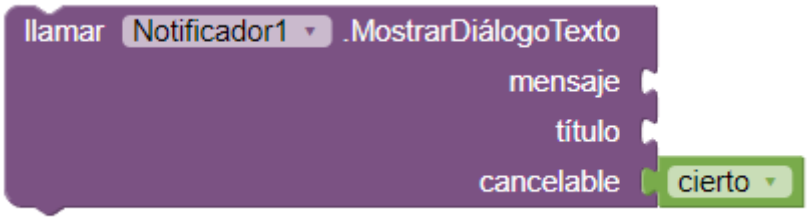
Notificador

El notificador se trata de un componente no visible, el cuál simplemente tenemos que añadir a nuestro diseño, para programarlo posteriormente.

LA PROGRAMACIÓN

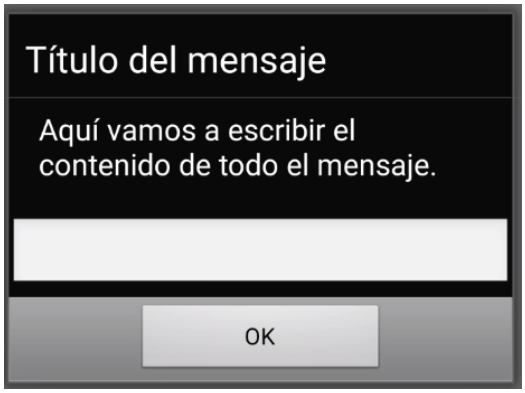
Ya desde “bloques”, será muy sencilla la programación de este pequeña mejora.

- Acudiendo a nuestro botón, seleccionaremos el bloque “cuando botón clic”, bloque al que añadimos “llamador notificador mostrar diálogo de texto” que encontramos dentro del componente notificador. A continuación, le agregamos dos bloques de texto, que nos permitirán poner un mensaje que aparecerá cuando pulsemos el botón, así como un título a este cuadro de texto y modificaremos la opción cancelable a falso.

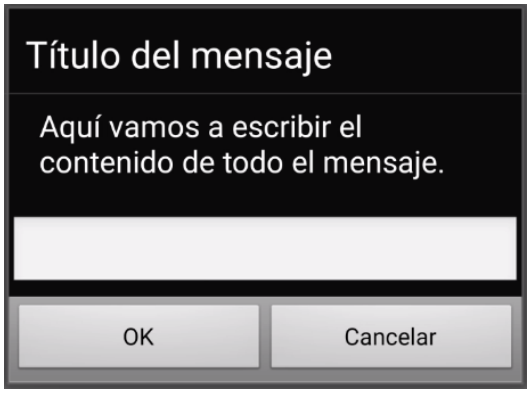


Hemos creado un botón a nuestra pantalla y vamos a programar el botón.

La última opción puede ser cancelable en falso o en cierto, vamos a ver las diferencias en nuestro móvil.



En falso



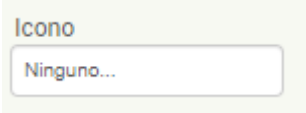
En cierto

Podemos elegir si el botón Cancelar se muestra o no.

Icono y nombre de nuestra App

ICONO

Es muy simple, una vez lo tienes listo con las medidas adecuadas (48x48) lo subes a tu proyecto. Vamos a las propiedades de Screen1 y en la pestaña icono seleccionamos la imagen.



Listo ya tenemos agregado el icono a nuestra app creada con AppInventor.

Nota: Yo he agregado imágenes con otras dimensiones y me las ha aceptado.

NOMBRE DE NUESTRA APP

Desde la misma ventana de Screen1 podemos cambiar el nombre de la aplicación en el campo Appname.

9.- Movimiento: Animación y Dibujo

Lienzo, Sprite y Pelota

LIENZO

Es un panel rectangular de dos dimensiones y sensible al tacto dentro del cual se puede dibujar, y también mover los sprites.

SPRITES

Es un dibujo que debe ir sobre un Lienzo y puede moverse.

PELOTA

Pelota es un tipo específico del conjunto SpriteImagen. La única diferencia es que el caso del componente Pelota no podemos cambiar su aspecto, la imagen del objeto, que siempre será una circunferencia. Si podemos hacerlo sin embargo par cualquier otro SpriteImagen.

Esto se encuentra en el grupo de objetos “Dibujo y animación”.

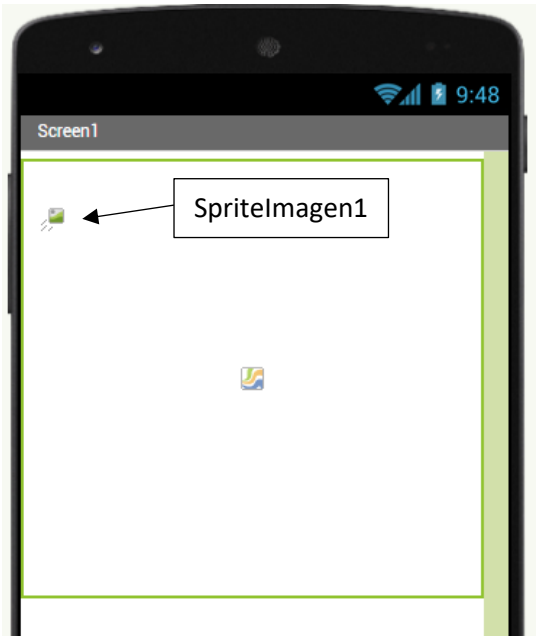
Lo primero que hay que hacer es poner un Lienzo en la base y dentro de esta el resto de los objetos.

Ahora vamos a arrastras un SpriteImagen dentro del lienzo.

En la parte exterior de lienzo no nos la dejará colocar.

Esta es la estructura:

A continuación vamos a modificar el ancho del lienzo, en acho todo el acho de la pantalla y de alto 300 pixeles.



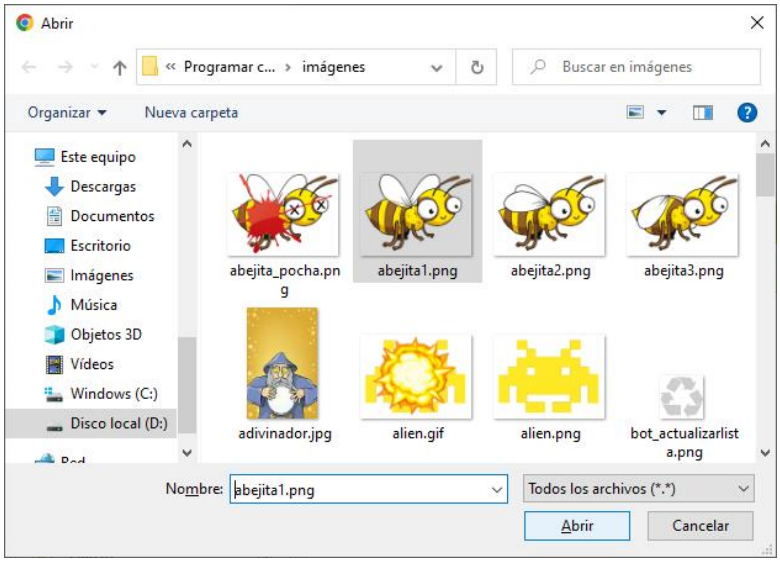
Ahora vamos a agregar una imagen a nuestro SpriteImagen.

Primero lo tenemos que seleccionar y en el la ventana de propiedades:

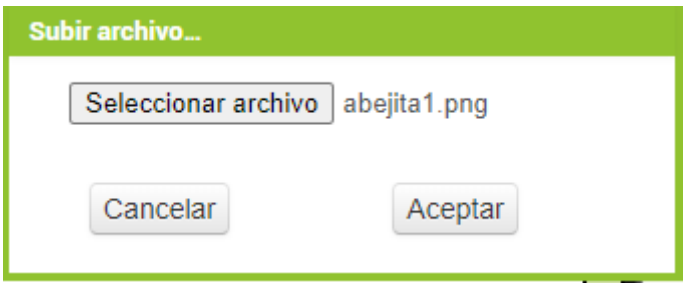
Seleccionar Foto, que nos permitirá subir una imagen y asignársela.

Seleccionamos Subir Archivo… y la cargaremos desde nuestro ordenador.

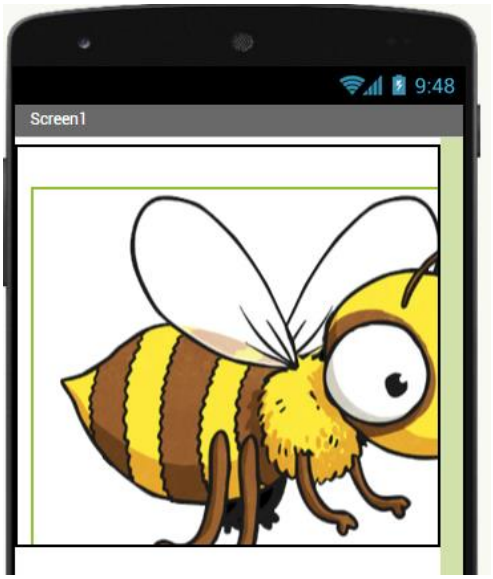
Seleccionaremos el botón Seleccionar archivo.



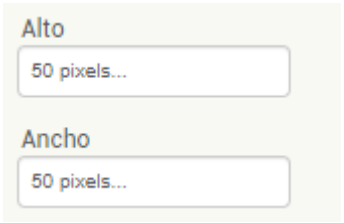
Seleccionamos una imagen seguido del botón Abrir.



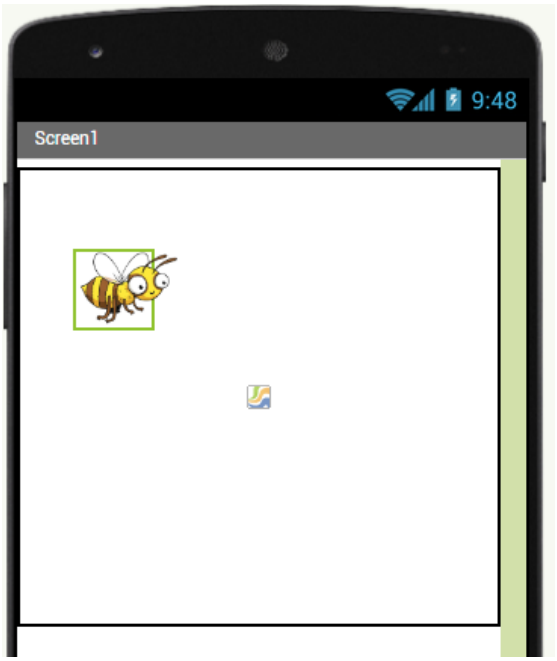
Seguido del botón Aceptar.



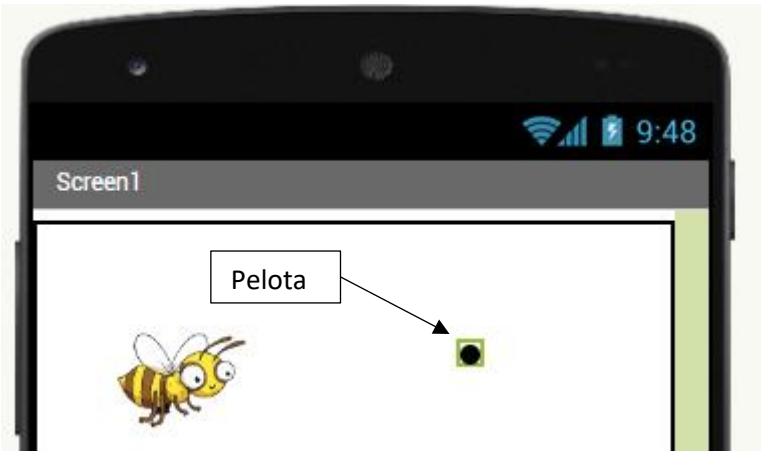
Observamos que la imagen es muy grande, ahora vamos a cambiar las dimensiones:



Este será el resultado:

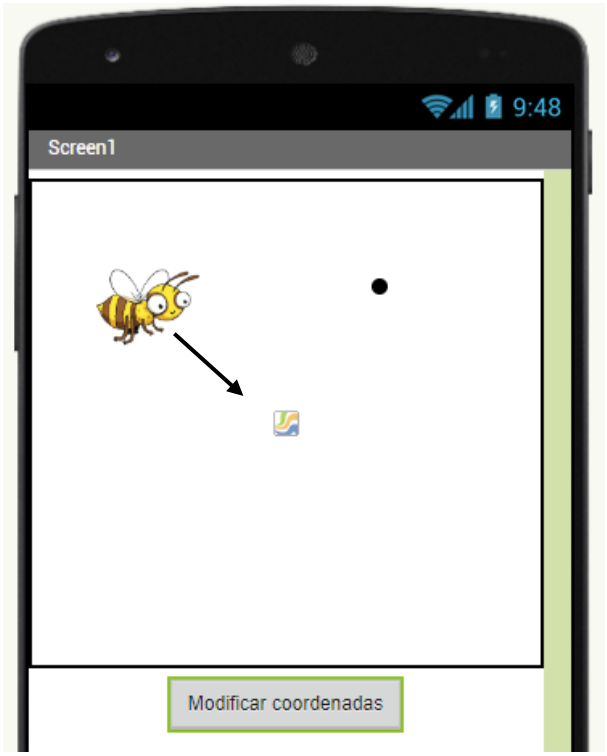


Ahora vamos a agregar una pelota, este no admite modificar al imagen, si el color y el tamaño.

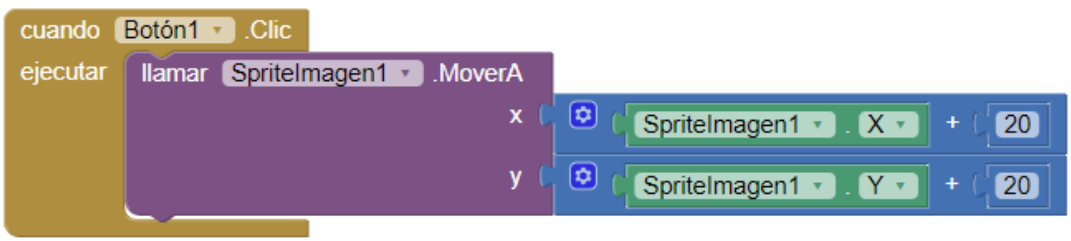


Movimiento de un Sprite por coordenadas

Podemos situar un Sprite en las coordenadas que deseemos a través de las siguientes instrucciones:

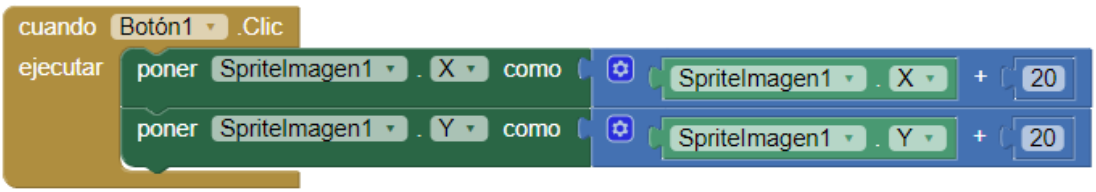


Agregamos un botón, cuando lo seleccionaremos moveremos el Sprite.



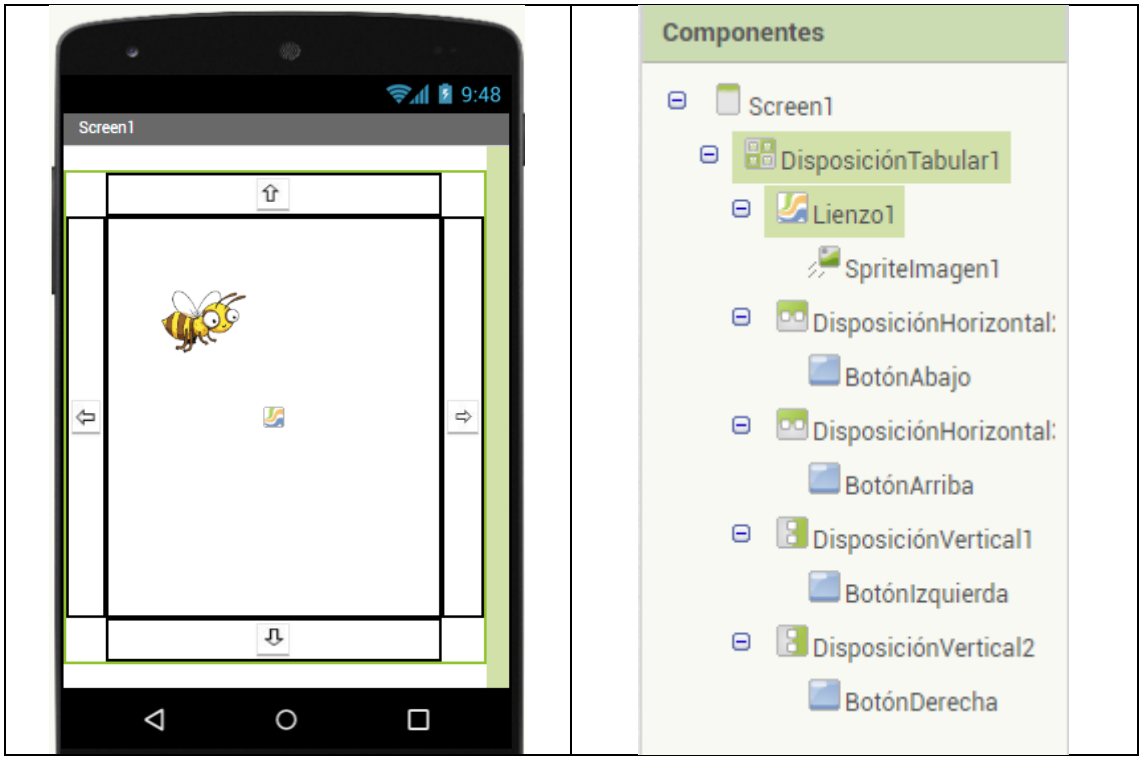
Cada vez que presionemos el botón se moverá en la dirección que indica la flecha.

Otra forma de hacerlo es:

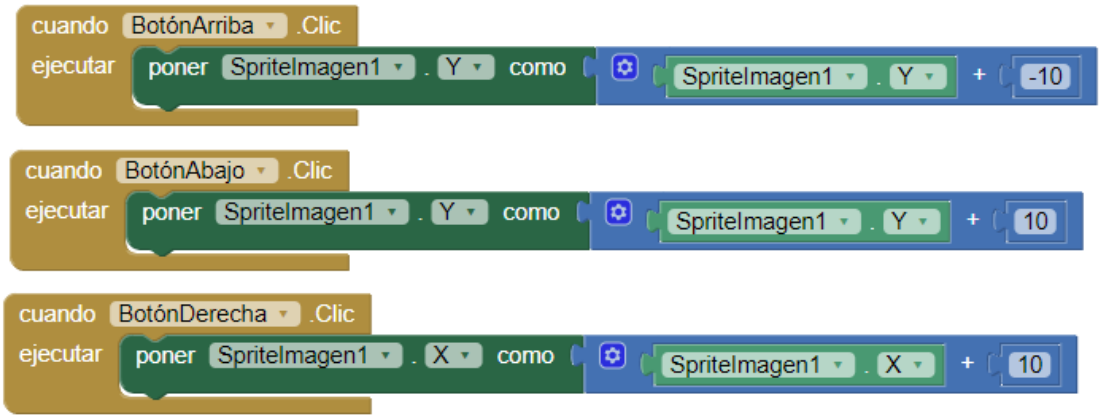


Te propongo la siguiente práctica:

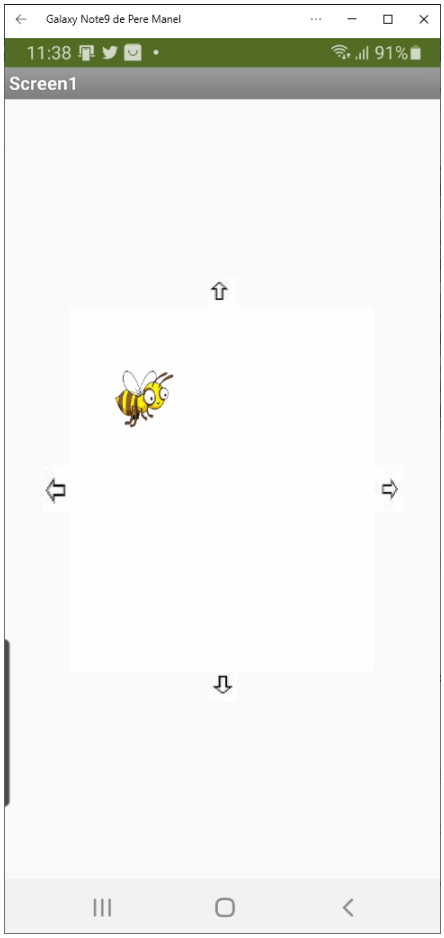
Este tiene que ser el diseño:



Es un buen momento para trabajar las disposiciones.
Ahora vamos a programar los bloques.



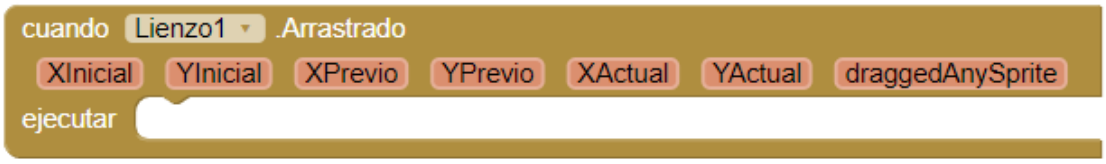
Así se puede ver desde nuestro dispositivo:



Dibujo en la pantalla con el dedo (táctil)

Agregar el Evento de arrastre para dibujar una línea luego, agregarás el controlador del eventos de arrastre. La diferencia entre un touch y un arrastre es el siguiente: Un touch es cuando pones tu dedo en la pantalla y solo lo levantas para tocar sin desplazarlo. Un arrastre es cuando pones tu dedo en la pantalla y lo mueves al mantener contacto con al pantalla. En un programa de pintura, cuando arrastras tu dedo en la pantalla para dibujar una línea igual al camino que sigue tu dedo, incluso un poco, extiendes la línea desde la última posición de tú dedo hacia su nueva posición.

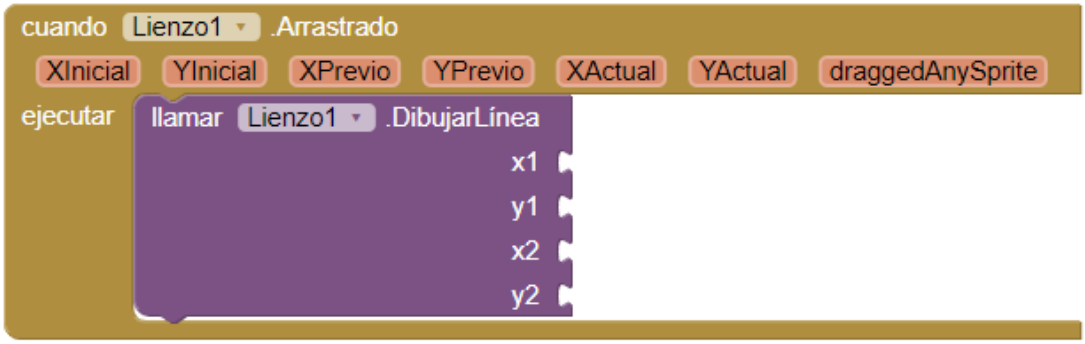
1. En el Editor de bloques, desde la sección del LienzoDeDibujo, arrastras el bloque LienzoDeDibujo. Arrastrando al espacio de trabajo. Deberías ver el gestionador de eventos.



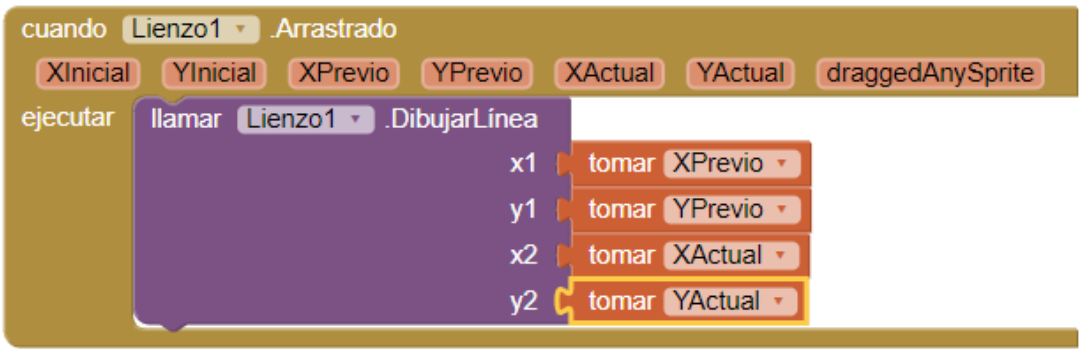
El evento LienzoDibujo.Arrastrado viene con los siguientes argumentos:

- XInicial, YInicial: La posición de tu dedo donde empieza el arrastre.
- XActual, YActual: La posición actual de tu dedo.
- XPrevio, YPrevio: La posición inmediata anterior de tu dedo.
- SpriteArrastrado: Un booleano, será verdad si el usuario hace el arrastre directamente sobre una imagen Sprite.

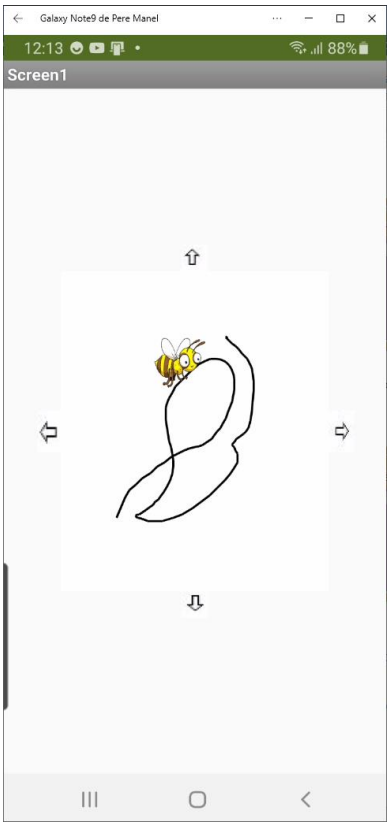
2. Arrastra el bloque LienzoDeDibujjo.DibujarLinea dentro del bloque Lienzo deDibujo.Arrastrado.



3. Arrastra los bloques tomar para los argumentos que vas a necesitar. Los valores XPrevio e YPrevio deberían ir conectados en los espacios para los argumentos x1 e y1, respectivamente.



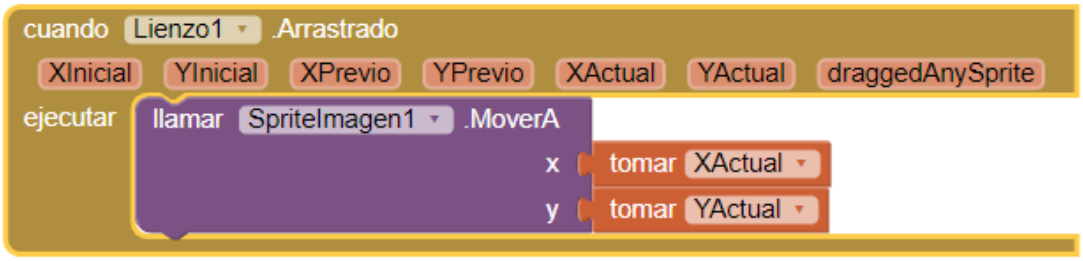
Prueba en tu móvil, ya podrás dibujar en la pantalla de tu móvil.



Anda el lienzo tiene unas líneas, vamos a hacer que nuestro spray siga una ruta que previamente hayamos dibujado.

Movimiento de un Sprite con el dedo (táctil)

Vamos a agregar el siguiente bloque:



Anda, ahora con nuestro dedo podemos mover al sprite.

Lanzar un Sprite con el dedo (táctil)

Al poner el dedo sobre una imagen y arrastrar rápidamente, la imagen se mueve en esa dirección y con la velocidad que lo hayamos realizado con el dedo.

10.- Medios: Sonido e Imagen

Grabar

Todos ellos se encuentran en el grupo de Medios.

GRABAR SONIDO

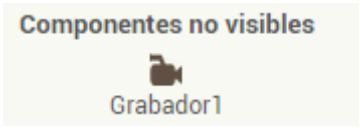
Un componente multimedia que reproduce archivos de sonido y, opcionalmente, vibra durante la cantidad de milisegundos (milésimas de segundo) especificada en el Editor de bloques. El nombre del archivo de sonido que se reproducirá se puede especificar en el Diseñador o en el Editor de bloques.

Para conocer los formatos de archivo de sonido compatibles, consulte Formatos de medios compatibles con Android.

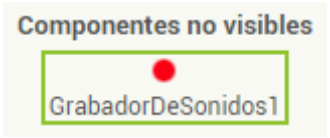
Este Soundcomponente es mejor para archivos de sonido cortos, como efectos de sonido, mientras que el Playercomponente es más eficaz para sonidos largos, como canciones.

GRABADORA VÍDEO

Un componente para grabar un vídeo usando la videocámara del dispositivo. Después de grabar el vídeo, el nombre del archivo en el teléfono que contiene el clip está disponible como argumento del AfterRecordingevento. El nombre del archivo se puede utilizar, por ejemplo, para establecer la propiedad de origen de un VideoPlayercomponente.

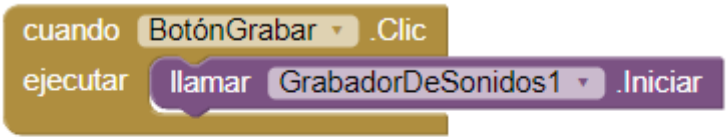


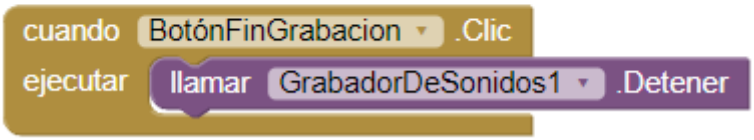
Ahora vamos a realizar una prueba con el grabador de sonidos.



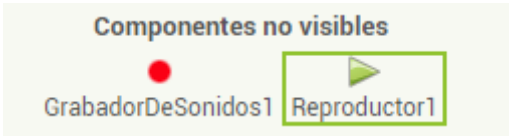
Arrastraremos este componente no visible.

Ahora vamos a programar los bloques.



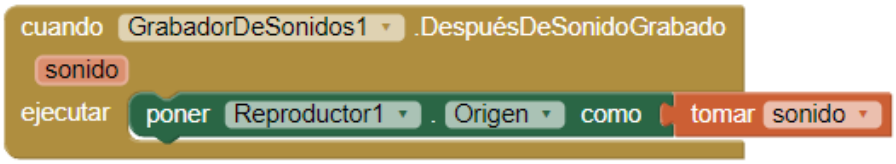


Ahora vamos agregar otro objeto que es el reproductor.



Se sitúa como los elementos no visibles.

Cuando se finaliza la grabación utilizamos la variable sonido para decirle lo que queremos reproducir.



Ahora en el botón de reproducir agregaremos el siguiente bloque de programación:

Para grabar video es algo muy similar utilizando sus respectivos elementos.

Fotos

HACER UNA FOTO

Utilice un componente de la cámara para tomar una foto con el teléfono.

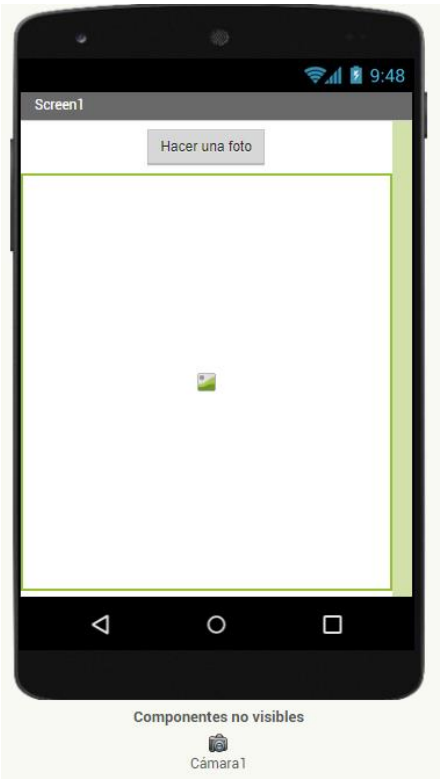
Camera es un componente no visible que toma una foto con la cámara del dispositivo. Después toma la foto, la ruta al archivo en el teléfono que contiene la foto está disponible como argumento del AfterPictureevento. La ruta se puede utilizar, por ejemplo, como Picture Propiedad de una Image componente.

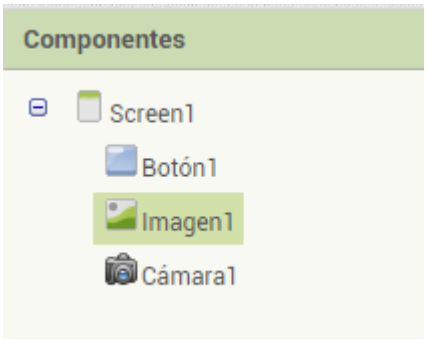
ACCEDER A LA GALERIA DE FOTOS

Un botón de propósito especial. Cuando el usuario toca un SelectorDeImagen, aparece la galería de imágenes del dispositivo y el usuario puede elegir una imagen. Después de seleccionar una imagen, se guarda y la Selection propiedad será el nombre del archivo donde se almacena la imagen. Para no llenar el almacenamiento, se almacenará un máximo de 10 imágenes. Al seleccionar más imágenes, se elimina las imágenes anteriores. En orden de más antigua a la más reciente.

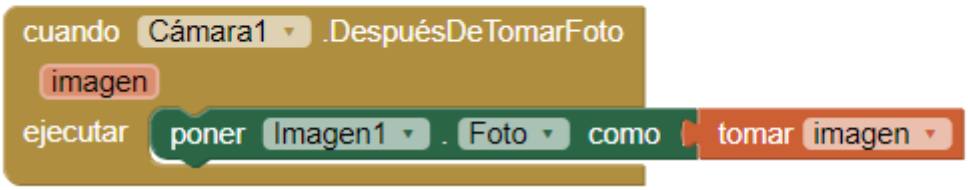
Al arrastrarla esta funcionará como un botón.

Vamos a realizar un ejemplo de como debemos realizar un proyecto utilizando la cámara.

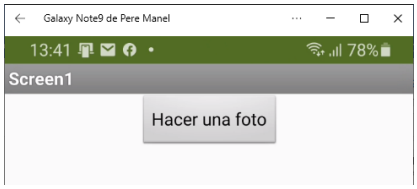




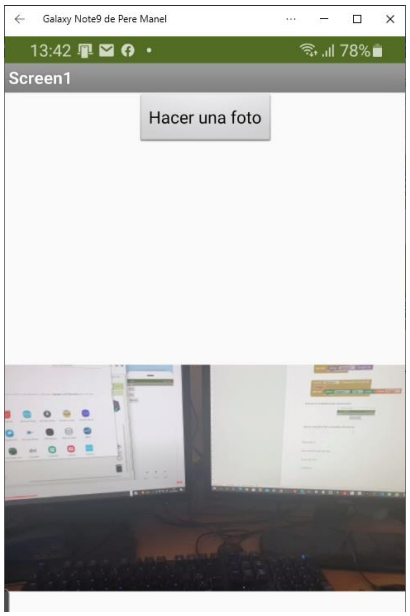
Ahora vamos a los bloques de programación:



Este será el resultado desde nuestro móvil.

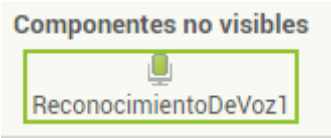


Una vez tomada la foto y aceptada como buena.



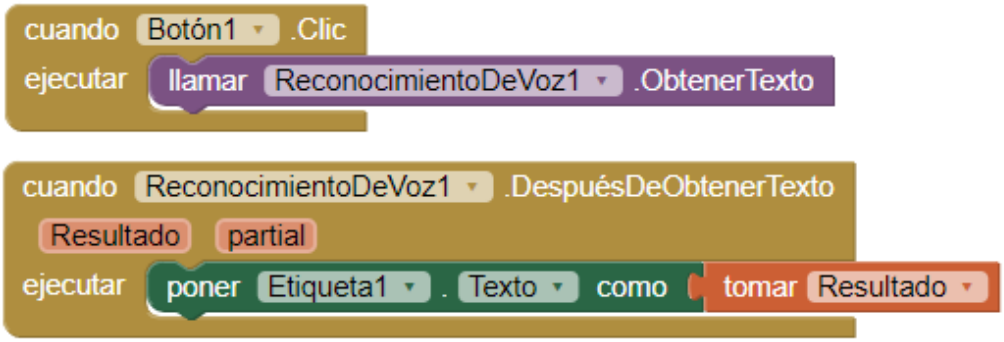
Reconocimiento de Voz

Hay una instrucción que es ReconocimientoDeVoz.ObtenerTexto que lanzamos cuando queremos y cuando esto ha ocurrido saltará un evento: cuando Reconocimiento DeVoz.DespuesDeobtenerTexto.

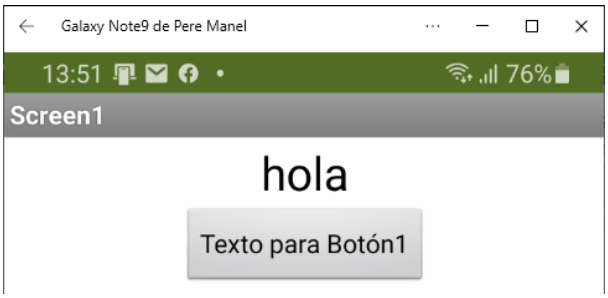


Vamos crear un nuevo proyecto que además de agregar el ReconocimientoDeVoz, vamos a agregar una etiqueta y un botón.

Este será la programación por bloques:



Una vez de decir hola este será el resultado en mi móvil:

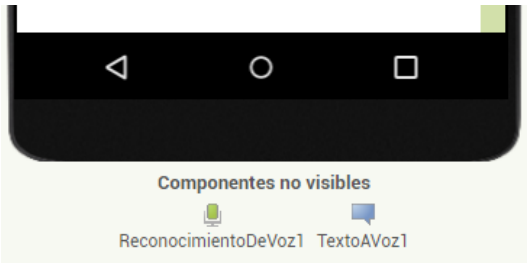


Texto de Vox

Cuando queremos utilizarlo hay que usar estas instrucciones.

- Poner TextoAVoz.Pais como “ESP”
- Poner TextoAVoz.Idioma como “es”

Y luego para oírlo llamar a TextoAvoz.Hablar y poner el mensaje que queramos decir.



La estructura:

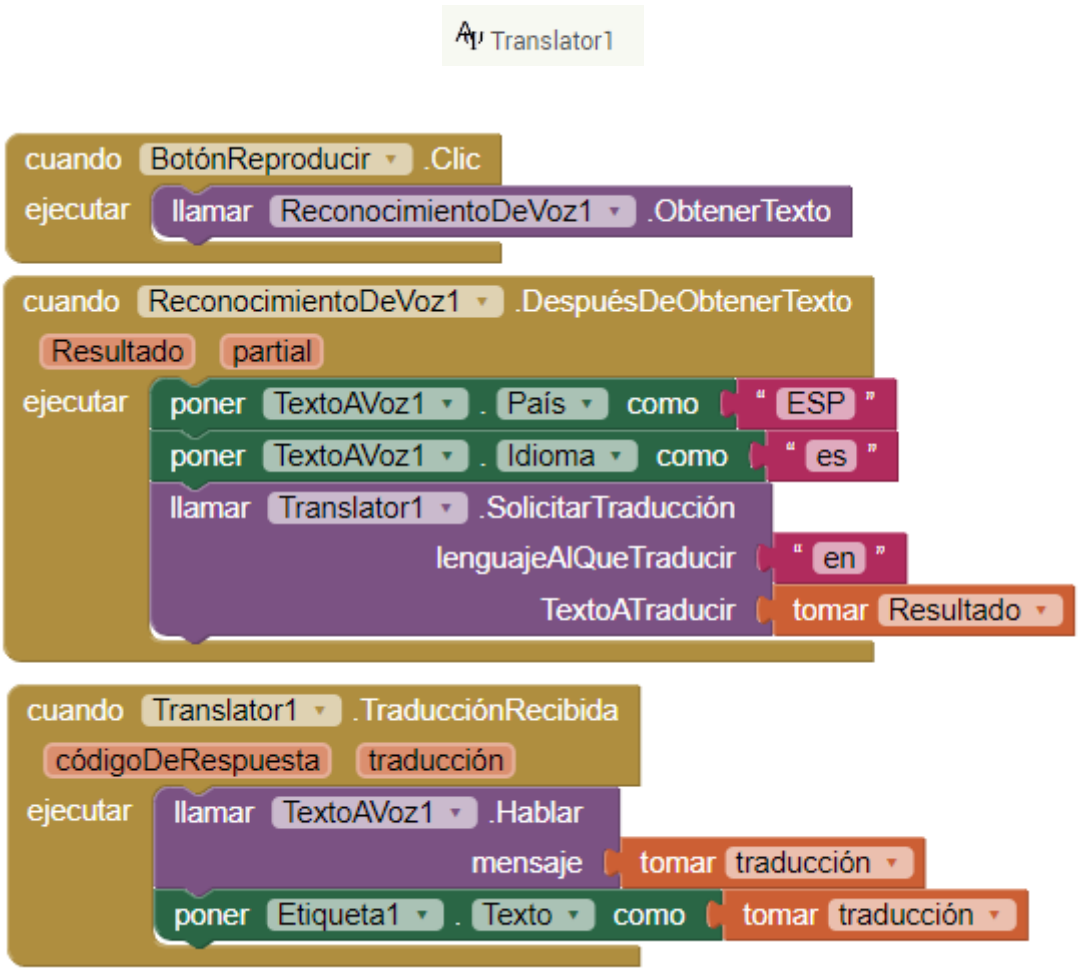
Por último la programación en bloques:

Escribe el texto que tú quieras y dale al botón Reproducir desde tu móvil.

Traductor

Llamar Traductor1.SolocitarTraduccion (lenguajes al que traducir, y el texto que queremos traducir)

Cuando se ha producido la traducción salta el evento… CuandoTraductor1.TraduciónRecibida que el valor “traducción” tiene el elemento traducido.



Ahora ya lo puedes probar en tu móvil.

Adjunto tabla de idiomas:

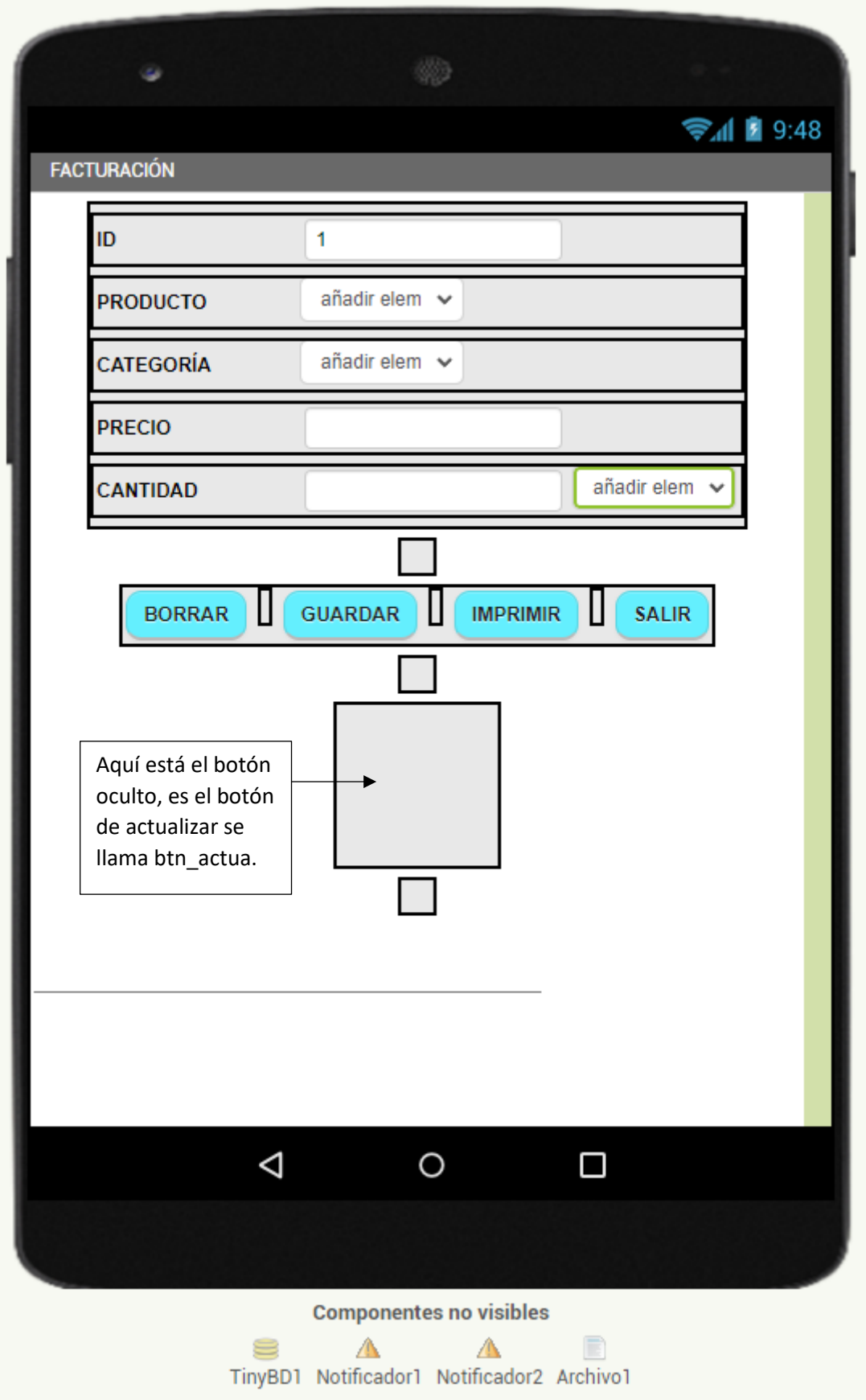
Language	Code	Language	Code
Azerbaijani	az	Malayalam	ml
Albanian	sq	Maltese	mt
Amharic	am	Macedonian	mk
English	en	Maori	mi
Arabic	ar	Marathi	mr
Armenian	hy	Mari	mhr

Afrikaans	af	Mongolian	mn
Basque	eu	German	de
Bashkir	ba	Nepalese	ne
Belarusian	be	Norwegian	no
Bengal	bn	Punjabi	pa
Burmese	my	Papiamento	pap
Bulgarian	bg	Persian	fa
Bosnian	bs	Polish	pl
Welsh	cy	Portuguese	pt
Hungarian	hu	Romanian	ro
Vietnamese	vi	Russian	ru
Haitian (Creole)	ht	Cebuano	ceb
Galician	gl	Serbian	sr
Dutch	nl	Sinhalese	si
Hill Mari	mrj	Slovak	sk
Greek	el	Slovenian	sl
Georgian	ka	Swahili	sw
Gujarati	gu	Sundanese	su
Danish	da	Tajik	tg
Hebrew	he	Thai	th
Yiddish	yi	Tagalog	tl
Indonesian	id	Tamil	ta
Irish	ga	Tartar	tt
Italian	it	Telugu	te
Icelandic	is	Turkish	tr

Spanish	es	Udmurt	udm
Kazakh	kk	Uzbek	uz
Kannada	kn	Ukrainian	uk
Catalan	ca	Urdu	ur
Kirghiz	ky	Finnish	fi
Chinese	zh	French	fr
Korean	ko	Hindi	hi
Xhosa	xh	Croatian	hr
Khmer	km	Czech	cs
Laotian	lo	Swedish	sv
Latin	la	Scottish	gd
Latvian	lv	Estonian	et
Lithuanian	lt	Esperanto	eo
Luxembourg	lb	Javanese	jv
Malagasy	mg	Japanese	ja
Malay	ms		

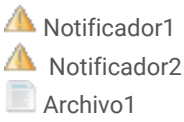
11.- Proyecto de facturación

Diseño:



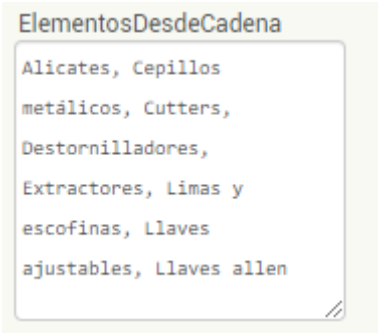
Estructura:


Screen1
DisposiciónVertical1
DisposiciónHorizontal1
Etiqueta1
cp_id
DisposiciónHorizontal2
Etiqueta2
id_producto
DisposiciónHorizontal3
Etiqueta3
id_categoría
DisposiciónHorizontal4
Etiqueta4
camp_prc
DisposiciónHorizontal5
Etiqueta5
camp_cant
Id_um
DisposiciónVertical2
DisposiciónHorizontal6
btn_borrar
DisposiciónHorizontal7
btn_guardar
DisposiciónHorizontal8
btn_imprimir
DisposiciónHorizontal9
btn_salir
DisposiciónHorizontal10
DisposiciónHorizontal11
btn_actua
DisposiciónHorizontal12
VisorDeLista1
TinyBD1

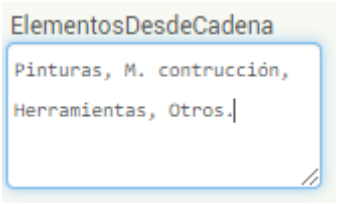


Después de realizar el diseño renombrar los elementos según la estructura.

En la lista id_producto en las propiedades de ElementosdesdeCadena vamos a agregar las siguientes opciones que tienen que estar separadas por coma.



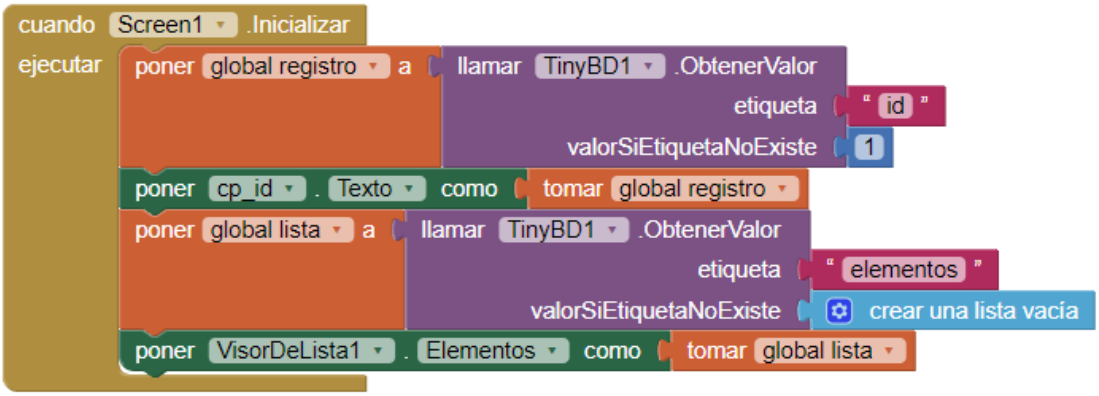
En la lista id_categoría:



El botón btn_actua no tiene que estar visible.

Ahora vamos a comentar una parte de la programación en bloque:

Definimos dos variables globales, una llamada lista que llevará la información de los productos que queremos agregar y otra llamada registro que almacenará el número de registro del siguiente artículo que agreguemos.



Cuando ejecutemos el programa este código se va a ejecutar automáticamente.

A la variable registro le pasamos el valor del último registro que está almacenado en la etiqueta id que se almacenó cuando estábamos agregando artículos , si no hubiera registros porque es la primera vez que lo ejecutamos le asignaríamos a la variable registro el valor de 1, se lee en la base de datos.

Queremos que muestre en la etiqueta cp_id.texto el valor obtenido por la variable registro.

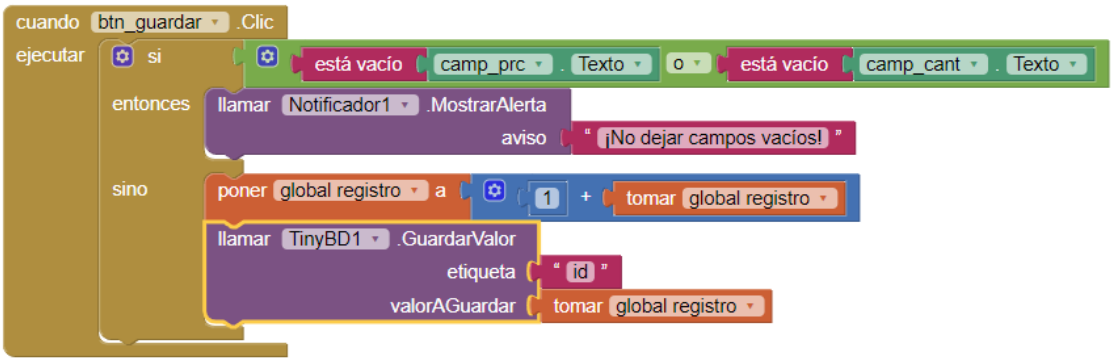
En la variable lista queremos obtener los valores almacenados en la etiqueta elementos si no tuviera porque lo estamos ejecutando por primera vez crearía una lista vacía, se lee en la base de datos.

Para que muestre los artículos que hemos agregado los agregamos al VisorDeLista1.Elementos con el valor de la variable lista.

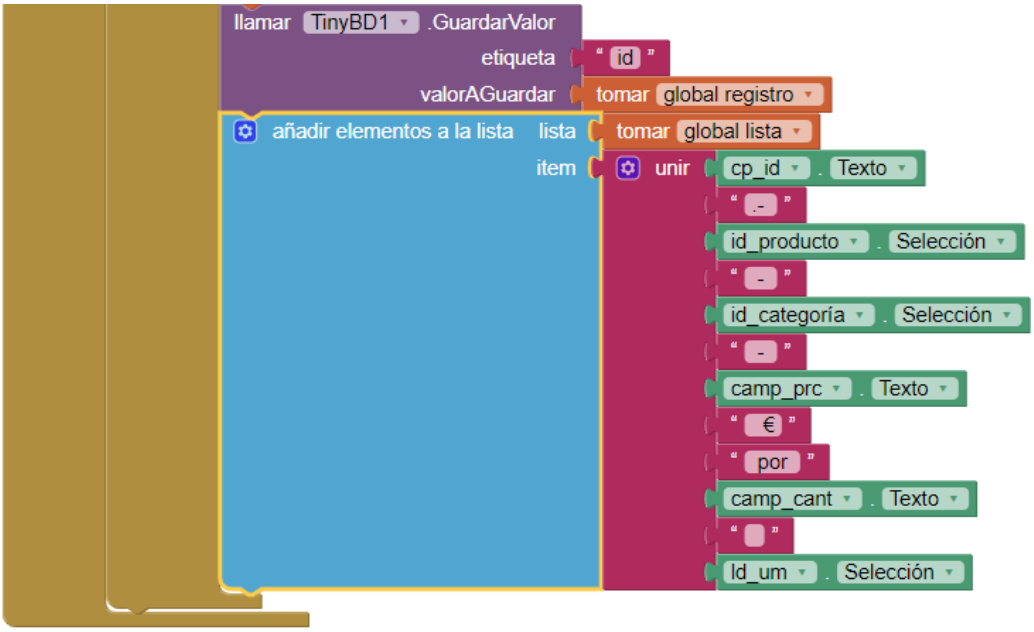
Vamos a programar el botón btn_guardar con el evento clic.

Al hacer clic sobre el botón btn_guardar vamos a comprobar que tanto el campo camp_pre o el campo camp_cant no están vacíos, a que de este modo nos aparecería un mensaje notificando ¡No dejar campos vacíos!.

De lo contrario, quiere decir que hemos introducido todos los campos la variable registro a su valor se le incrementa 1.

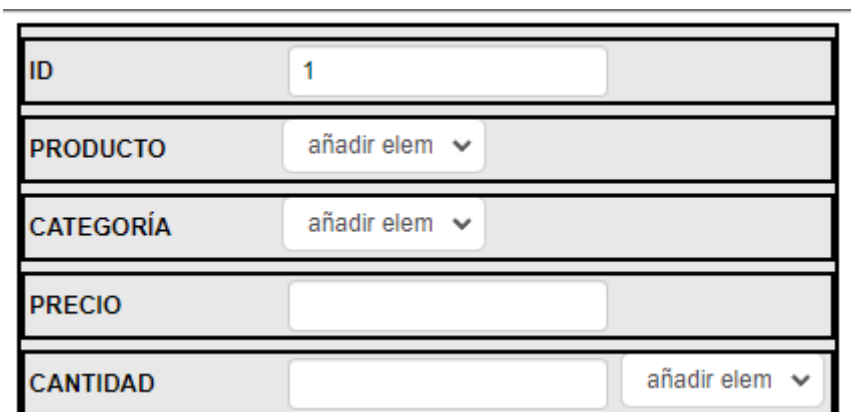


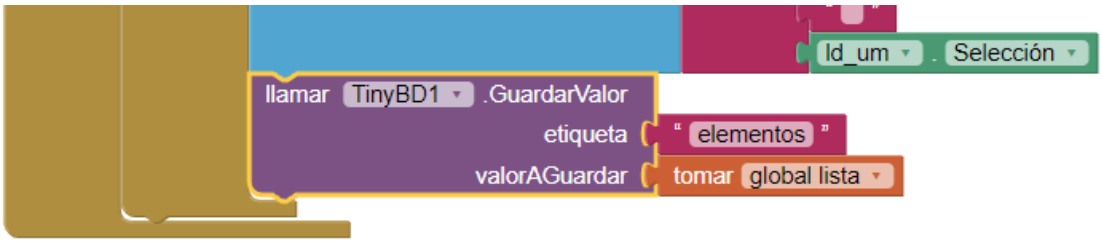
Guardamos en la etiqueta id el valor de la variable registro (Se guarda en la base de datos).



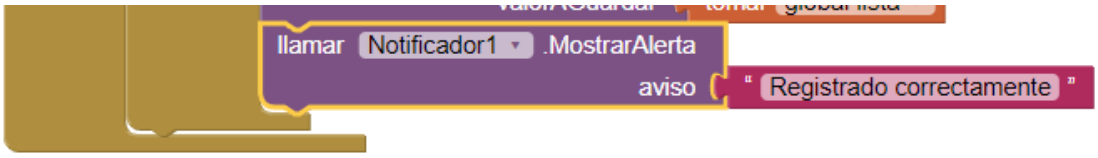
El la variable lista, le agregamos los valores de cp-id, id_producto, id_categoria, camp_prc, camp_cant y id_um todo ello concatenado con distintos bloques de caracteres.

Estos valores has sido extraídos de los valores del formulario:

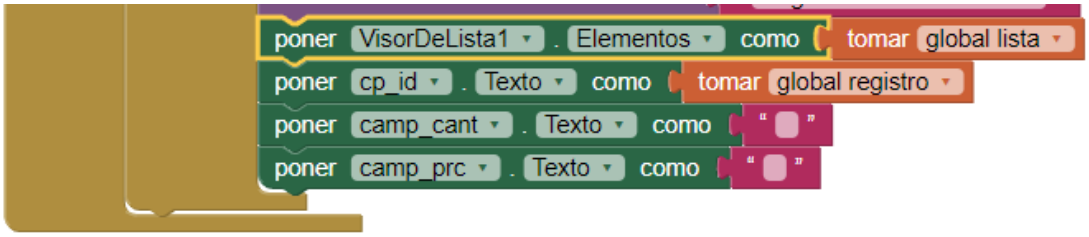




A continuación en la base de datos con la etiqueta elementos guardamos el valor que tiene la variable lista con la etiqueta elementos.



Mostramos un mensaje que se mostrará en pantalla con el siguiente mensaje: Registrado correctamente.

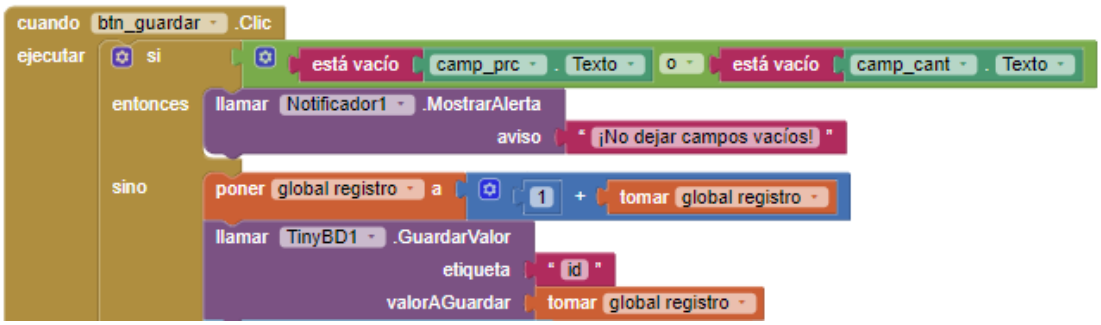


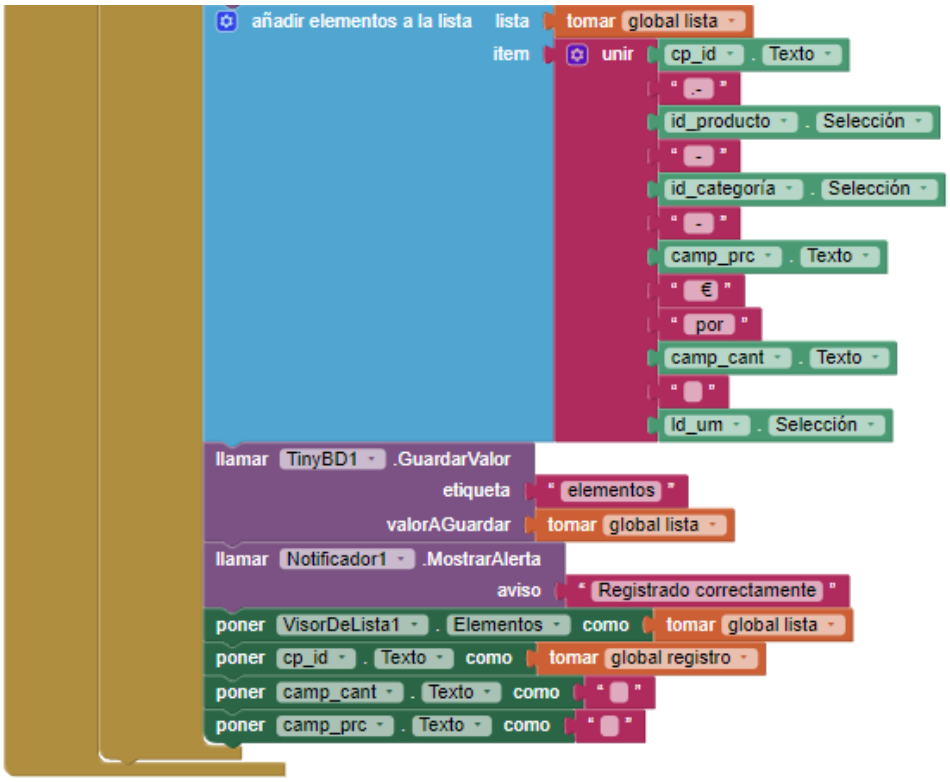
Que nos muestre la información de la variable lista en el VisorDeLista1, la parte inferior de nuestra pantalla.

En el campo cp_id.Texto el valor de registro, será el número siguiente ya que en anteriores bloques lo incrementamos 1.

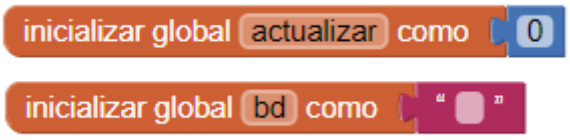
Los campos camp_cant y camp_prc los dejamos vacíos para poder agregar más registros.

Este tiene que ser el resultado final:

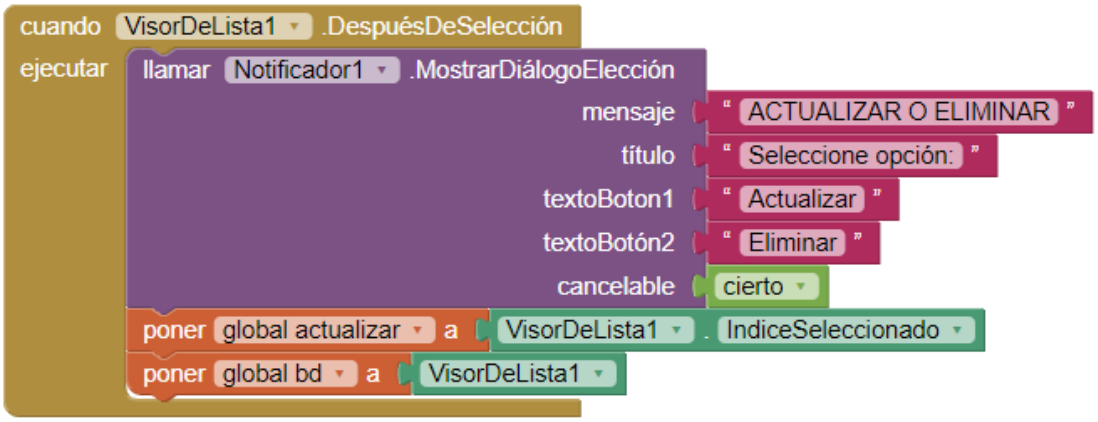




Definimos dos variables globales más:



Después de seleccionar el visor:



Llamamos el Notificador1.MostrarDiálogoElección, con un mensaje “ACTUALIZAR O ELIMINAR”, un título “Seleccione opción:” y dos botones con el texto “Actualizar” y “Eliminar”, cancelable igual a cierto porque queremos que además nos muestre un botón de “Cancelar”.

A la variable actualizar le pasamos el valor del índice que hemos seleccionado en el VisorDeLista1.

La variable bd le pasamos todos los elementos que tiene el VisroDeLista1.

Después de seleccionar el notificador:

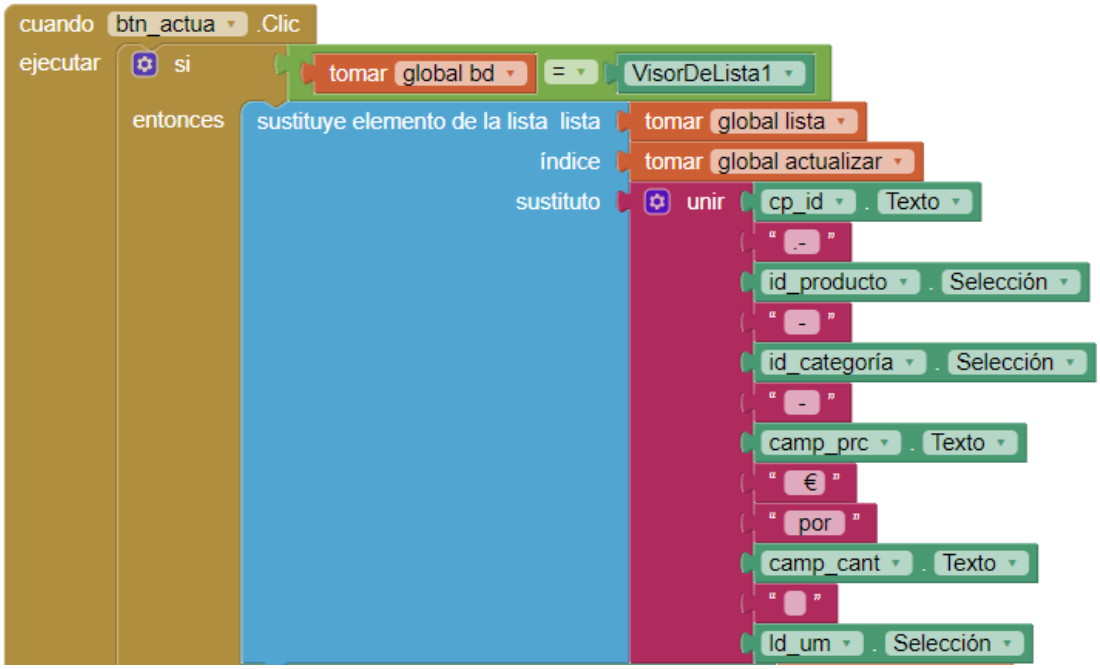
Comparamos si has seleccionado Eliminar, si es así eliminamos de la variable lista en índice del valor de la variable actualizar.

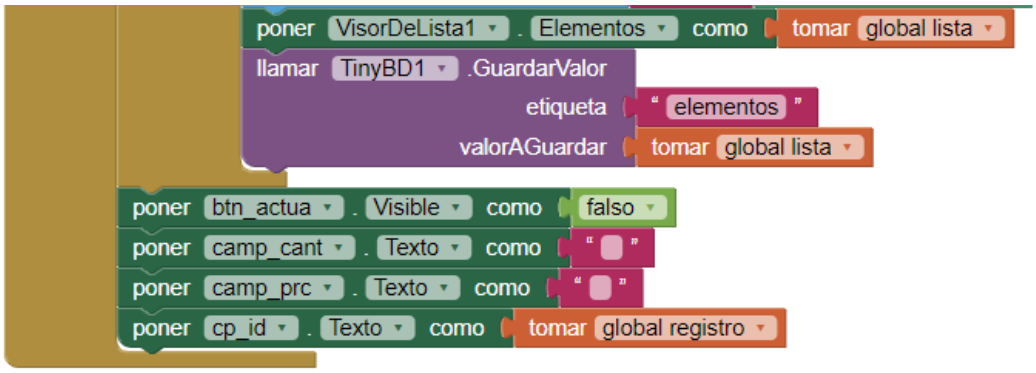
Si has seleccionado Actualizar en el campo de texto cp_id del segmento de texto la la variable lista del índice actualizar seleccionamos desde 0 al segundo carácter, es el número de registro que tiene el producto.

A continuación hacemos visible el botón btn_actua para poderlo seleccionar.

A continuación tanto se elegimos Eliminar como Actualizar, guardaremos en la base de datos con la etiqueta “elementos” el valor de la variable lista.

Mostraremos en el VisorDeLista1.Elementos el valor de la variable lista, en el caso de haber elegido la opción “Eliminar” este registro ya no se visualizará.





Cuando hacemos clic sobre el botón btn_actua:

Si la variable bd es igual a lo que contiene el VisorDeLista1

Reemplazamos de la variable lista en el índice actualizar con toda la información que tiene el formulario, si un campo se ha modificado este se actualizará.

El VisorDeLista1.Elementos mostrará los cambios realizados en el registro que hemos actualizado.

Guardamos de nuevo en la base de datos con la etiqueta “elementos” le valor de la variable lista.

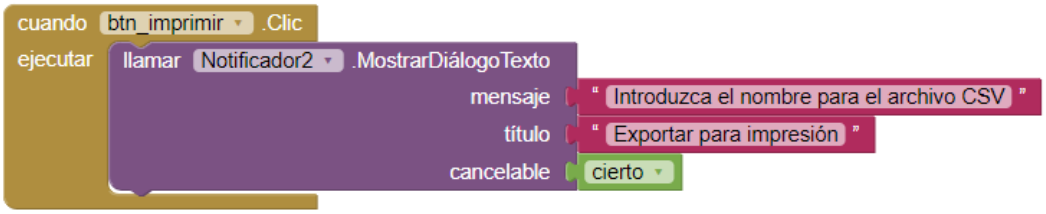
Fuera de condición:

El botón btn_actua lo hacemos ni visible de nuevo.

Los campos camp_cant y camp_prc los dejamos vacíos.

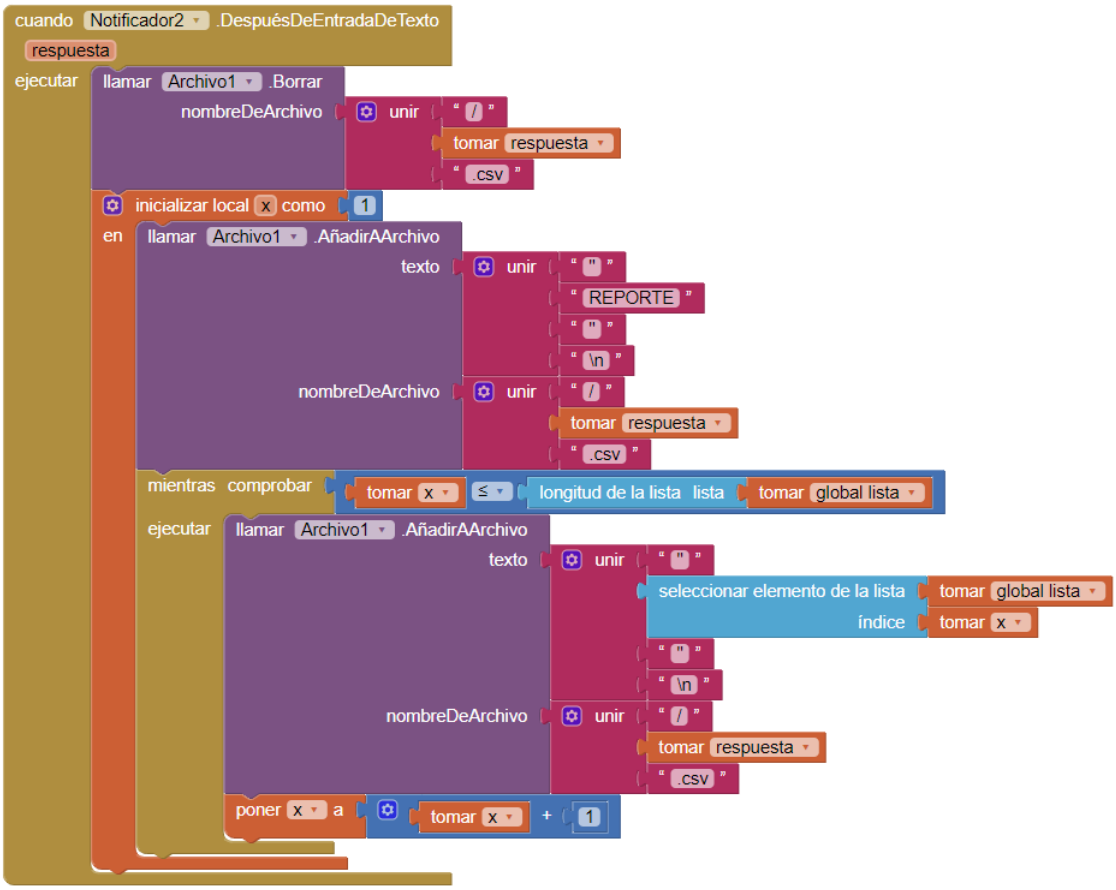
Mostramos en el campo cp_id.Texto el valor de la variable registro que será el siguiente valor para que podamos agregar un nuevo registro.

Ahora vamos a continuar con el botón imprimir.



Cuando hacemos clic en el mostrará el siguiente Diálogo:

Donde pondremos el nombre del archivo seguido del botón OK.



Cuando hayamos introducido el nombre del archivo:

Llamamos a Archivo1.Borra, para que elimine el fichero que generamos con anterioridad.

Definimos una variable local llamada x con el valor inicial de 1.

Como nombre de archivo “/respuesta.csv” y texto “REPORTE” + “\n” Salto de línea.

Hacemos un bucle que mientras x sea menos o igual a la longitud de la lista de la variable lista.

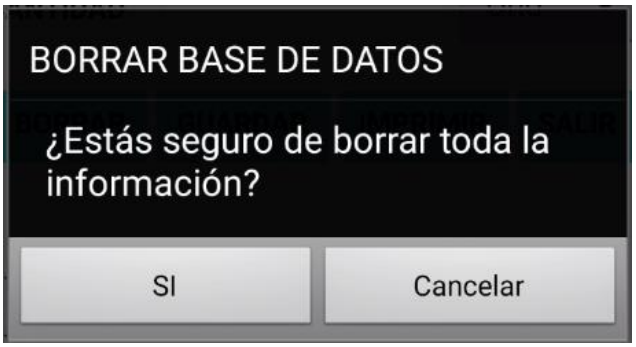
Añadimos al archivo el texto “ de la variable lista el elemento que tiene la variable x.

Como nombre de archivo “\” más el valor de la variable respuesta, nombre que le dimos al archivo, seguido de un punto más su extensión (.csv).

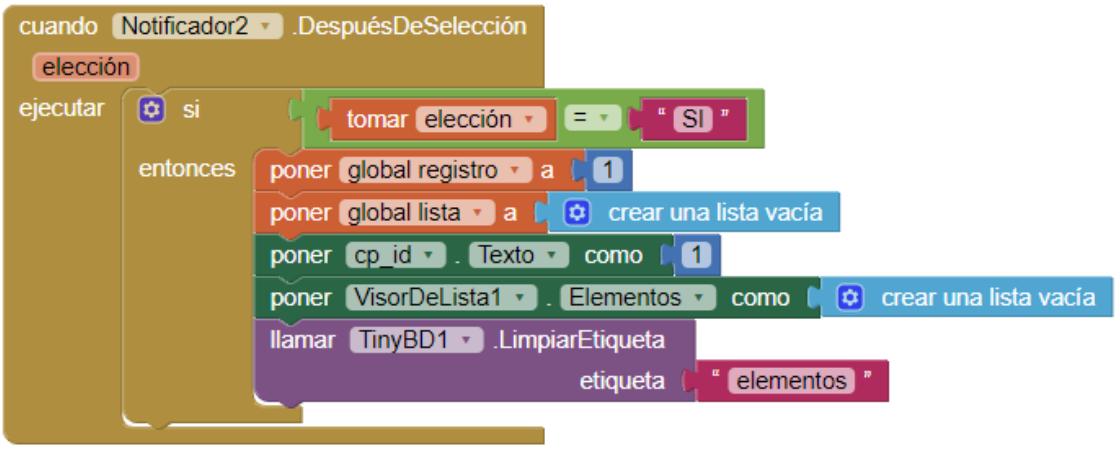
A la variable x a su valor le incrementamos 1 para poder leer el siguiente registro, este bucle continuará hasta que x sea mayor al número de registros de la variable lista.

Ahora para borrar la base de datos vamos a programar el botón btn_borrar.

Cuando hacemos clic sobre el botón btn_borrar observaremos la siguiente ventana de diálogo:



Si seleccionamos el botón “SI”.



Después de selección del Notificador2 comparamos si la variable elección es igual a “SI”.

La variable registro le asignamos el valor de 1.

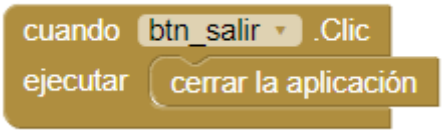
La variable lista le asignamos una lista vacía.

El campo cp_id le pasamos el valor de 1.

El visorDeLista1.Elementos tiene que mostrar una lista vacía.

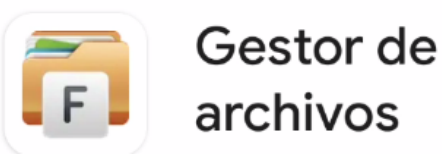
Borramos de la base de datos todos los datos de la etiqueta elementos.

Ahora vamos a programar el botón btn_salir:



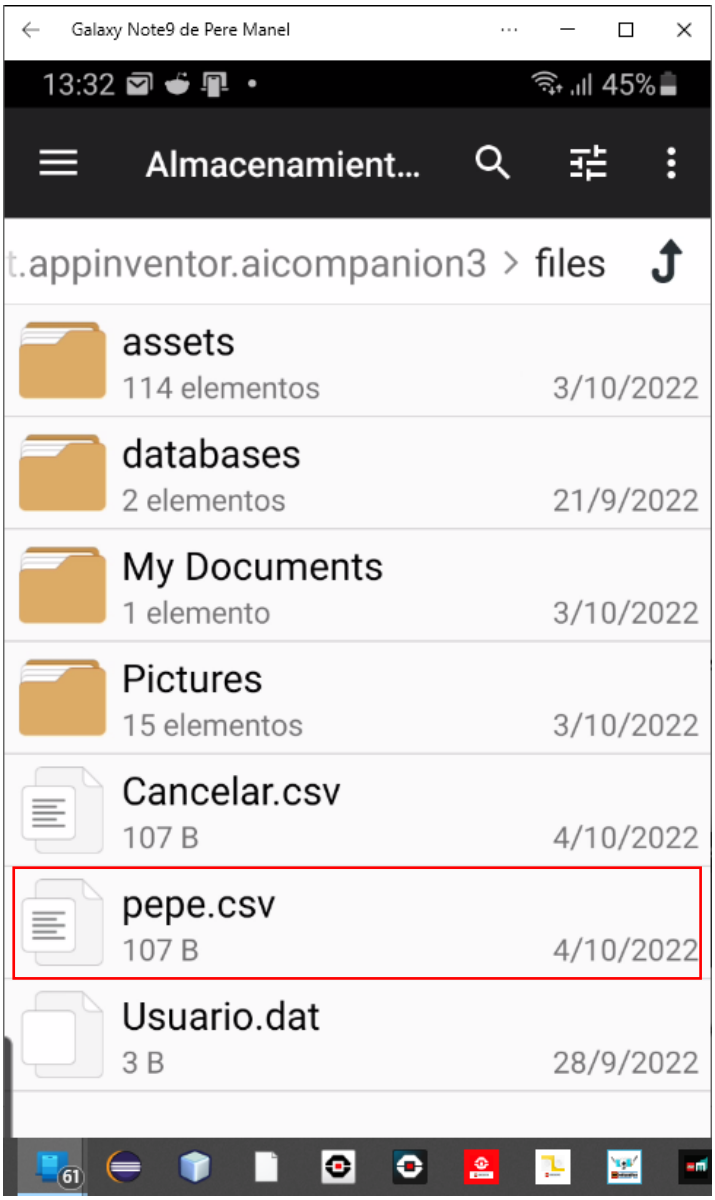
Cerramos la aplicación.

Ahora para comprobar si nos ha generado el archivo:



Instalamos esta aplicación desde la AppStore.

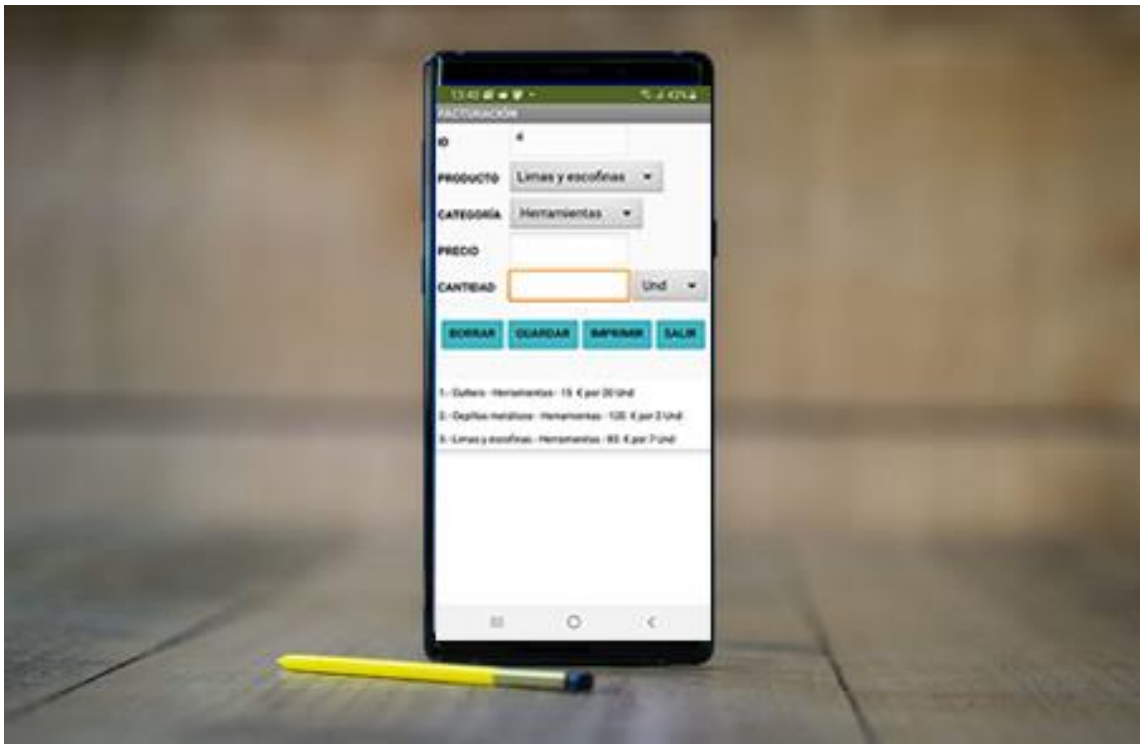
Desde la siguiente ruta: Inicio > Android > data > edu.mit.appinventos.aicompanion3 > files



Si lo abrimos veremos el contenido del archivo.

Todo funciona correctamente, este será el aspecto de nuestra aplicación desde nuestro móvil.

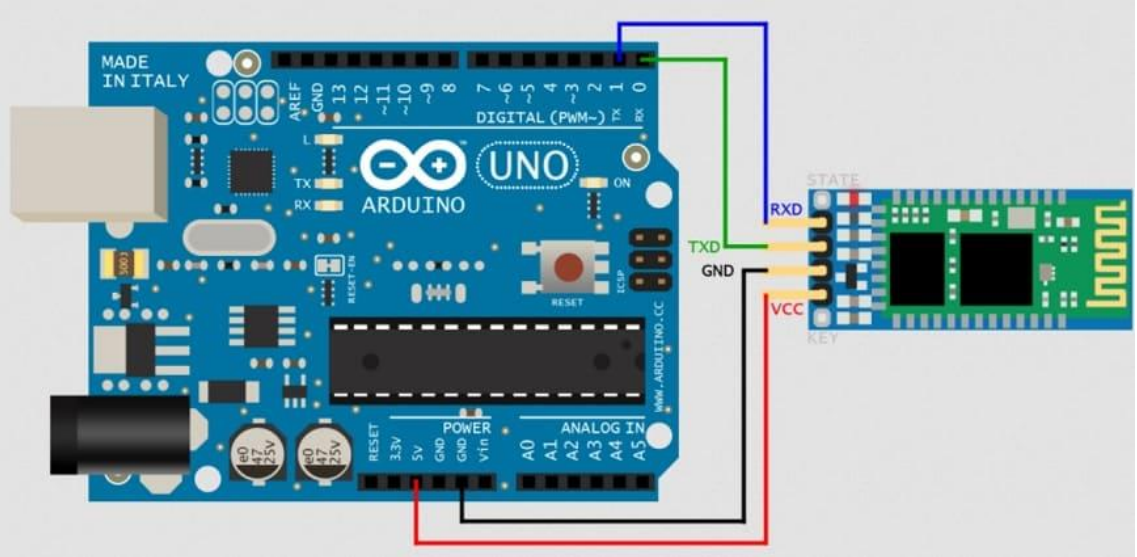
Este será el resultado final en mi móvil.



12.- Bluetooth y Arduino

Antes de empezar este capítulo vamos a dar una pequeña introducción de Arduino para poder realizar esta práctica.

Conectar el Bluetooth a la placa de Arduino.



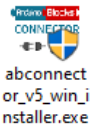
Descargar esta app de AppStore para instalarla en nuestro móvil.

Cuando tengamos nuestros proyectos realizados la lo emparejaremos con nuestro móvil.

Vamos a programar nuestro Arduino on-line (desde una página web) para ello tenemos que instalar una pequeña aplicación en nuestro ordenador para que desde la página web donde vamos a programarlo reconozca nuestro dispositivo.

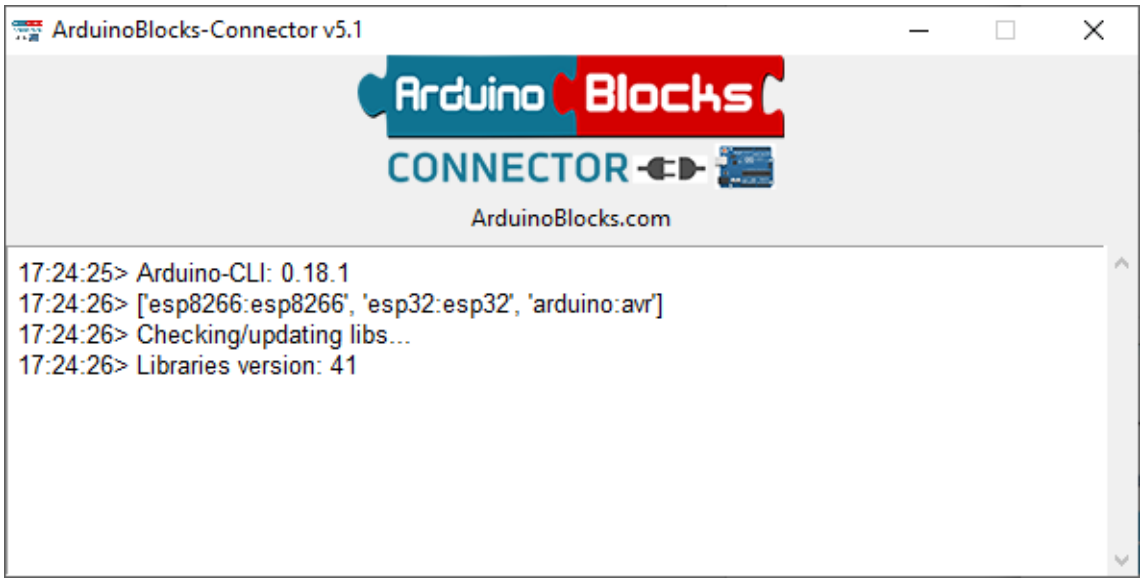
Accederemos a la siguiente página web:
http://www.arduinoblocks.com/web/site/abconnector5

Nosotros vamos a descargar la versión para Windows.



Una vez lo tengas descargado lo podrás instalar.

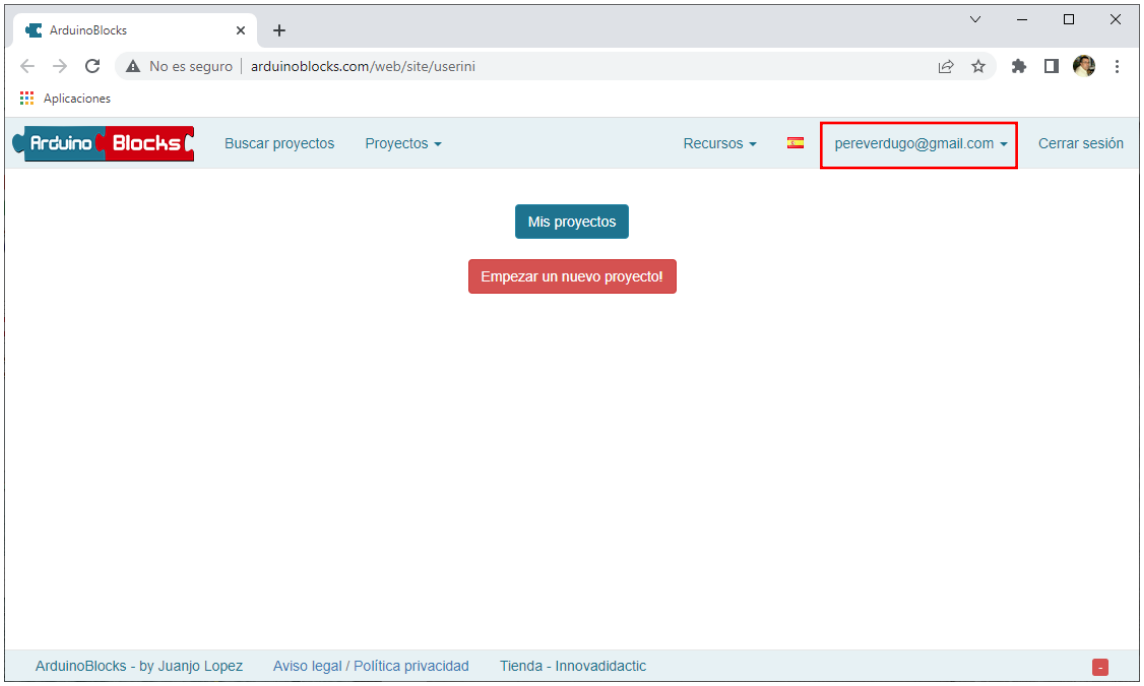
Antes de entrar a la página web donde programaremos el Arduino tienes que ejecutar esta aplicación.



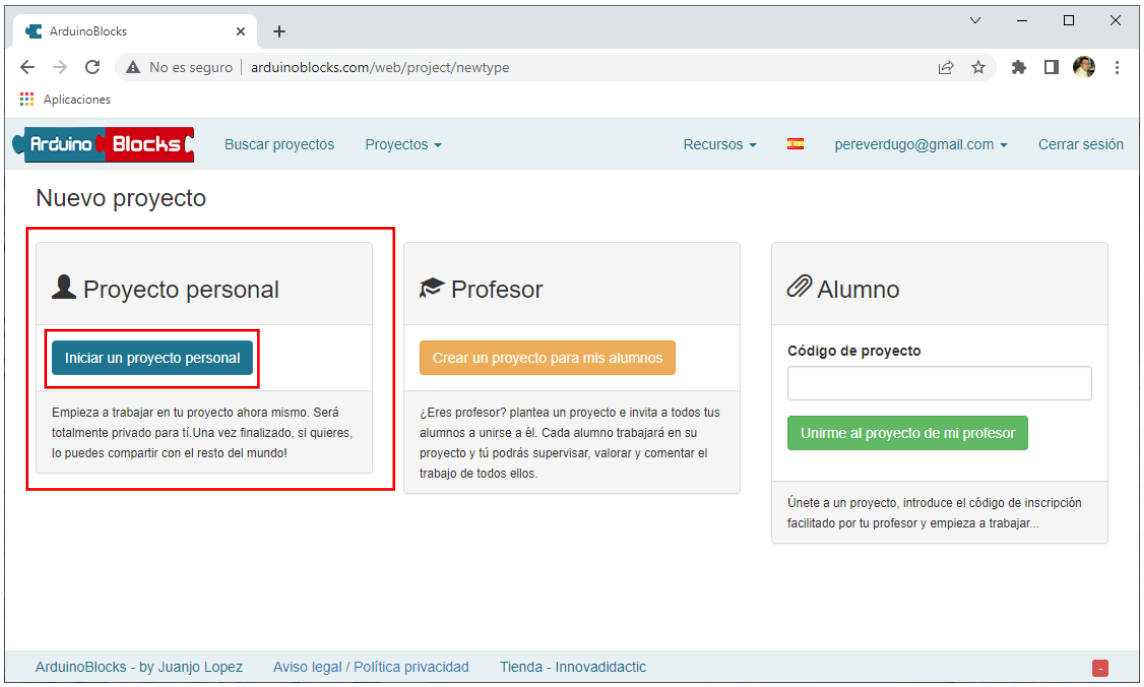
Para que reconozca en que puerto USB está conectado el Arduino.

La dejas ejecutando y ahora vamos a ir a la siguiente dirección para empezar a programar el Arduino.

http://www.arduinoblocks.com/web/site/userini

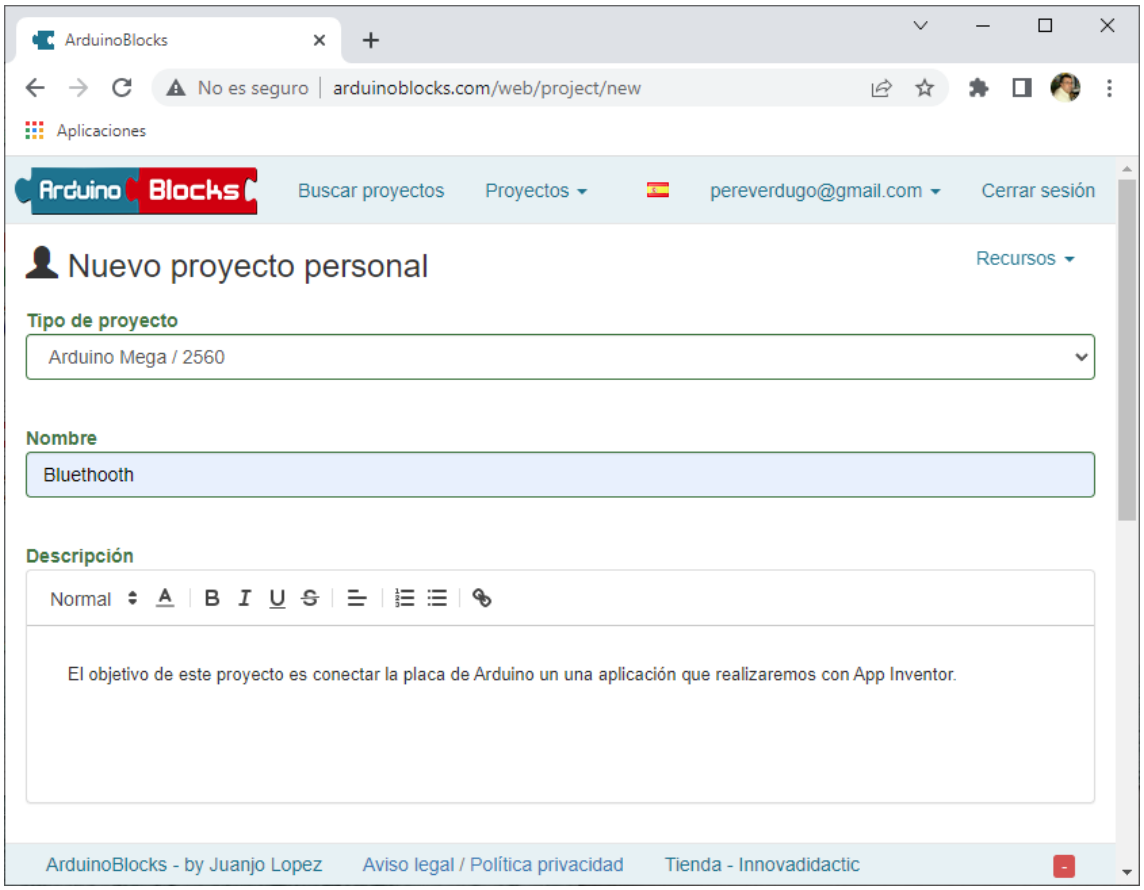


Vamos a seleccionar “Empezar un nuevo proyecto”.



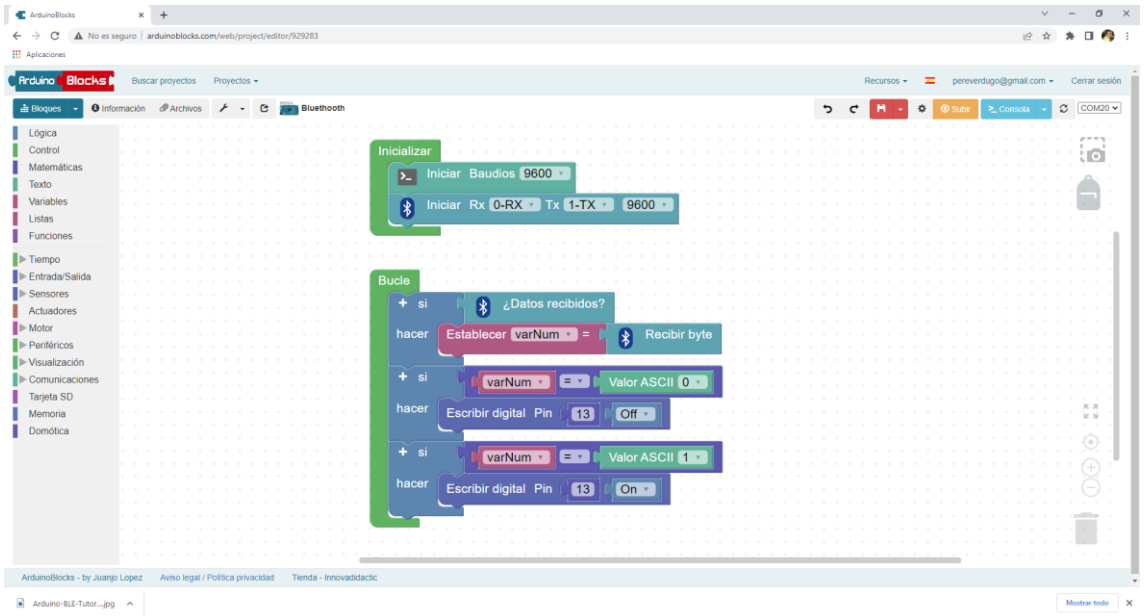
Seleccionaremos Proyecto personal.

Para poder programar desde esta página web que tendrás que registrar para poder acceder.



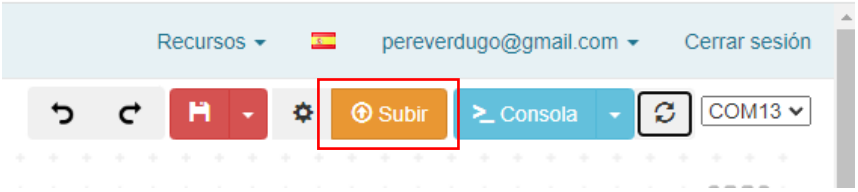
Al final de esta página seleccionaremos “Nuevo proyecto”.

Este será la programación por bloques:



En la parte superior derecha ya ha reconocido el puerto.

El valor ASCII de 0 es 48 y el valor ASCII de 1 es 49, lo tenemos que tener en cuenta en la programación de App Inventor.



Seleccionaremos el botón Subir, de este modo programaremos la placa de Arduino.

Ahora seguimos con App Inventor.

¿Qué es Bluetooth?

Mucha gente puede tener la impresión de que el Bluetooth es una tecnología anticuada, que se usaba para transmitir datos entre dispositivos, y que actualmente está en desuso. Nada más lejos de la realidad.

Bluetooth tiene la enorme ventaja de estar integrado de fábrica en la mayoría de dispositivos. Portátiles, Tablets, Smartphones llevan integrado Bluetooth. Además, su uso es independiente del sistema operativo (Windows, Linux, Mac o Android).

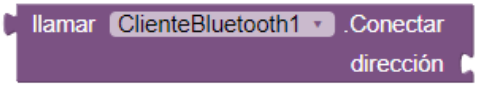
Comunicar App Inventor y Arduino por Bluetooth

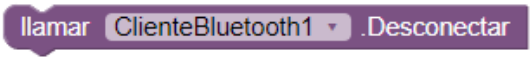
Vamos a utilizar la programación que nos ofrece App Inventor de acceder a la funcionalidad Bluetooth de nuestro dispositivo móvil para comunicarse con el módulo Bluetooth que podemos conectar a Arduino, de forma que ambos dispositivos puedan intercambiar información a través de Bluetooth.

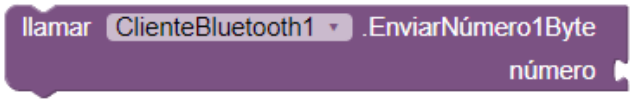
El objetivo de nuestra práctica será poder encender y apagar el LED INTEGRADO de Arduino a través del programa que realizaremos en App Inventor.

App Inventor Bluetooth

App Inventor integra bastantes bloques para trabajar con Bluetooth para nuestra práctica solo será necesarios los siguientes bloques.

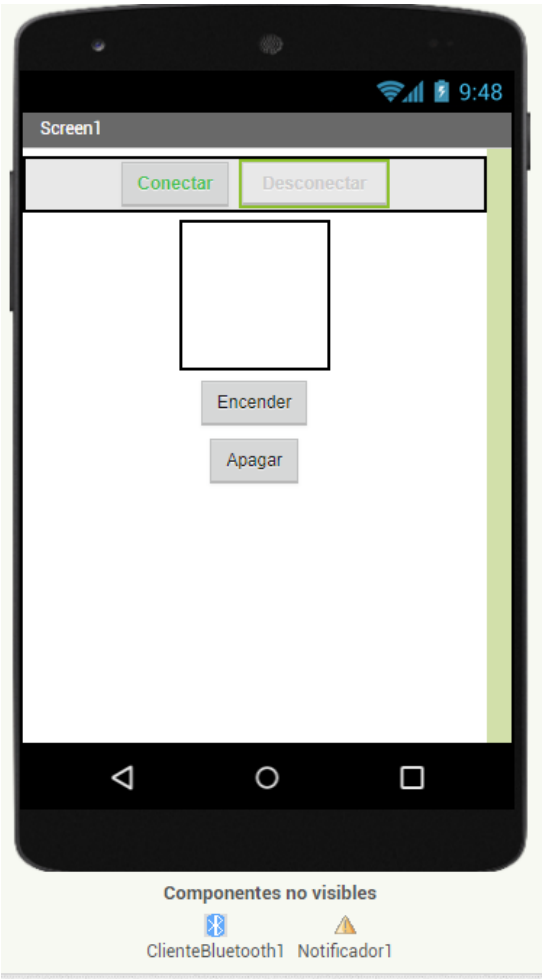




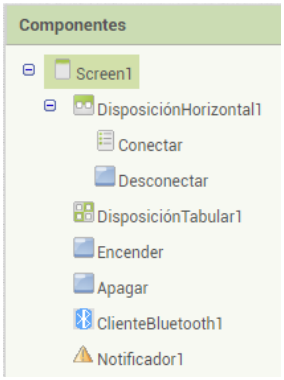


Estos serán los bloques que vamos a utilizar.

Este será el diseño:

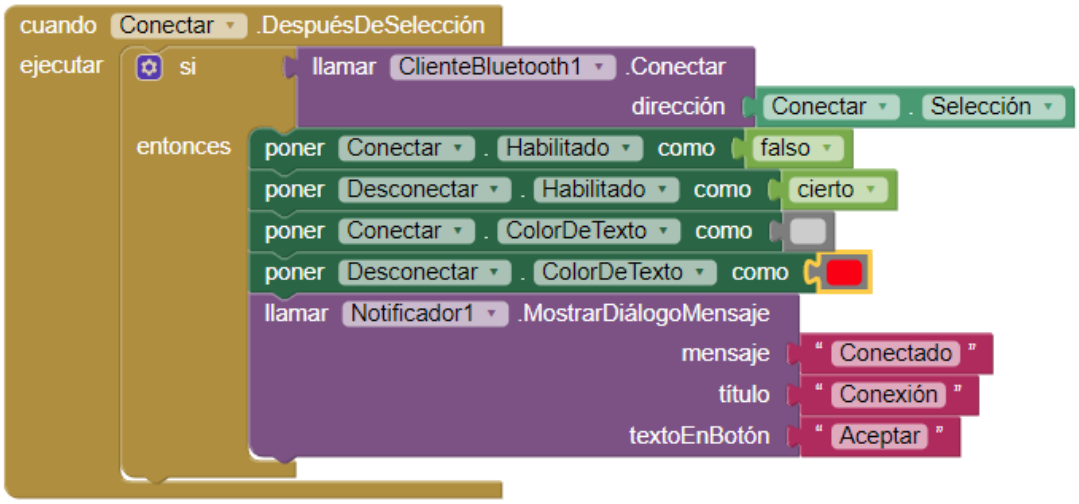


Y esta la estructura de los componentes:



Vamos a programar los bloques:

Cuando Conectar.AntesDeSelección la lista almacena una serie de dispositivos de detecta su Bluetooth.



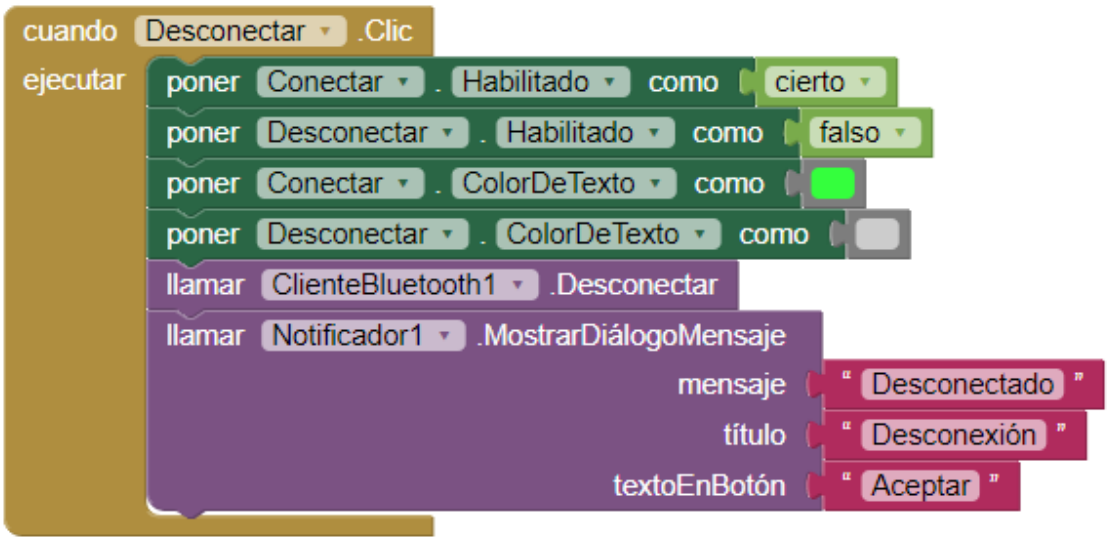
Cuando seleccione el elemento lista me mostrará los dispositivos que ha encontrado, yo tengo que seleccionar el Bluetooth que tengo conectado en la tarjeta Arduino.

A continuación el botón Conectar lo inhabilita y Habilita el botón Desconectar.

El color del texto del botón Conectar lo pone de color gris y el botón Desconectar lo pone de color rojo.

Y muestra un mensaje diciendo que ya estamos conectados.

Después de seleccionar



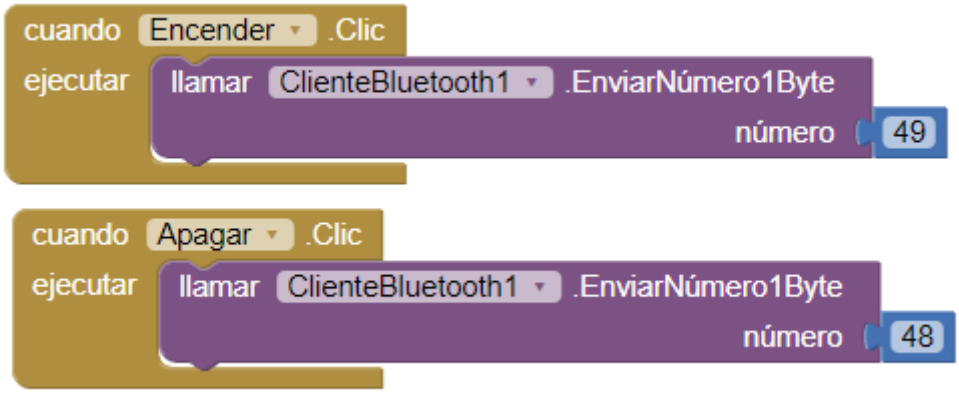
En el momento que hacemos clic en el botón Desconectar.

Habilitamos el botón Conectar y deshabilitamos el botón Desconectar.

Cambiamos el texto del botón Conectar a color verde y el botón Desconectar a color gris.

Realizamos la desconexión del Bluetooth.

Por último nos aparece un menaje de que estamos desconectados.



Al hacer clic en el botón Encender enviamos el valor 49 en ASCII que el Bluetooth lo recibirá como el valor 1.

Al hacer clic en el botón Apagar enviamos el valor 48 en ASCII que el Bluetooth lo recibirá como el valor 0.

Prueba de comunicación

Recuerda que antes de nada tienes que vincular tu móvil con el Bluetooth.

Hemos de cargar la aplicación para Arduino, que ya lo hemos comentado.

Vamos a ejecutar la aplicación que hemos realizado con App Inventor.

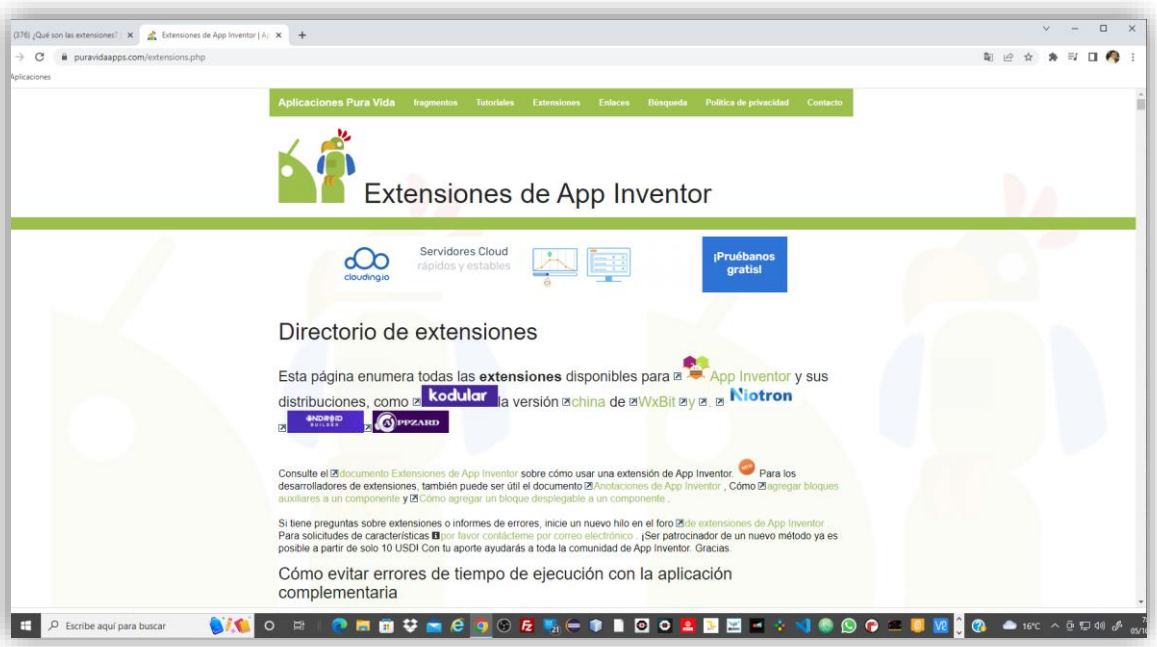

18:55
75%
Screen1
Conectar
Encender
Apagar


type in the 6-character code
-or-
scan the QR code
connect with code
scan QR code
UNO

13.- Trabajando con extensiones

Las extensiones son herramientas que nos ayudarán a realizar otras acciones que sin ellas no podríamos realizar, en el siguiente enlace vamos a encontrar un conjunto de extensiones:

https://puravidaapps.com/extensions.php

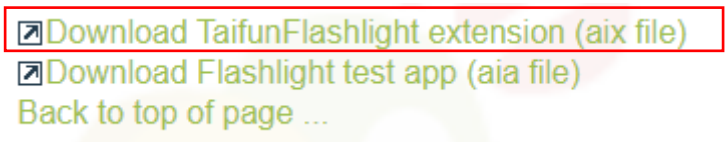


Si tu ya conoces un poco el funcionamiento de App Inventor sabrás que no hay ninguna acción que nos permita encender y apagar la linterna de nuestro móvil.

Pues en esta página he encontrado una extensión que nos lo permitirá.

Linterna Extensión para encender/apagar la linterna.

En la parte inferior encontraremos un enlace para su descarga.

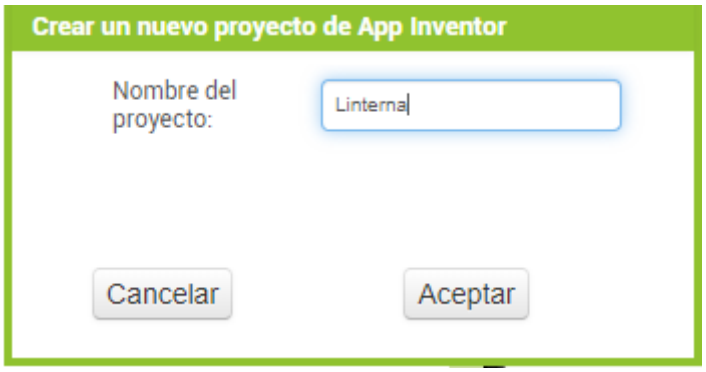


Ya lo hemos descargado.

com.puravidaapps.TaifunFlashlight.aix

Ahora vamos a App Inventor.

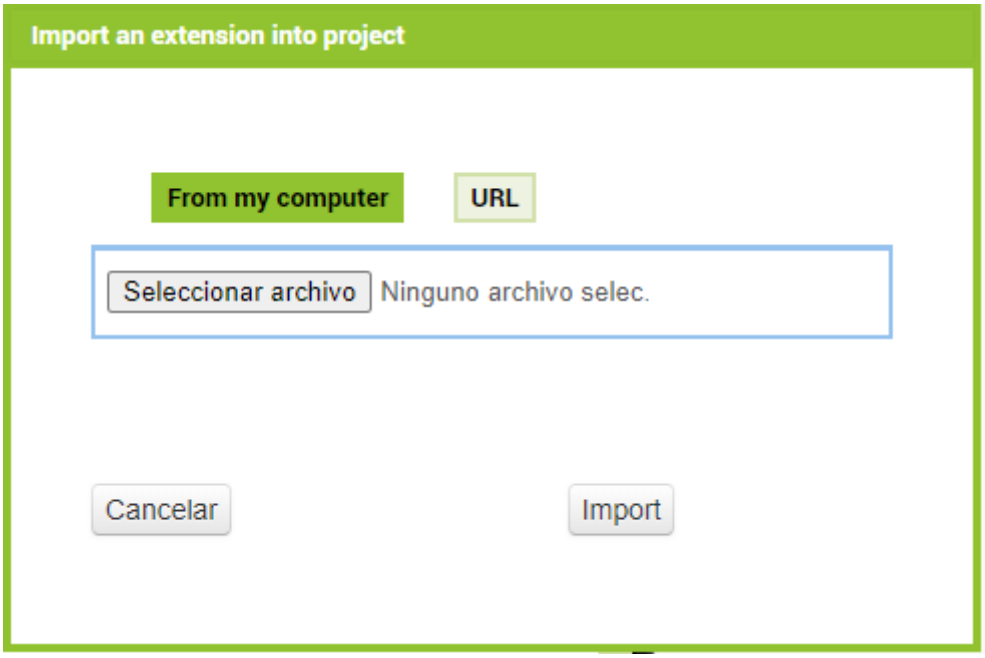
Empezamos un nuevo proyecto llamado Linterna.



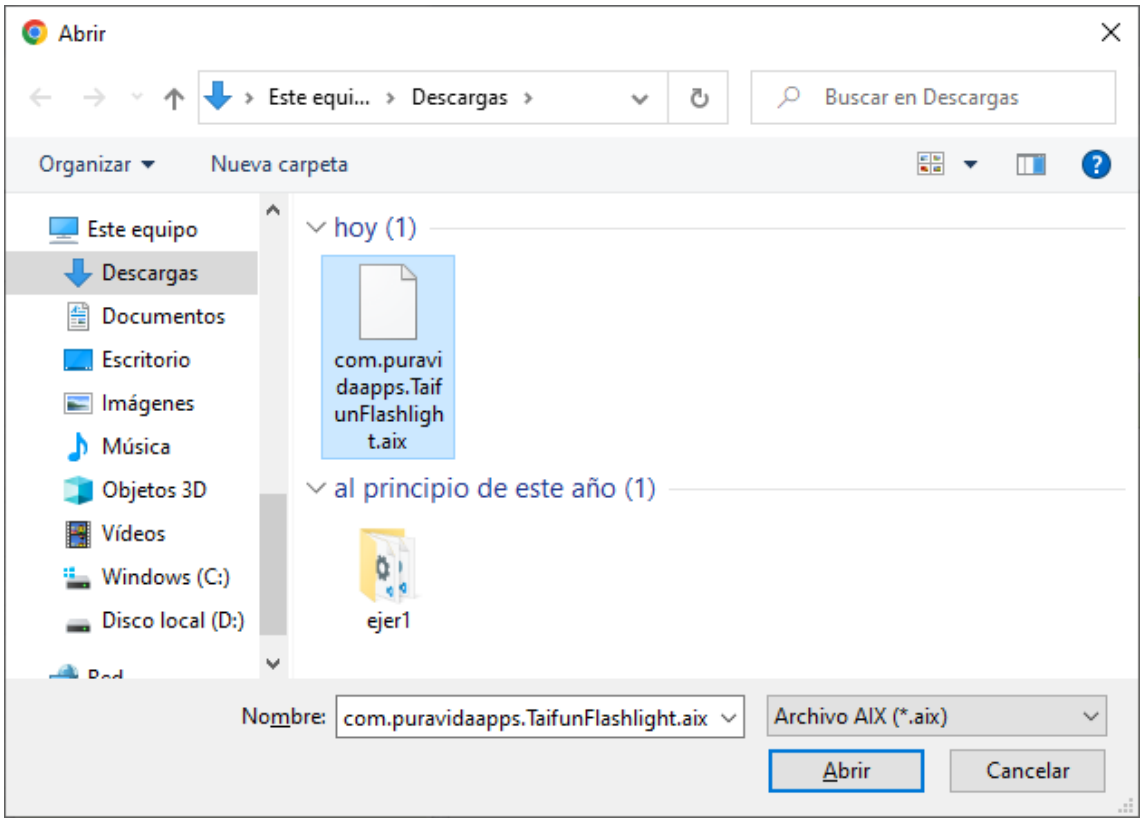
En el apartado de interfaz de usuario en la parte inferior vamos a encontrar Extensión.

La seleccionamos.

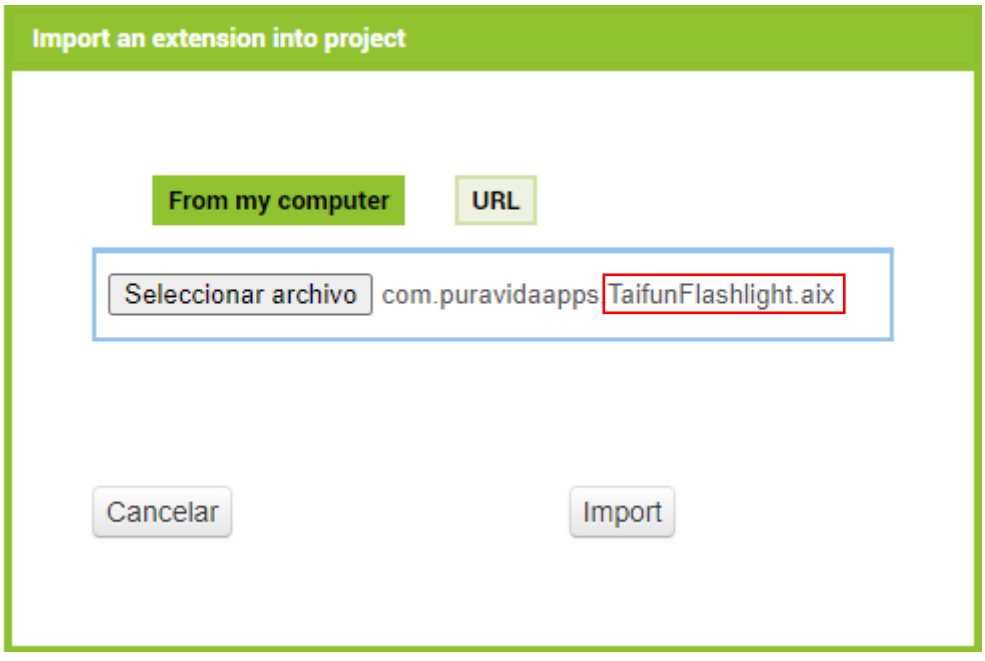
Se nos abre una opción para poder importar la extensión.



Vamos a seleccionar el archivo que descargamos con anterioridad.

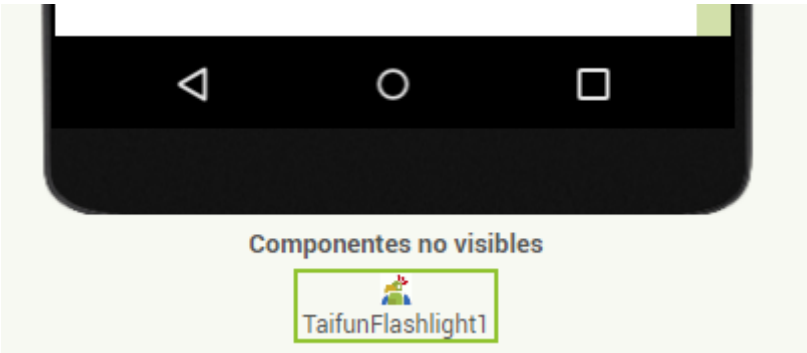


Una vez seleccionado le damos al botón Abrir.

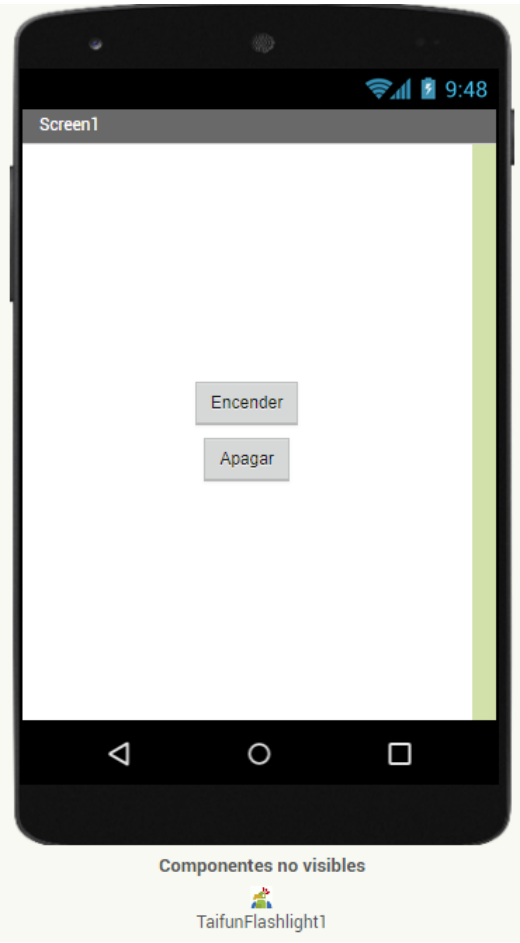


Seguido del botón Import.

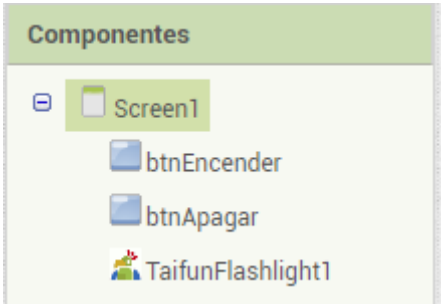
Ya lo tenemos ahora para tener acceso al el lo vamos a arrastrar hacia el visor, como agregamos cualquier componente, de este modo tendremos acceso a los bloques en modo de progración.



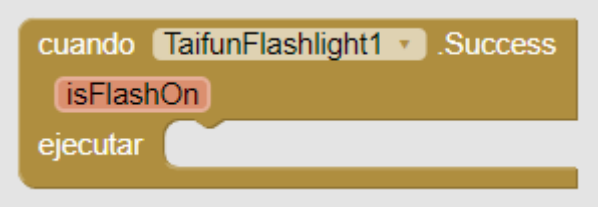
Vamos a agregar dos botones una para encender y otro para apagar.



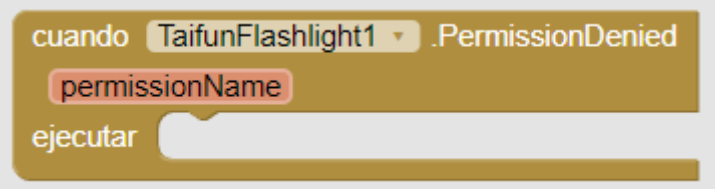
Esta es tu estructura:



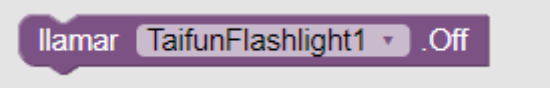
Vamos a programación de bloques:



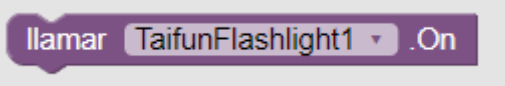
Enciende la linterna.



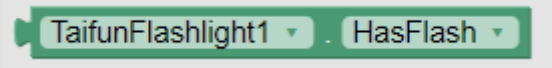
Evento que indica que se ha denegado el permiso.



Apaga la linterna.

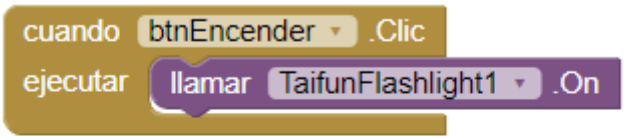


Enciende la linterna.



Devuelve si el dispositivo tiene flash.

Este será el código por bloques:

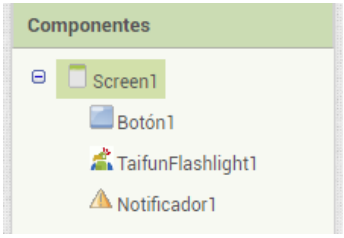


Muy fácil, vamos a realizarlo un poco más complejo.

Vamos a buscar por internet dos bombillas una encendida y otra apagada.

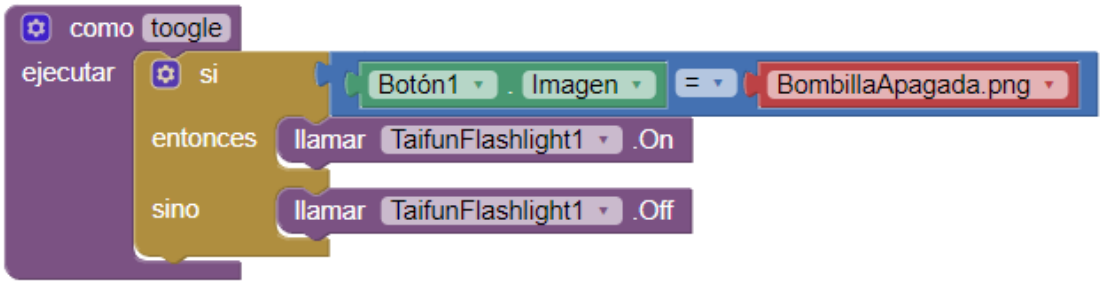
Vamos con el anterior proyecto que vamos a modificar.

Esta es su estructura:



Agregamos un botón con la imagen de la bombilla apagada.

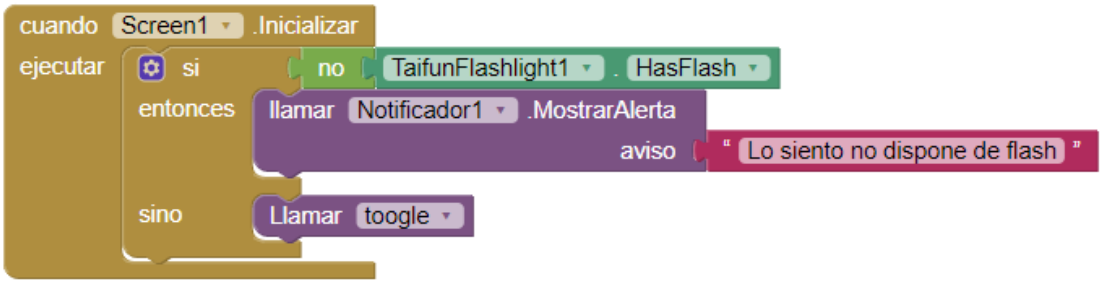
Ahora vamos a los bloques:



Creamos un procedimiento porque este será llamado más de una vez y para no repetir el código lo hacemos una vez y lo utilizaremos las veces que sea necesario.

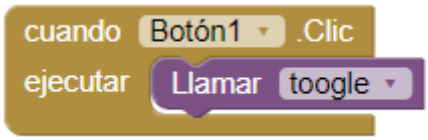
Realiza una comparación si la imagen del botón es “BombillaApagada.png”

Si es así enciende el flash de lo contrario lo apaga.

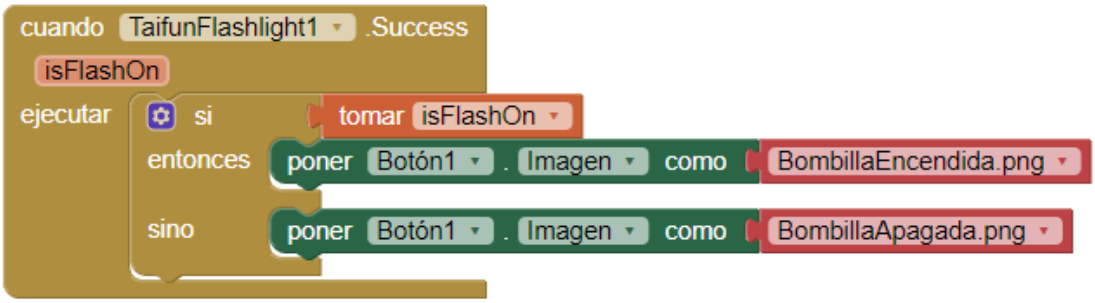


Al empezar comprueba si el dispositivo no tiene flash, de ser así nos envía un mensaje diciéndome “Lo siento no dispone de flash.

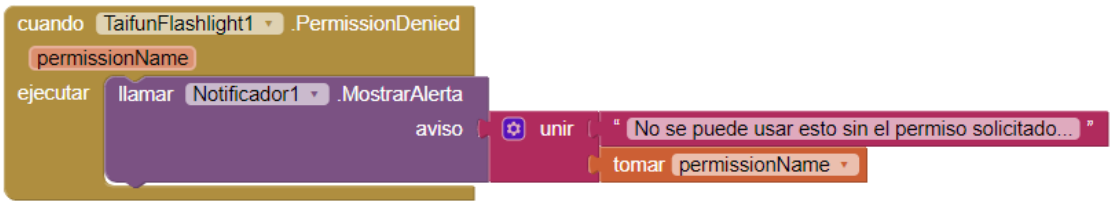
De lo contrario si tiene flash llama al procedimiento toogle.



Cuando hacemos clic sobre el botón también llama al procedimiento toogle.



Cuando se enciende el flash muestra el botón con la imagen BonbillaEncendida.png de lo contrario muestra la imagen BonbillaApagada.png.



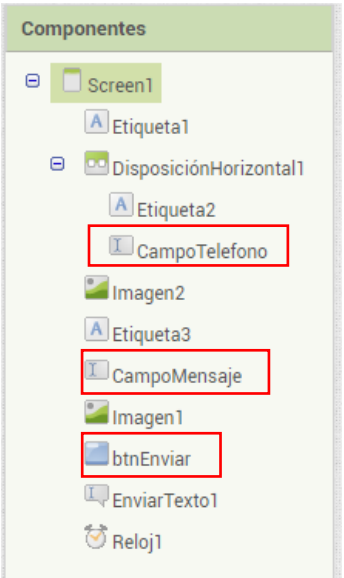
Si no tienes permiso del usuario mostrará la siguiente mensaje “No se puede usar esto sin permiso solicitado… CAMARA”

14.- Enviar SMS

Vamos a realizar un proyecto para que desde nuestra App puedas enviar un mensaje a otro móvil.

Vamos a realizar el siguiente diseño:

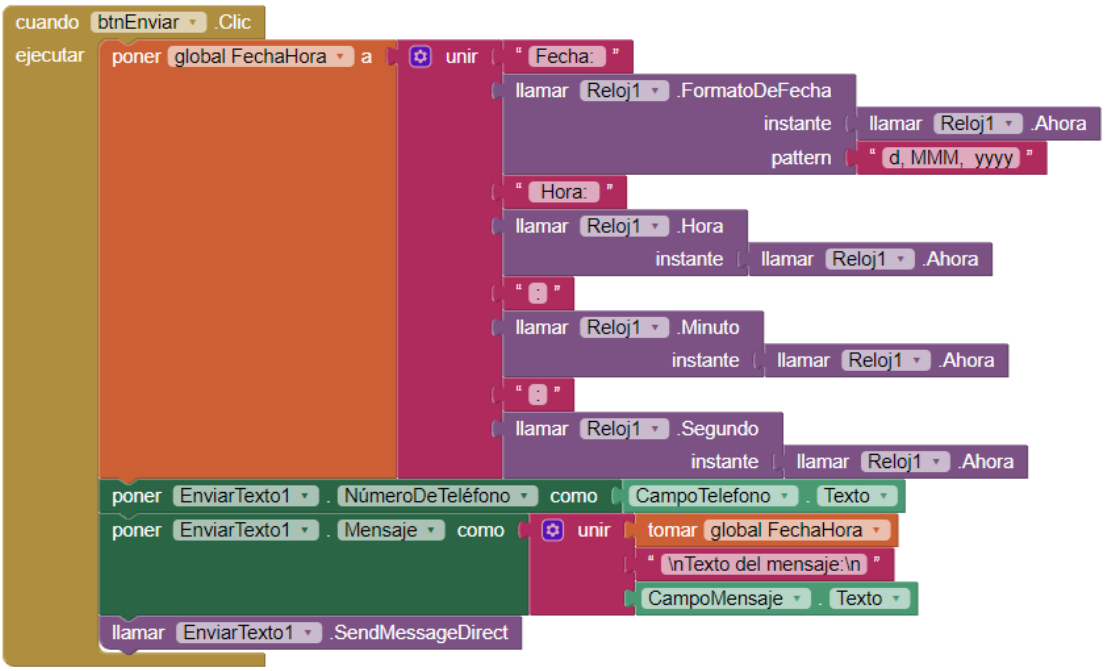
Como ya eres un experto no tengo que explicarte como he realizado este diseño pero te paso las



Para que puedas ver los componentes y si alguno le he cambiado el nombre para su mejor identificación:

Ahora vamos a la programación en bloques:

Definimos una variable local con el nombre de FechaHora y como valor vacío.



Cuando hacemos clic sobre el botón btnEnviar:

A la variable FechaHora le asignamos la Fecha y Hora, para que quede reflejado la fecha de cuando lo ha recibido.

En el objeto EnviarTexto1.NúmeroDeTeléfono le asignamos el valor del CampoTelefono.

A continuación mandamos un texto concatenado con la Fecha y la hora un título “Texto del mensaje” cuando agregamos \n Estamos diciendo que queremos un salto de línea y por último el contenido del mensaje de texto.

Para poder probar esta aplicación es necesario que la instales primero ya que te tiene que pedir permiso para poder enviar el mensaje.

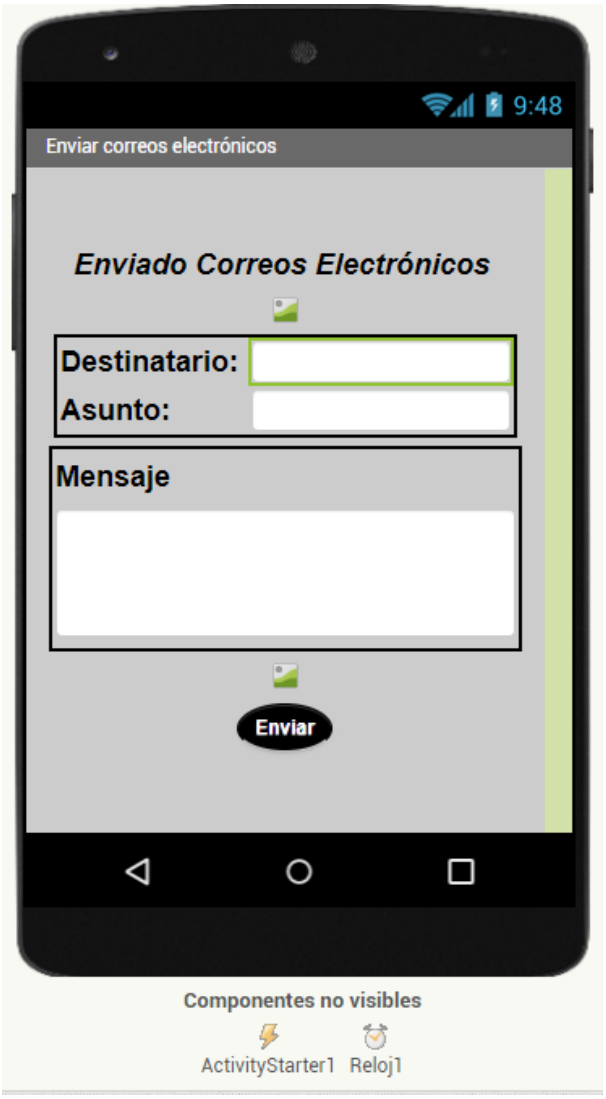
Mira que gastar un SMS solo para decir hola, bueno he comprobado que funciona.

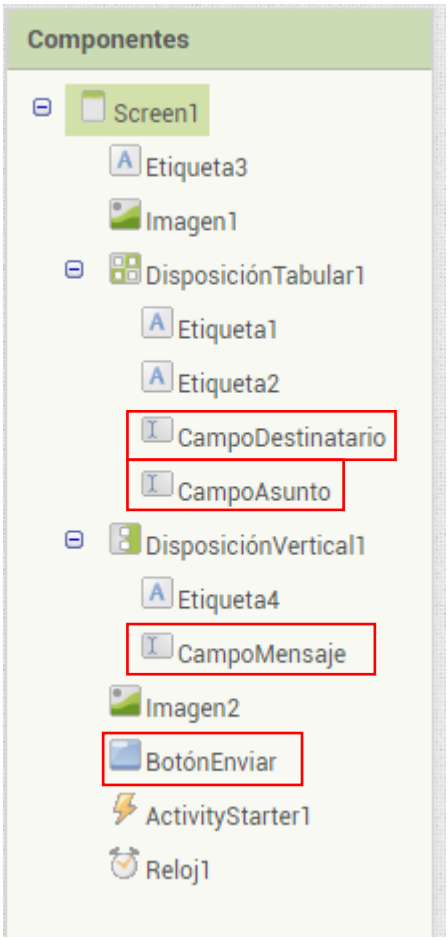
15.- Enviar correos electrónicos.

Ahora queremos crear una App que nos permita enviar correos electrónicos.

Desde un nuevo proyecto que llamaremos Correo.

Este será el diseño:



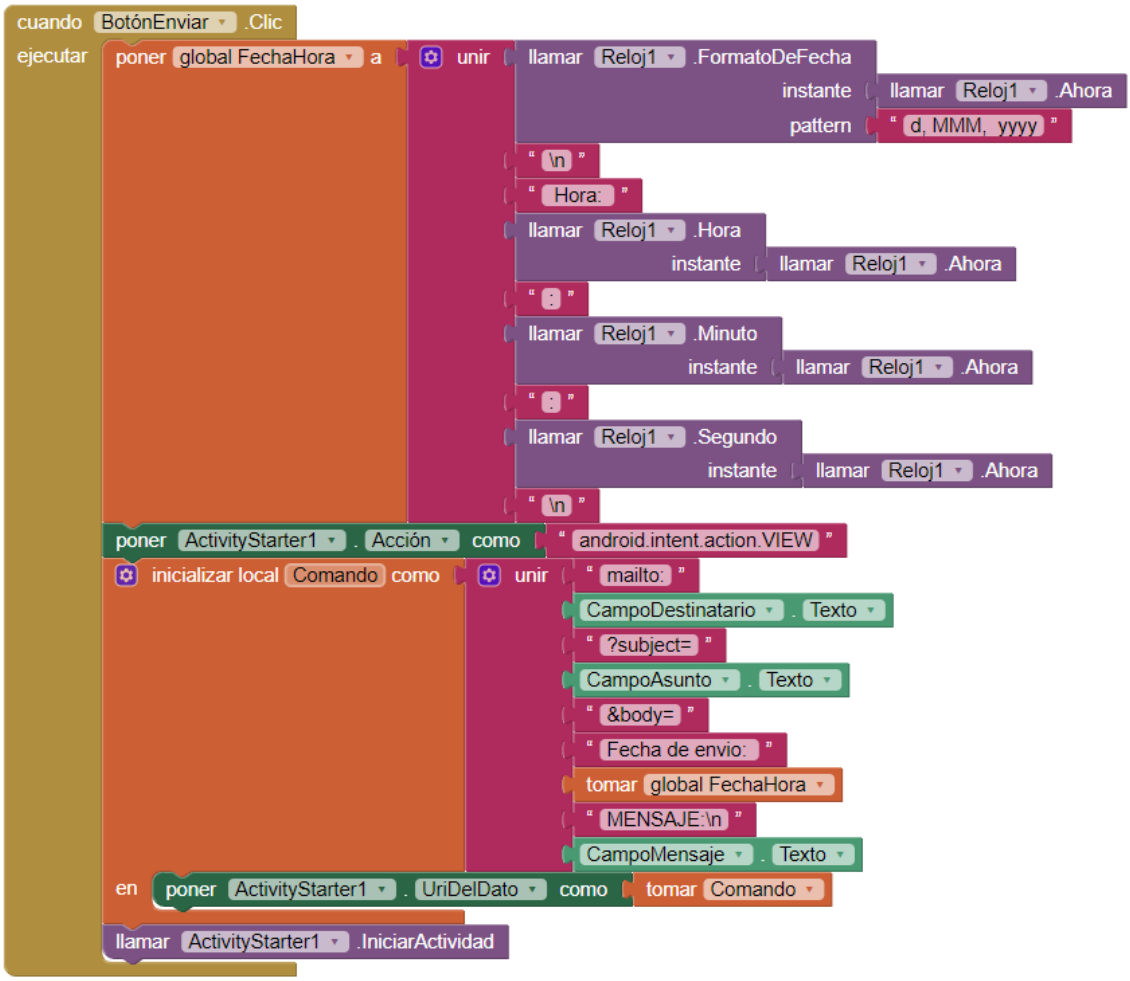


El elemento ActivityStarter1 se encuentra en el grupo de Conectividad.

En la estructura, donde podrás observar a que elementos les he cambiado el nombre:

Ahora vamos a la programación por bloques:

Inicializamos la variable global FechaHora con un valor de vacío.



Al hacer clic en el BotónEnviar lo primero es asignar a la variable FechaHora el valor de la fecha y la hora actual.

Con el elemento ActivityStarter1.Accion le decimos el tipo de acción que queremos, para no profundizar limítate a copiarlo.

Declaramos una variable local Comando que en ella concatenamos una serie de instrucciones agregando además los campos de CampoDestinatario.Texto, CampoAsunto.Texto y CampoMensaje.Texto que solo los campos que rellenamos antes de enviar el correo.

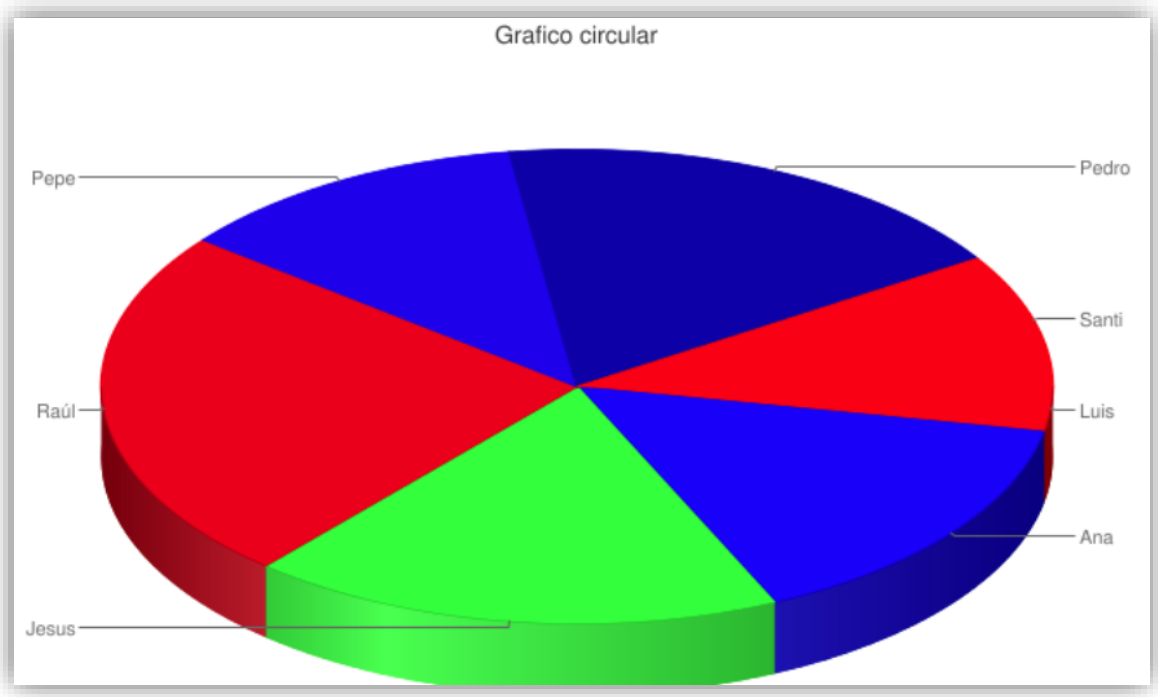
Totas estas instrucciones se la asignamos al elemento ActivityStarter1.UriDelDato.

Por último ejecutamos estas instrucciones.

Ahora cuando ejecutes la aplicación y envíes el formulario se te abrirá tu correo en el móvil con toda la información introducida solo tendrás que dar al botón Enviar de tu correo.

16.- Gráficos con App Inventor

¿Cómo mostrar un gráfico con Google chart api?



Introducción

Google chart, es una API de Google que nos permite escribir una línea de dirección web, con cierto formato, y nos devuelve un gráfico de tipo estadístico, barras, tarta, trazos, … y también lo podemos utilizar para que nos dibuje una función.

Estructura de los gráficos

Para crear un gráfico de barras con la api de Google chart necesitamos generar una dirección web como la siguiente.

https://chart.apis.google.com/chart?cht=bvg&chxt=y&chs=250x400&chco=A2C180&chtt=Grafico+de+barras&chd=t:10,50,60,80,40,60,30&chl=Luis|Ana|Jesus|Raúl|Pepe|Pedro|Santi&chco=FF0000|0000FF|00FF00|F0000F|0F00F0|0000AA

Vamos a desglosar cada uno de los términos.

https://chart.apis.google.com/chart?cht=bvg&chxt=y&chs=250x400&chco=A2C180&chtt=Grafico+de+barras&chd=t:10,50,60,80,40,60,30&chl=Luis|Ana|Jesus|Raúl|Pepe|Pedro|Santi&chco=FF0000|0000FF|00FF00|F0000F|0F00F0|0000AA

Anatomía de la dirección web

https://chart.apis.google.com/chart? Es la dirección web que se utiliza para llamar a la Api de Google indicando al final que es un gráfico y que se le va a mandar una serie de parámetros.

Cht=bvg cht=Indica el tipo de gráfico a mostrar.

- cht=bvg : Mostrará un gráfico de barras verticales.
- cht=bhs : Mostrará un gráfico de barras horizontales.
- cht=lc : Mostrará un gráfico de líneas.

- cht=ls : Mostrará un gráfico de tipo Palabra gráfica.
- cht=p : Mostrará un gráfico circular.
- cht=p3 : Mostrará un gráfico circular con efecto en 3D.
- cht=gom: Mostrará un gráfico simulando un indicador métrico.
- cht=v : Mostrará un gráfico de tipo diagrama de Ven.
- cht=s : Mostrará un gráfico de tipo puntos.
- cht=r : Mostrará un gráfico de tipo radar.
- cht=t : Mostrará un gráfico de tipo mapa (necesita parámetros).

chxt=y . De forma predeterminada, los valores de los ejes oscilan entre o y 100, a menos que la escala explícitamente con la chxr propiedad. Para ocultar todas las líneas de eje en un gráfico de líneas, especifique :nda después del valor del tipo de gráfico en el cht parámetro (ejemplo: cht=lc:nda).

De forma predeterminada, los ejes superior e inferior no muestran marcas de verificación por los valores, mientras que los ejes izquierdo y derecho sí los muestran.

Puede cambiar este comportamiento utilizando el chxs parámetro.

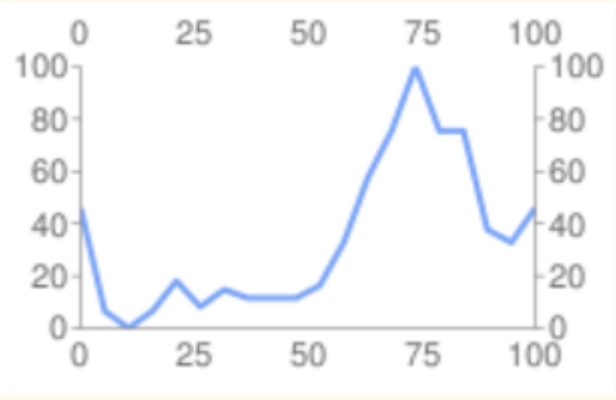
Un eje para mostrar en el gráfico. Los ejes disponibles son:

- x- Eje x inferior
- t- Eje x superior [No es compatible con Google-o-Meter]
- y- Eje y izquierdo
- r- Eje y derecho [No es compatible con Google-o-Meter]

Se puede especificar varios ejes del mismo tipo, por ejemplo: cht=x,x,y. Esto apila dos conjuntos de ejes x a lo largo de la parte inferior del gráfico. Esto es útil al agregar etiquetas personalizadas a lo largo de un eje que muestra valores numéricos (consulte el ejemplo a continuación). Los ejes se dibujan de adentro hacia afuera, por lo que si tienen x,x, la primera x se refiere a la copia más interna, la siguiente x se refiere a la siguiente copia hacia afuera, y así sucesivamente.

Ejemplos

Este ejemplo muestra un gráfico de líneas con un eje x, un eje y, un eje superior (t) y un eje derecho (r).



chxt=x,y,r,t

Debido a que no se especifican etiquetas, el gráfico tiene un rango predeterminado de 0 a 100 para todos los ejes.

& Cada vez que tengamos que unir cada uno de los parámetros que pasamos a la dirección Web tendremos que usar este símbolo & ampersand.

chs=400x400 Tamaño de la imagen en píxeles (ancho x alto). Existen varias restricciones respecto a los valores de este parámetro. El área establecida no puede superar los 300.000 píxeles y el ancho/alto de la imagen no puede ser mayor que 1.000 píxeles.

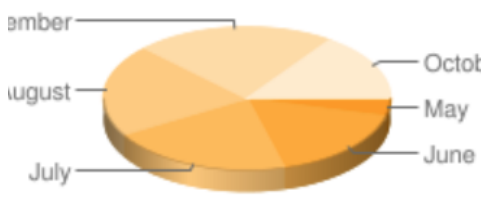
chtt=Vertical+bar+chart Título que mostrar el gráfico. Este título tiene que contener el signo + en cada uno de los espacios que contenga.

chd=t:10,50,60,80,40,60,30 Datos del gráfico. Se podrán definir varios conjuntos de datos para un determinado gráfico separados por el carácter "|". Estos datos deben estar codificados de la siguiente forma:

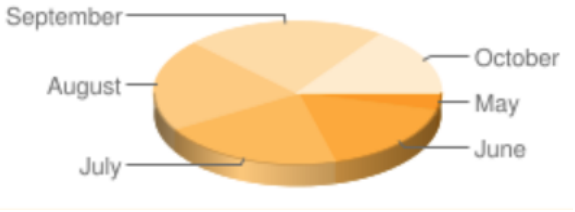
chl=Luis|Ana|Jesús Un texto para aplicar a cada segmento. Las etiquetas se aplican consecutivamente a los puntos de datos en chd. Si tienes varias series (para un gráfico circular concéntrico, por ejemplo), las etiquetas se aplican a todos los puntos en todas las secuencias, en el orden especificado en chd. Use un delimitador de barra (|) entre cada etiqueta. Para especificar un valor intermedio vacío utiliza dos caracteres consecutivos de barras sin espacio entre ellas: ||.

Ejemplos

Etiquetas para un gráfico circular tridimensional.



chl=May|June|July|August|September|Octuber
chs=220x100



chl=May|June|July|August|September|Octuber
chs=280x100

Al especificar el tamaño de un gráfico chs, mira cuánto espacio necesitarán tus etiquetas.

En general, un gráfico circular bidimensional debe tener aproximadamente el doble de ancho que alto, y un gráfico circular tridimensional debe ser aproximadamente dos vedes y media más ancho que alto, para mostrar las etiquetas correctamente.

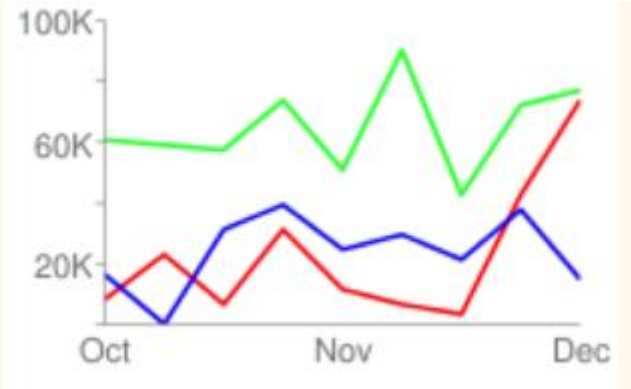
En el primer ejemplo, las etiquetas se muestran cortadas, porque el gráfico no es lo suficientemente ancho.

El segundo ejemplo muestra que el gráfico necesita un ancho de 280 píxeles para mostrar las etiquetas en su totalidad.

Puede especificar los colores de una serie específica, o todos los elementos de una serie, utilizando el chco parámetro. La sintaxis y el significado exactos pueden variar según el tipo de gráfico; vea su tipo de gráfico específico para más detalles.

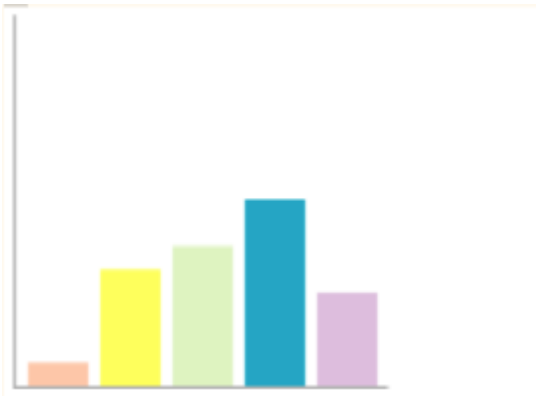
Ejemplos

Cuando especifica un solo color para cada serie en un gráfico de líneas, a cada línea se le asigna el color correspondiente. Este ejemplo tiene tres series de datos y tres colores.



chco=FF0000,00FF00,0000FF

Este ejemplo muestra cómo especificar un color para cada miembros individualmente de una serie (barras en este ejemplo).

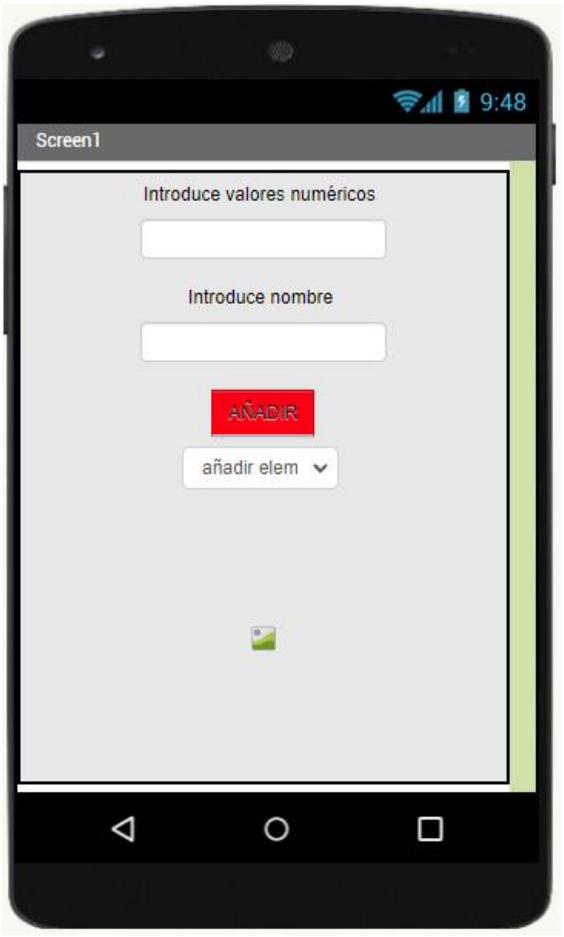


cht=bvs

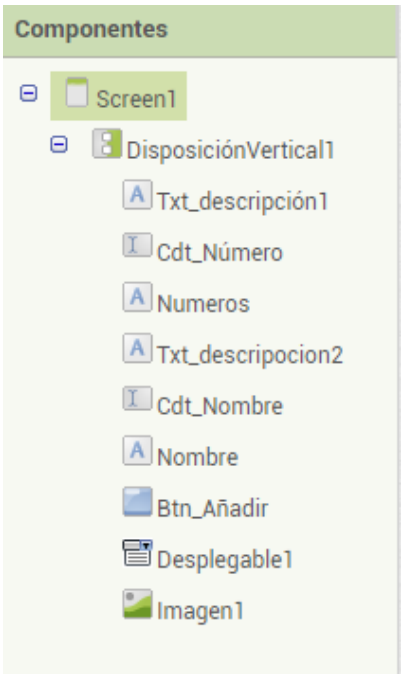
chco=FFC6A5|FFFF42|DEF3Bd|00A5C6|DEBDDE

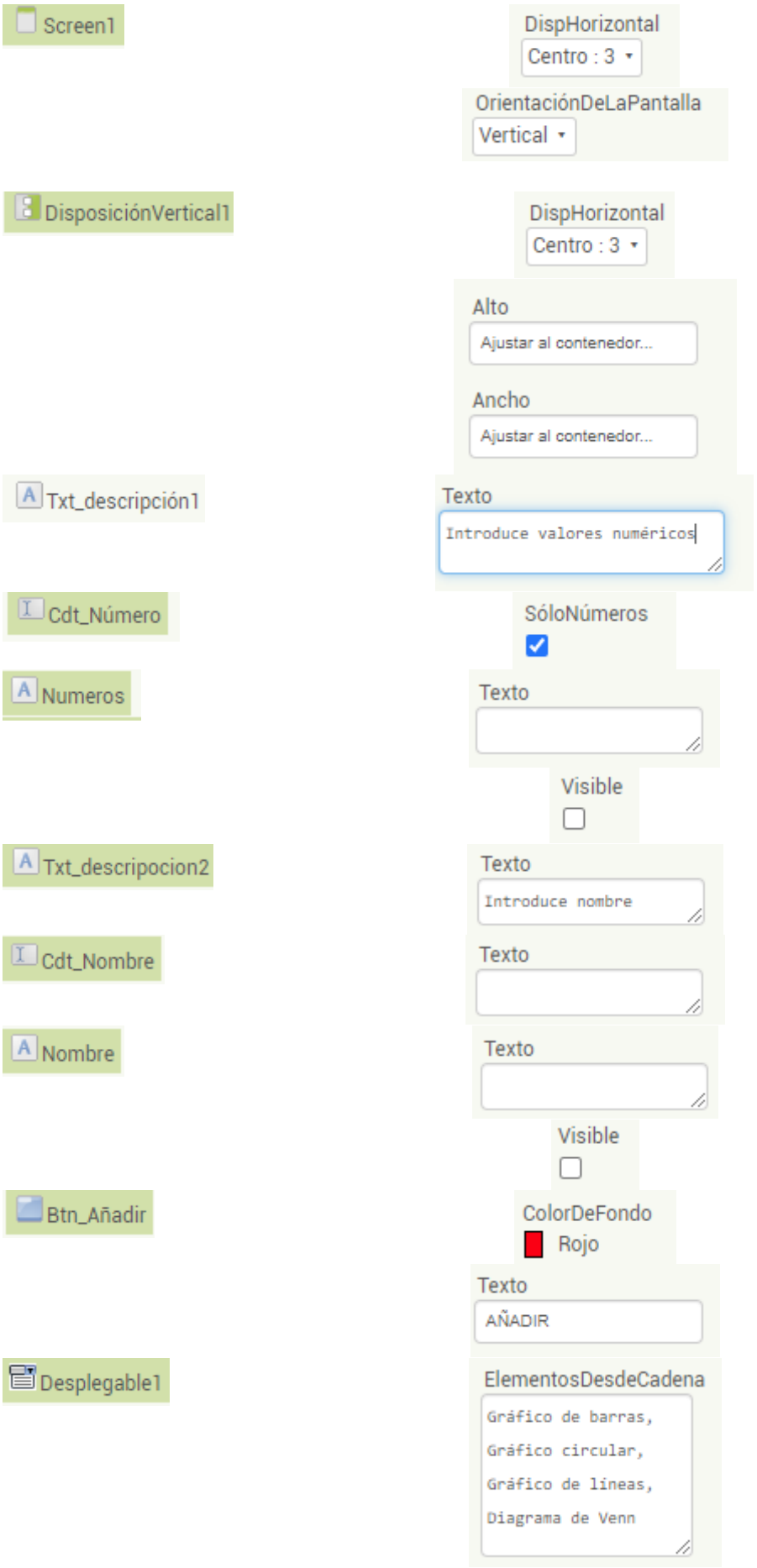
Vamos a empezar con un nuevo proyecto llamado Gráficos.

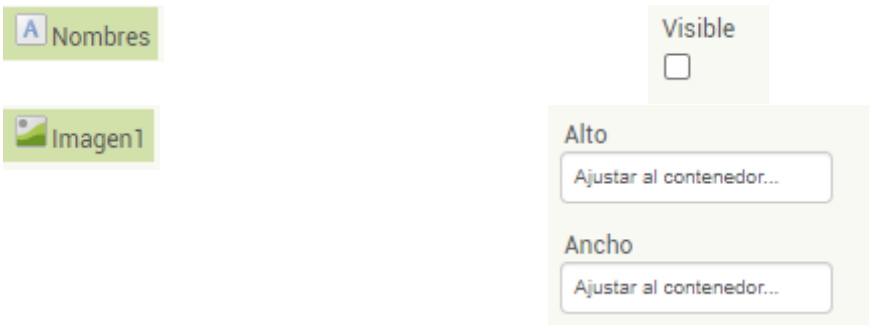
Este será el diseño:



Este es la estructura de los elementos:



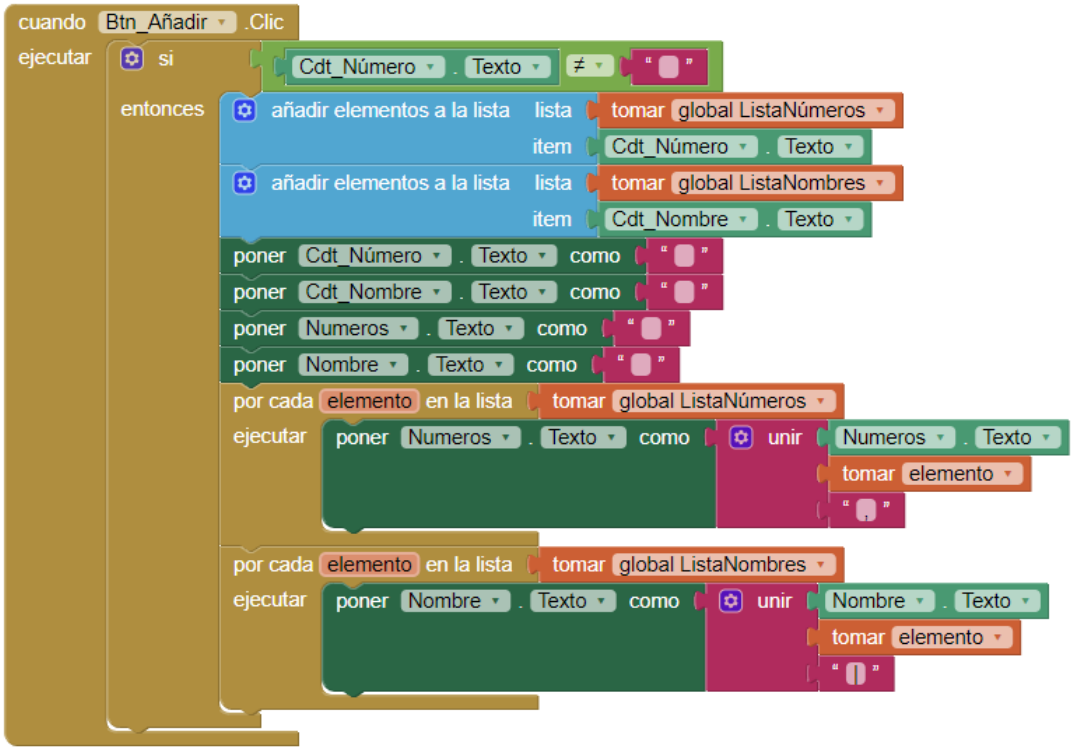

Screen1
DispHorizontal
Centro : 3
OrientaciónDeLaPantalla
Vertical
DisposiciónVertical1
DispHorizontal
Centro : 3
Alto
Ajustar al contenedor...
Ancho
Ajustar al contenedor...
Txt_descripción1
Texto
Introduce valores numéricos
Cdt_Número
SóloNúmeros
Numeros
Texto
Visible
Txt_descripocion2
Texto
Introduce nombre
Cdt_Nombre
Texto
Nombre
Texto
Visible
Btn_Añadir
ColorDeFondo
Rojo
Texto
AÑADIR
Desplegable1
ElementosDesdeCadena
Gráfico de barras,
Gráfico circular,
Gráfico de líneas,
Diagrama de Venn



Ahora vamos a la programación de bloques.

Definimos un variable global llamada ListaNúmeros de tipo lista y vacía.

Definimos un variable global llamada ListaNombres de tipo lista y vacía.



Cuando hacemos clic en el botón Btn_Añadir.

Si campo de texto Cdt_Número es distinto a vacío.

A la variable ListaNúmeros le agregamos el valor que contiene la caja de textos.

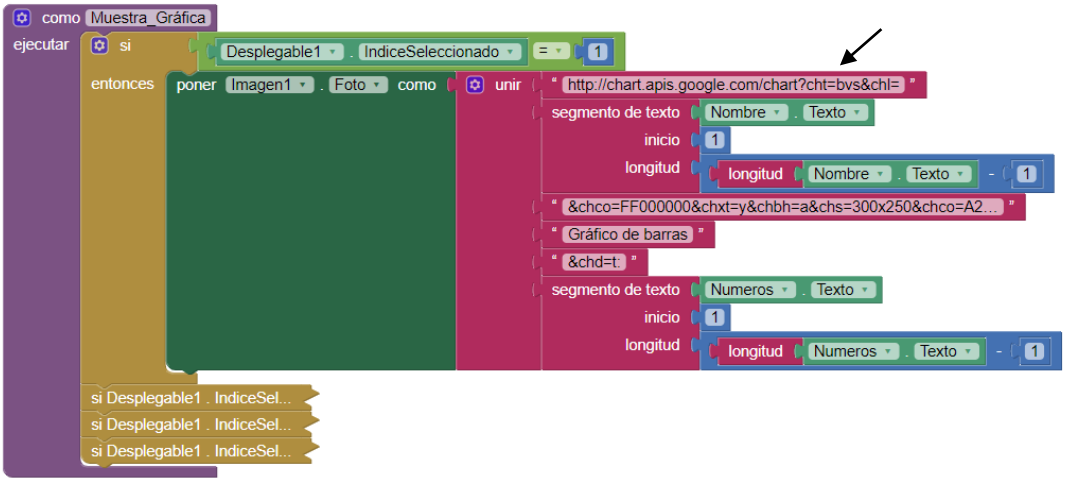
El campo de texto Cdt_Número le asignamos el valor de vacío, así como a las etiquetas Número y Nombres.

Ahora hacemos un bucle por cada elemento de la lista ListaNúmeros.

En la variable Números.Texto acumulados el último valor introducido por los anteriores, separado por una coma.

Ahora hacemos un bucle por cada elemento de la lista ListaNombres.

El la variable Nombre.Texto hacemos lo mismo pero separado por una barra diagonal.



El primer bloque del procedimiento compara si Desplegable1.IndiceSeleccionado es igual a 1, si es así a la Imagen1.foto le asignamos la siguiente Url.

https://chart.apis.google.com/chart? Es la dirección web que se utiliza para llamar a la Api de Google.

cht=bvs -> Mostrará un gráfico de barras.

Muestra los nombres que hemos introducido en el formulario, desde el principio hasta el final del mismo restándole el último carácter.

La serie de nombres se representa de la siguiente forma:

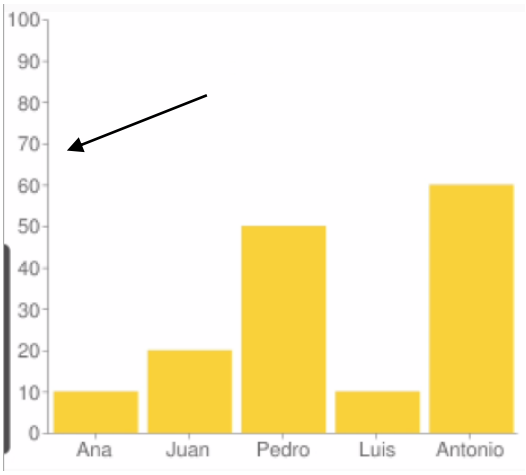
Ana|Juan|Pedro|Luis|Antonio| lo que hacemos es eliminar la última barra diagonal.

Este resultado lo igualamos a chl= Ana|Juan|Pedro|Luis|Antonio

chco=FF000000 estamos asignando un color a la gráfica.

chxt=y Especificamos en eje y

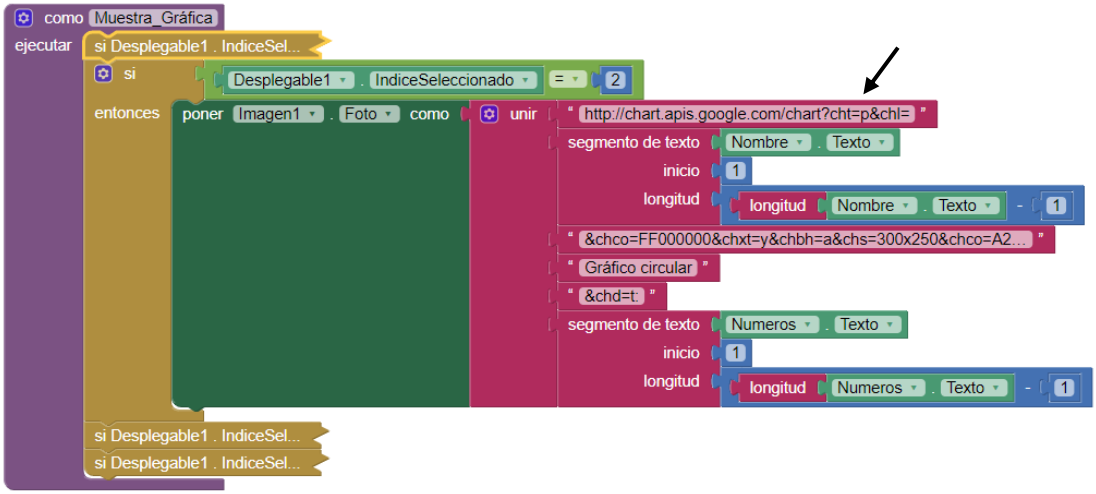
chs=300x250 Son las dimensiones del gráfico.



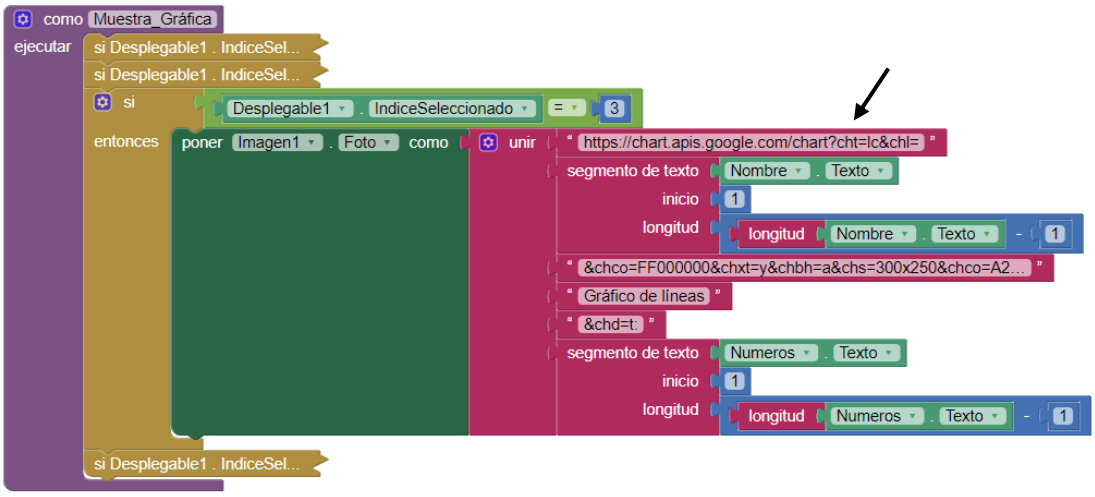
chbh=a Especifica el ancho de barra y espacio.

chco=A2… Para asignarle color

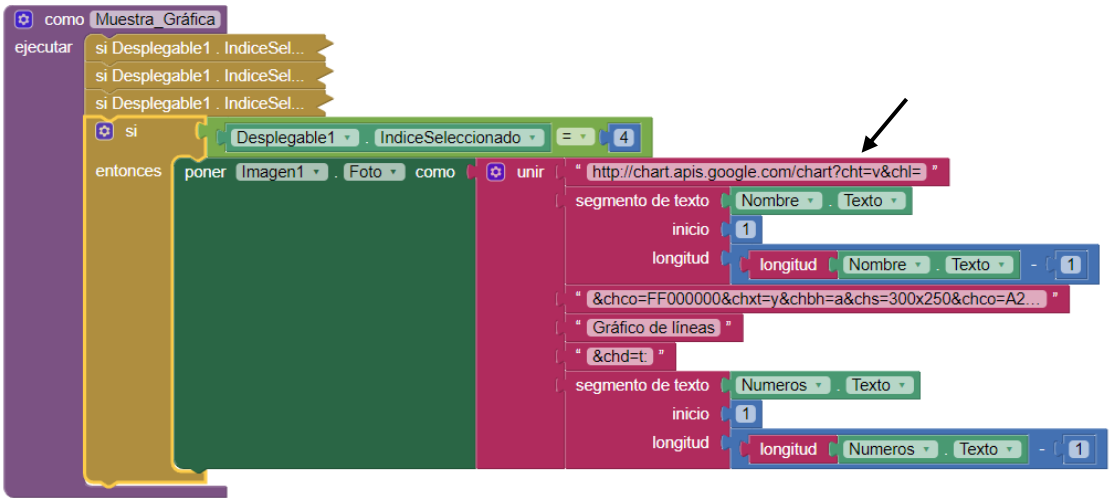
& Se utilizar para unir las instrucciones ya que no pueden haber espacios en blanco.



cht=p Muestra un gráfico circular.

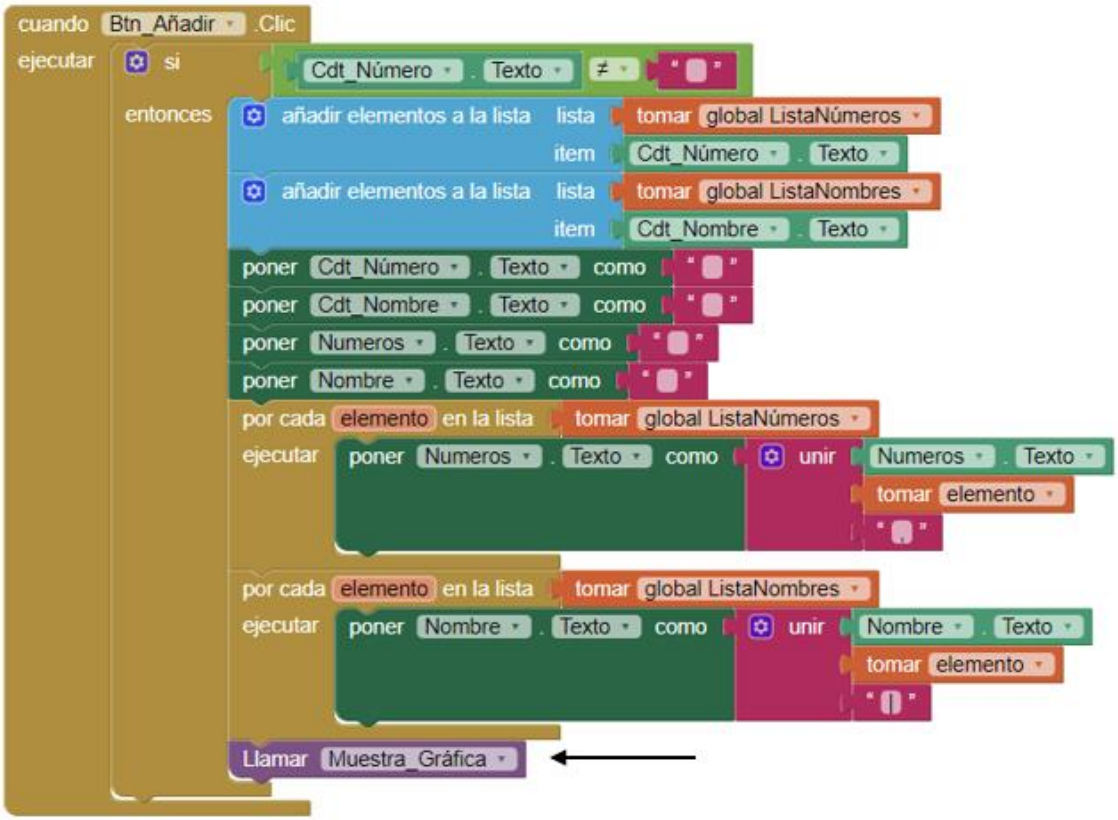


Cht=lc Muestra un gráfico de líneas.



cht=v Muestra un gráfico de tipo diagrama de Ven.

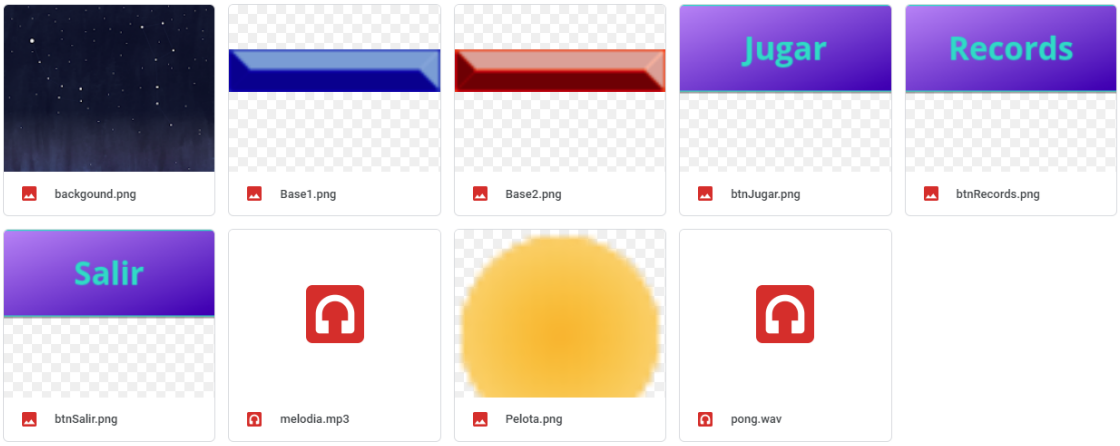
Con cuando Desplegable1.DespuésdeSeleccionar queremos que se actualice el gráfico a otro tipo.



En el BtnAñadir.Clic al final llamamos de nuevo al procedimiento “Muestra_Gráfica”

17.- Juego (Pong)

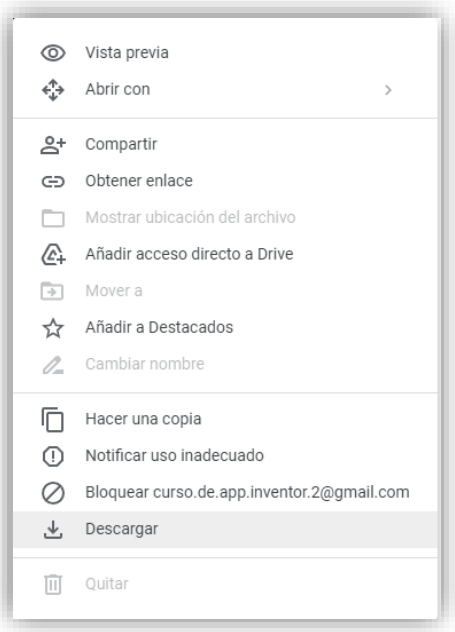
Este proyecto será un juego clásico de los 80, para ello vamos a necesitar el siguiente material:



Que podrás descargar desde el siguiente enlace:

https://drive.google.com/drive/folders/1D9gPnqHQJDHMXnMm48fyFZHXA6O_Mw1A

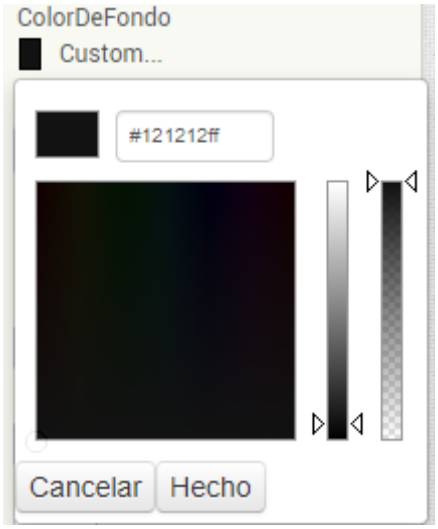
Selecciona cada objeto con el botón derecho del ratón y después del menú seleccionas:



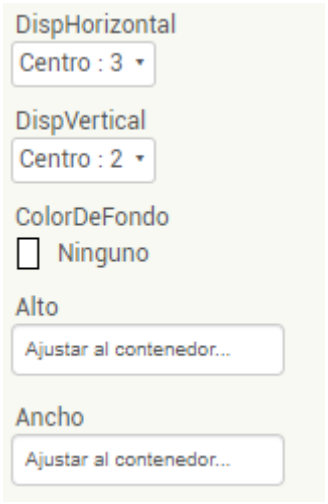
Ahora que ya tenemos todos los materiales vamos a empezar un nuevo proyecto que le vamos a llamar Pong.

Vamos a modificar las propiedades de la pantalla Screen1.

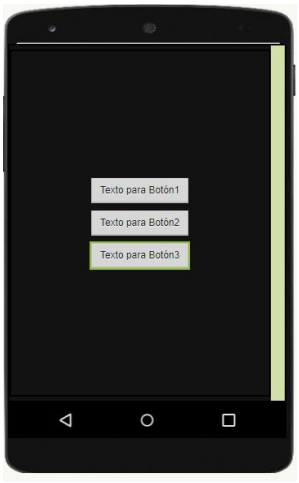
Color de fondo seleccionaremos Custom y pondremos el siguiente valor #121212ff, las dos últimas letras le decimos que lo queremos completamente opaco.



Ahora vamos a agregar una disposición vertical con las siguientes propiedades:



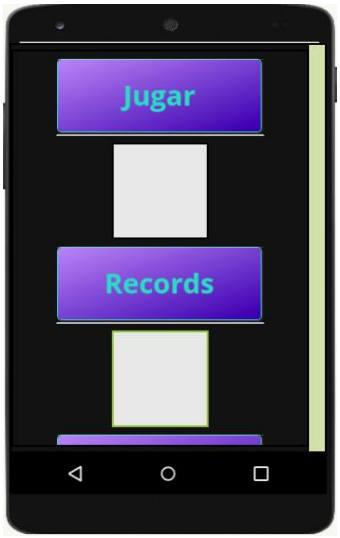
A continuación vamos a agregar 3 botones:



A cada botón le eliminamos el texto y agregamos la correspondiente imagen.

Si lo queremos ver desde nuestro dispositivo veréis que los botones están muy juntos.

Para ello vamos a insertar entre los botones disposiciones.

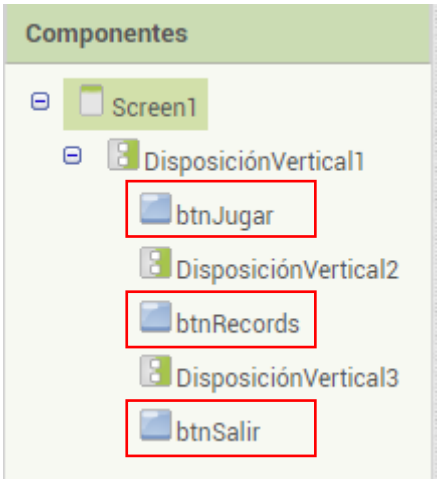


A cada disposición le vamos a modificar el alto por un porcentaje de 5%.

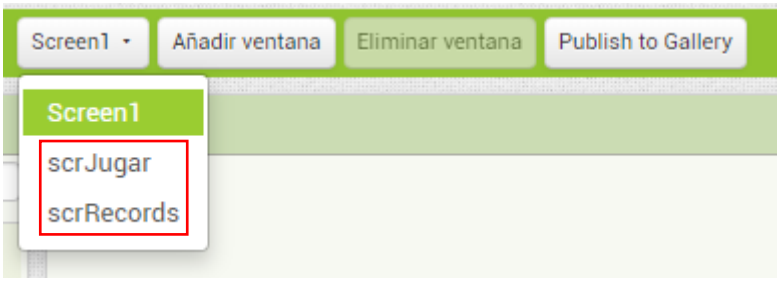
Utilizando porcentajes la separación será la misma dependiendo del móvil según la resolución que tenga.

Esta será la apariencia desde nuestro dispositivo móvil.

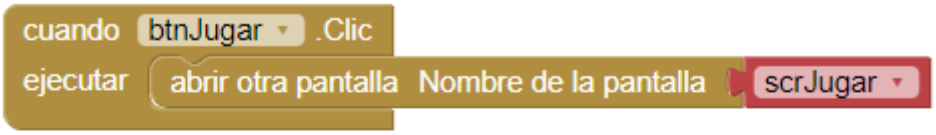
A continuación le vamos a cambiar el nombre a los botones:

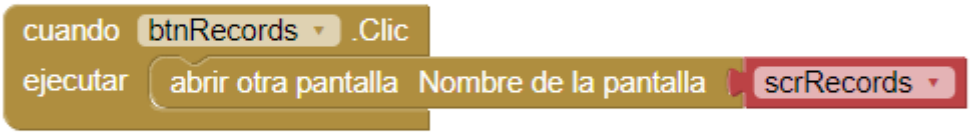


Vamos a agregar las correspondientes ventanas.

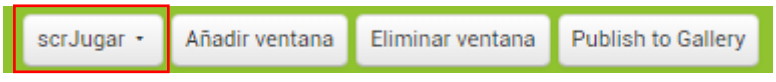


La codificación de los bloques será:



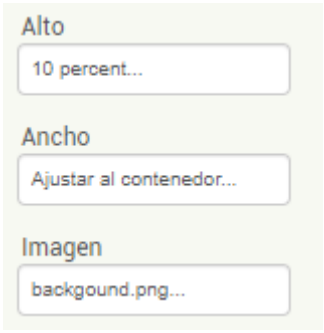


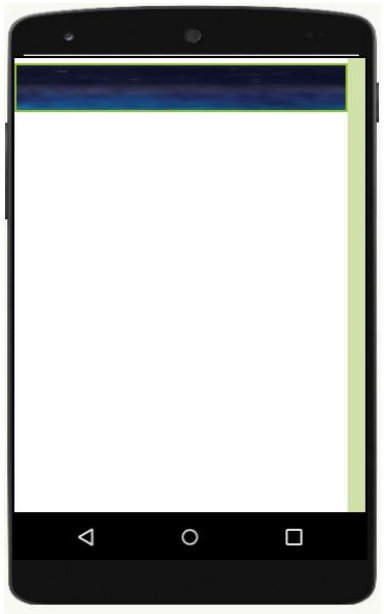
Ahora seleccionamos la pantalla donde se realizará el juego:



Vamos a modificar sus propiedades:

En la parte superior agregamos una disposición horizontal con las siguientes propiedades:



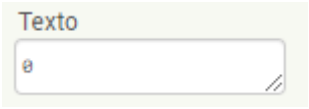


Vamos a agregar dos etiquetas:

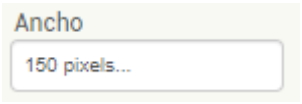
Con las siguientes propiedades:

El color de texto es Custom con el valor #bb86fcff

Agregamos la segunda etiqueta donde se pondrás los puntos con las mismas propiedades de la anterior.

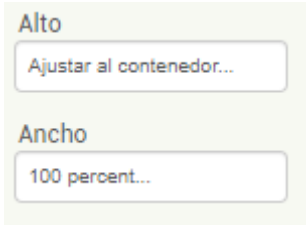


Modificamos el ancho de esta etiqueta:



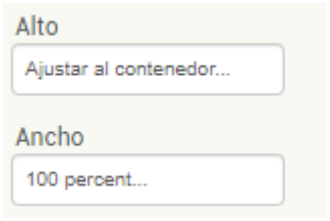
Ahora en la parte inferior vamos a agregar una disposición vertical.

Con las siguientes dimensiones:

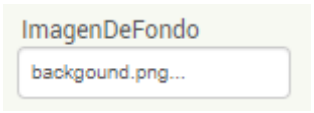


En el interior agregamos un Lienzo.

Cambiamos el tamaño.

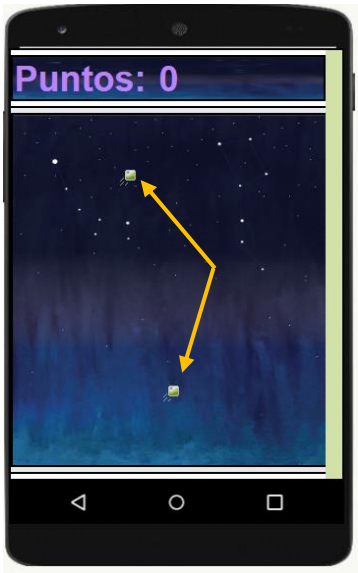


Agregamos imagen de fondo:



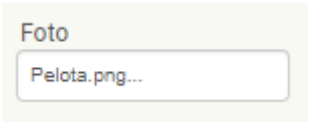
Este será el resultado:

Ahora vamos a agregar dos SpriteImage.

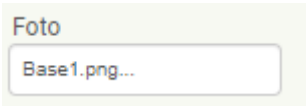


Uno será la base y el otro la pelota.

Para la pelota:

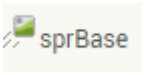


Para la base:



Este será el resultado:

Cambiamos el nombre a los SpriteImage.

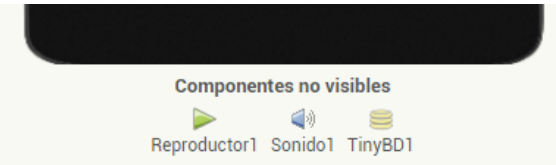


Vista desde el móvil.

Podrás observar que las posiciones no coinciden, no pasa nada esto lo vamos a controlar por medio de la programación.

Esto es porque el tamaño del visor y el de nuestra pantalla no es el mismo.

Ahora vamos a agregar 3 elementos de tipo no visible.



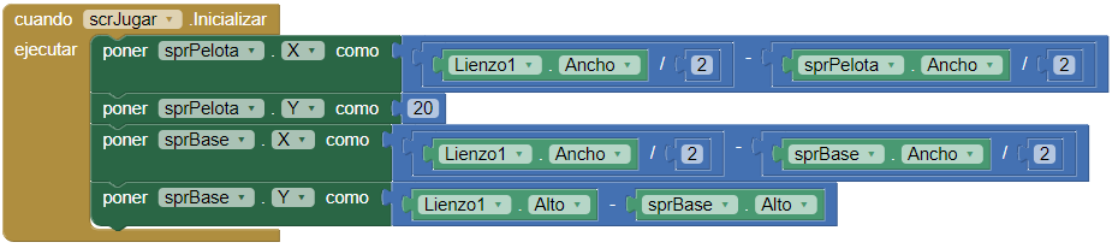
El Reproductor par reproducir una música de fondo (para melodías superiores a 10 segundos .

Sonido para reproducir un sonido cuando la bola colisiona, la reproducción no puede superar los 10 segundos.

TinyBD para almacenar la puntuación de cada jugador.

Antes de seguir vamos a ver como son las coordenadas en el lienzo.

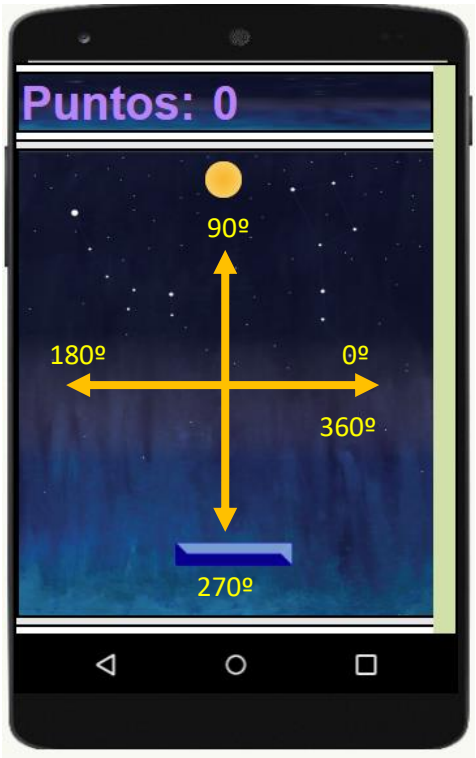
Ahora con programación de bloques vamos a colocar cada elemento en su sitio.



Este será el resultado:

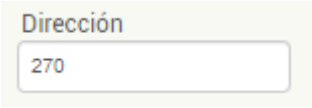
Ahora en diseño en las propiedades de la pelota.

¿Cómo funciona la dirección?



Hacia donde se va a mover el Sprite.

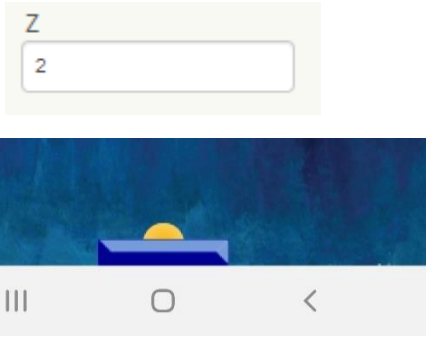
Vamos a poner:



Velocidad es el número de pixeles por el intervalo de milisegundos.

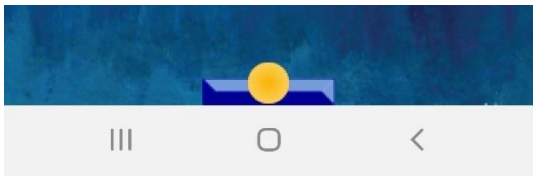
Que significa la coordenada Z: en el caso de que dos Sprite se superpongan, cal de ellos va a estar por encima del otro.

Si al Sprite de la base le ponemos un 2.



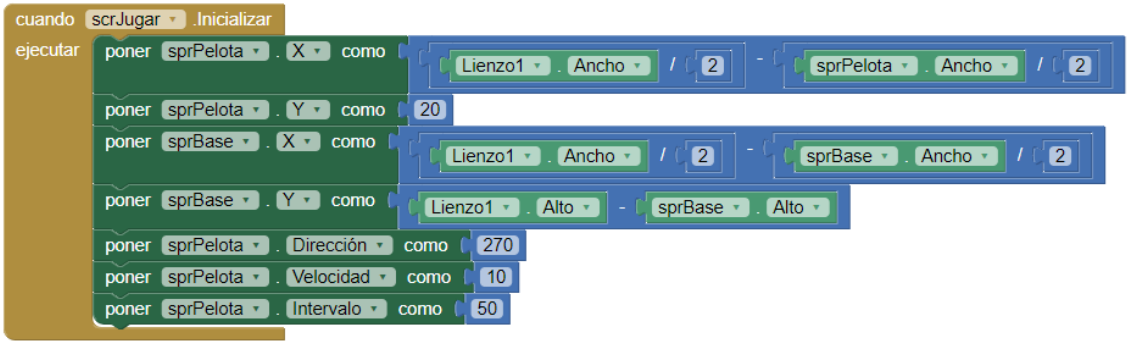
La base estará por encima de la pelota.

Si el dos se lo ponemos a la pelota:



Tu puedes dejar la opción que más te guste.

Volvemos a la programación de bloques:



Ahora a sprPelota le asignamos las propiedades Dirección, Velocidad e Intervalo.

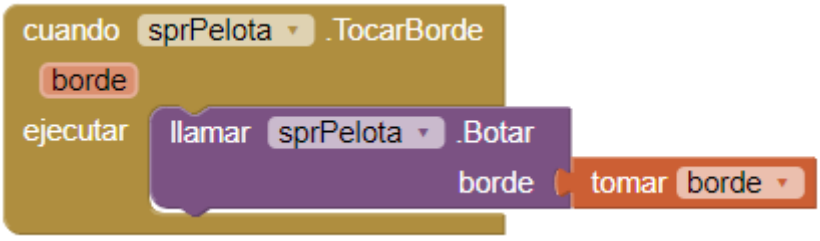
Cada 50 milisegundos la pelota se desplaza 10 pixeles en dirección hacia abajo.

Para que la dirección no sea siempre en sentido recto hacia abajo lo vamos a realizar con un valor aleatorio.

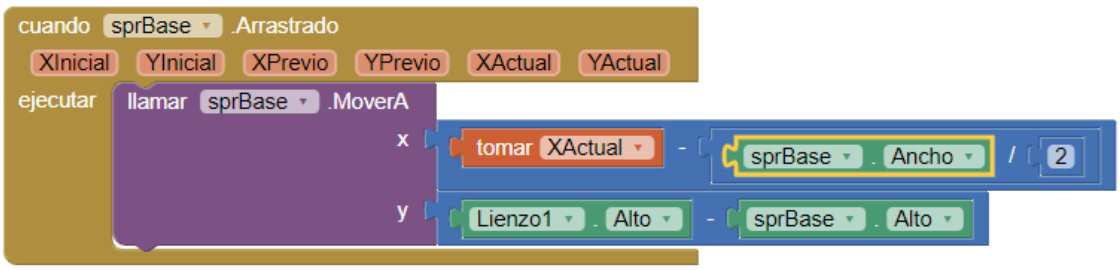
Le asignamos un valor aleatorio entre 235 y 315.

La siguiente pregunta ¿Cómo hacemos que rebote la pelota?

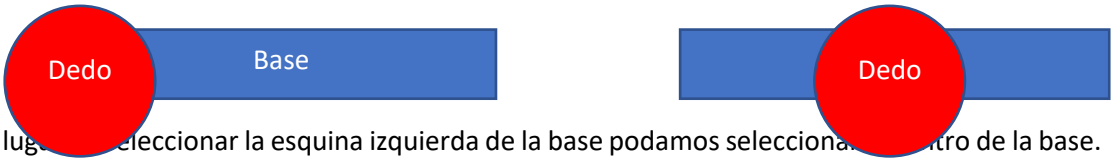
Ten en cuenta que tenemos los siguientes bordes.



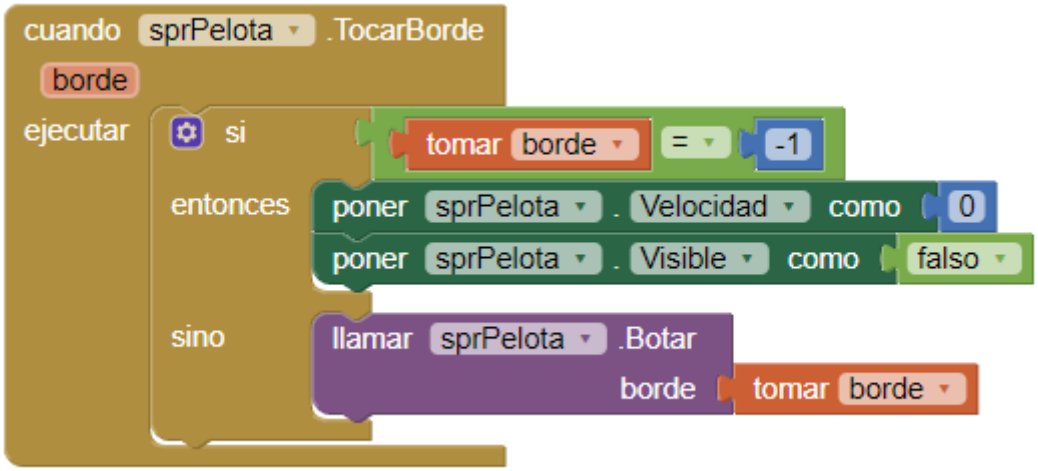
Ahora queremos mover la base.



La coordenada y tendrá un valor fijo, ya que no queremos que se mueva de arriba hacia abajo y viceversa.
La coordenada x será el valor actual, es decir a donde se desplaza el dedo, menos la mitad del ancho del sprBase, si no hacemos esto que pasaría, pues que al seleccionarlo:



En lug eleccionar la esquina izquierda de la base podamos seleccion tro de la base.

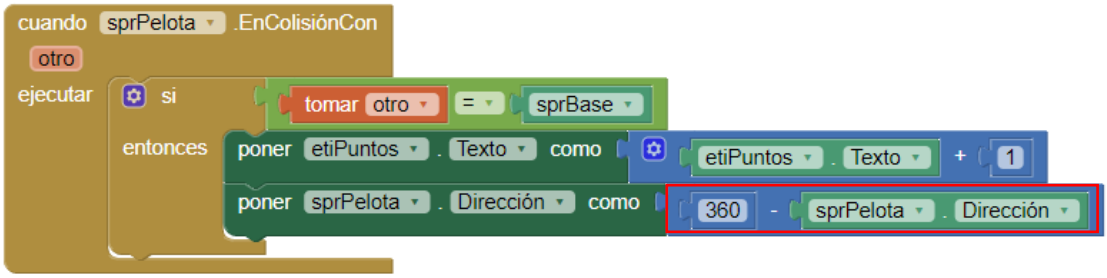


Hemos modificado los bloques cuando sprPelota.TocarBorde, con un condicional controlamos que si toca el borde -1 (es el borde inferior) la pelota se pare poniendo una velocidad de 0 y además desaparezca haciendo la propiedad visible en falso.

En la opción de que cuando se inicializa esta pantalla en la última línea hemos vuelto a hacer visible el sprPelota con la propiedad Visible a cierto, ya que si continuamos con una nueva partida la pelota no estaría visible.

Ahora queremos controlar que conado la pelota toca a la base esta tiene que rebotar sin tocar el borde inferior.

Vamos a renombrar la Etiqueta2 por el nombre de etiPuntos.

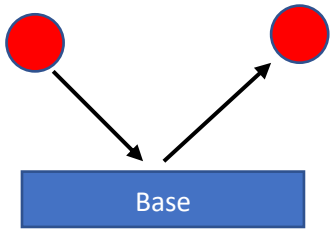


Cuando la pelota EnColisiónCon otro.

Ponemos un condicional porque si hay varios Sprites podemos controlar con cual se colisiona, en este ejemplo no haría falta pero así lo tenemos en cuenta.

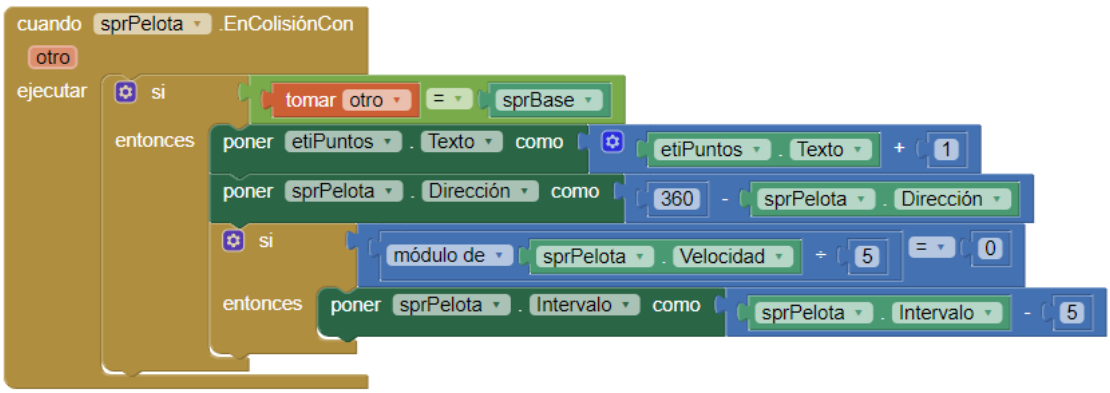
Si cumple la etiqueta etiPuntos incrementa su valor en 1.

La pelota cambia de dirección según la formula.

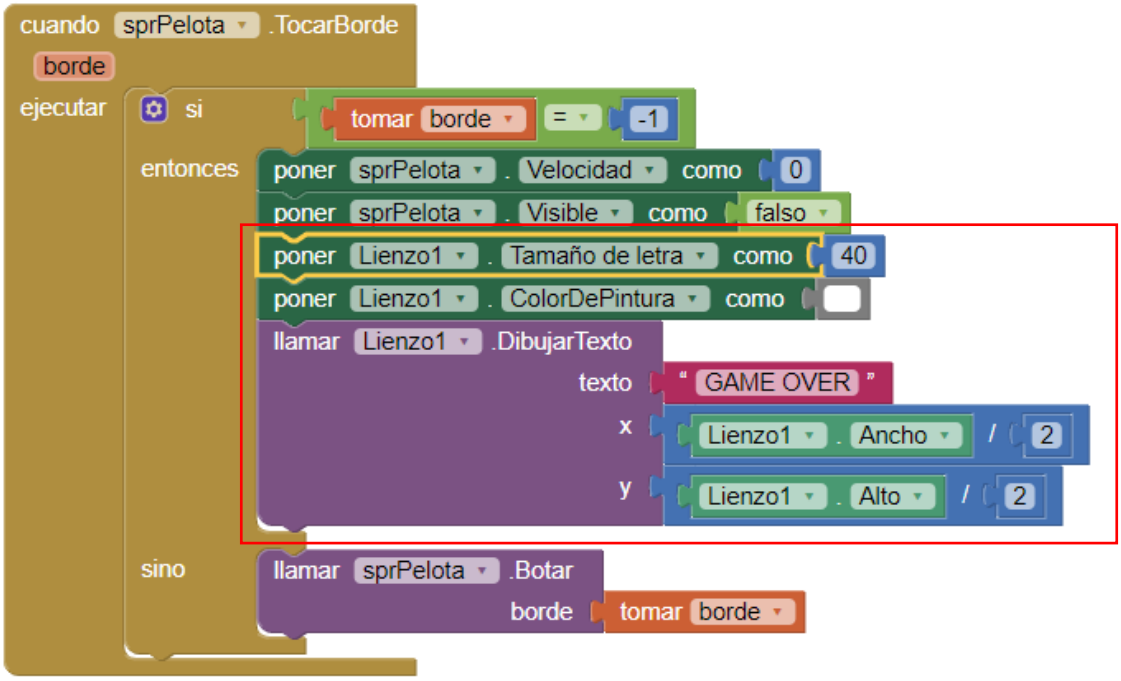


Si rebota en esta dirección que siga la nueva dirección.

Ahora queremos que cada vez que la pelota toque a la base la velocidad incremente un poco



Hemos agregado la condición de si modulo de velocidad entre 5 es igual a 0, esto significa que por cada 5 puntos el intervalo de sprPelota disminuye en 5.



En Cuando sprPelota.TocarBorde, cuando se cumple la condición de que ha tocado el borde inferior hemos agregado las siguientes instrucciones:

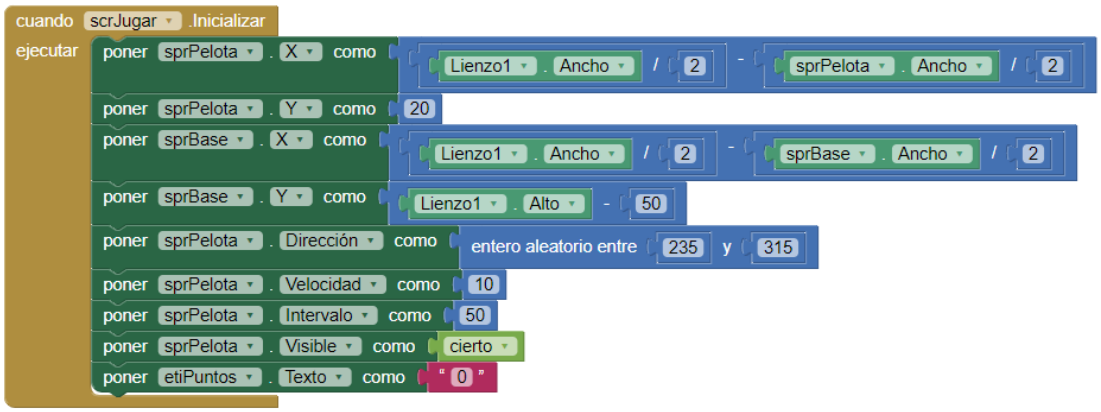
Modificamos el tamaño de la letra a 40.

Estas tienen que estar de color blanco.

Imprime el texto “GAME OVER” en las respectivas coordenadas x e y.

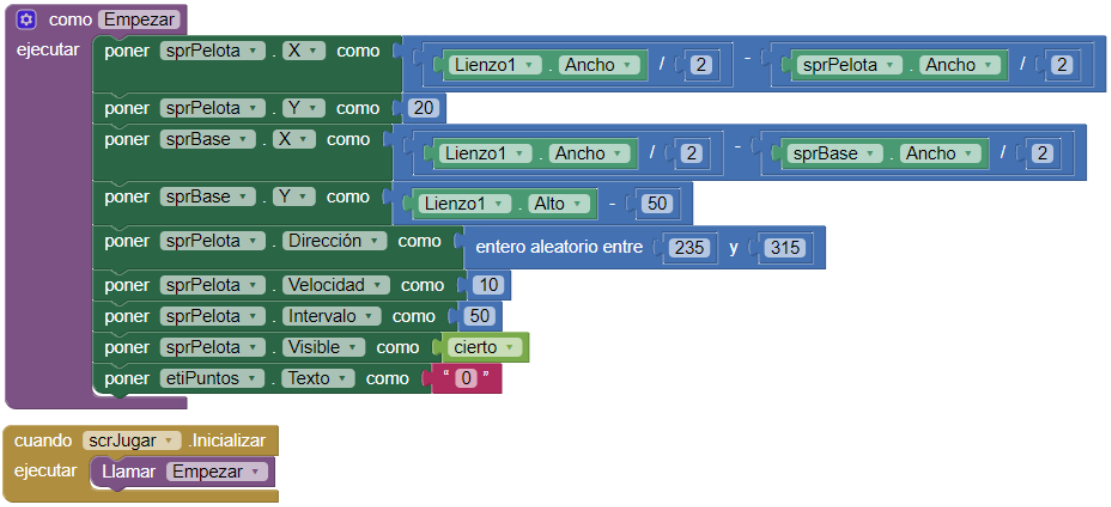
Ahora vamos a preguntarle al usuario si quiere volver a jugar.

Nos encontramos todo lo que se ejecutar al iniciar la pantalla se tiene que repetir cuando el usuario quiere volver a jugar, para ello vamos a utilizar procedimientos.



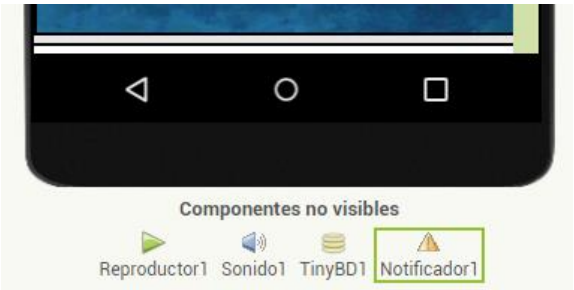
Cuando se inicializa la pantalla hemos agregado que la etiqueta etiPuntos.Texto es igual a 0.

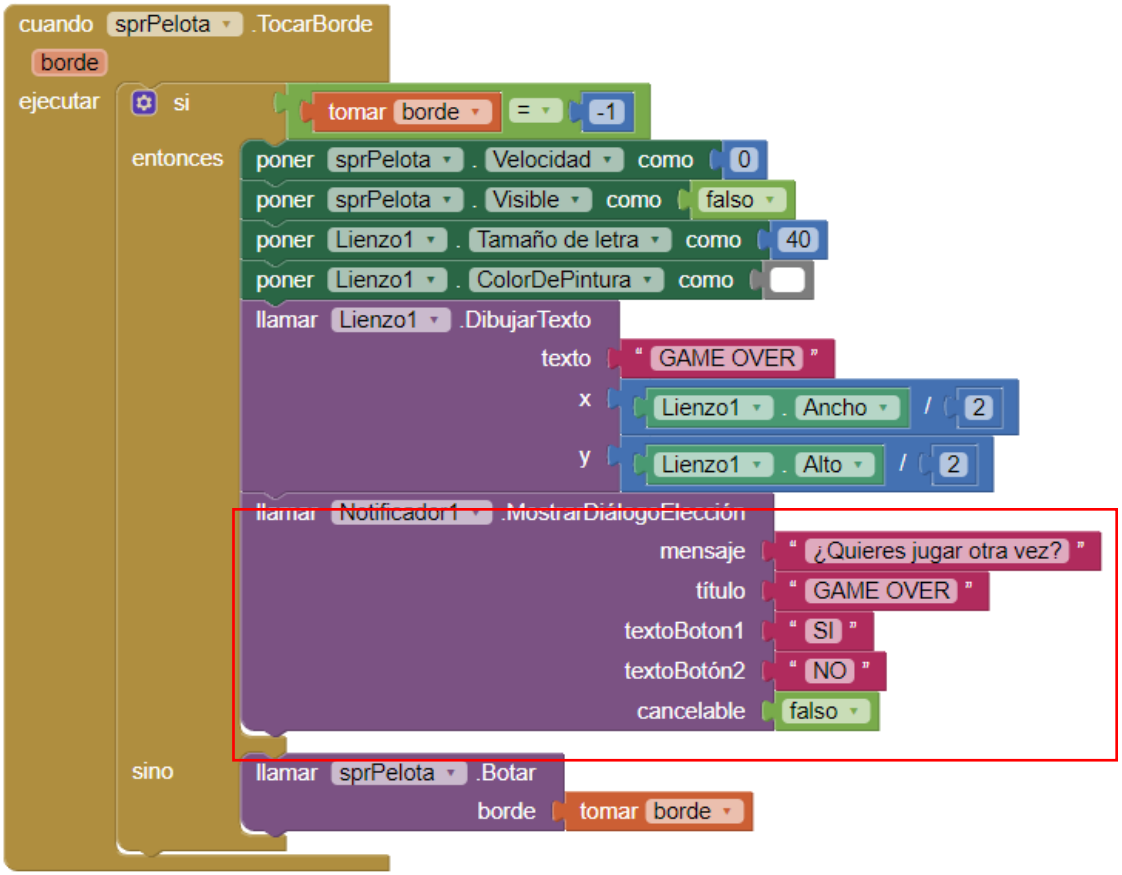
Ahora vamos a crear in procedimientos llamado Empezar.



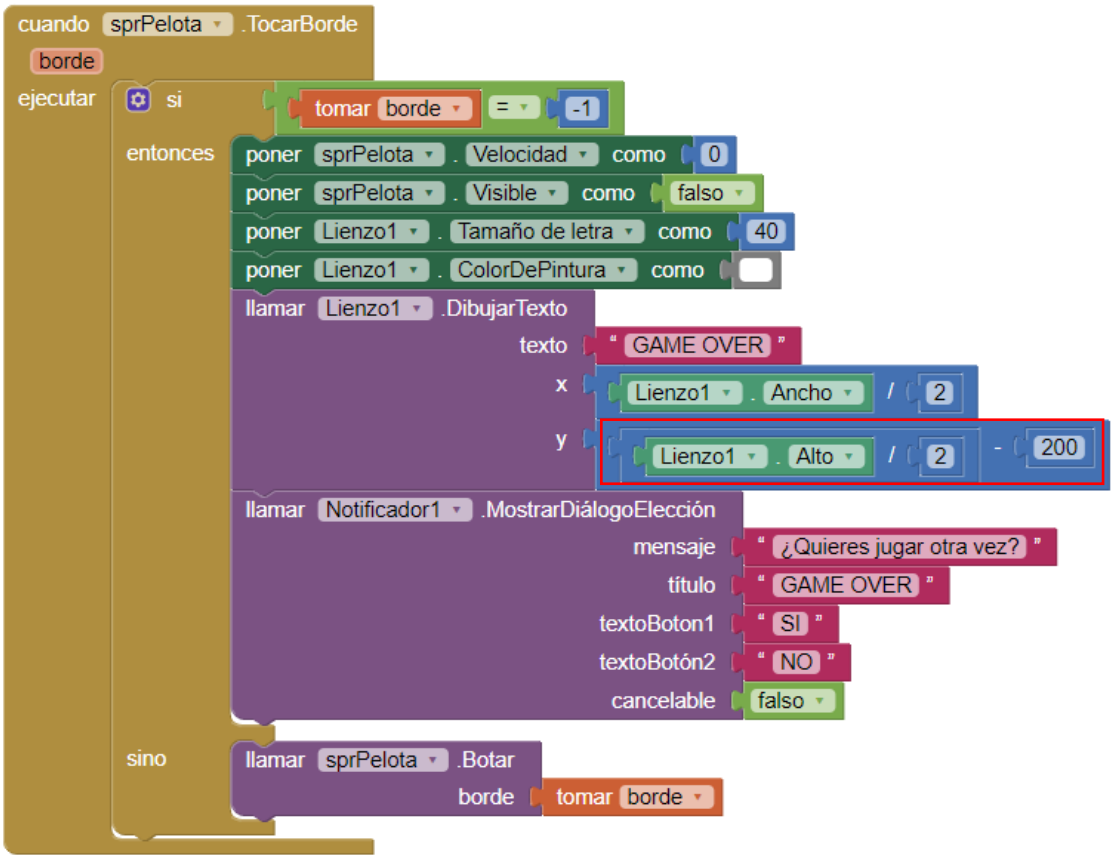
Hemos creado el procedimiento y lo llamados con el evento cuando scrJugar.Inicializar.

Vamos a agregar un notificador para que nos pregunte si queremos continual.

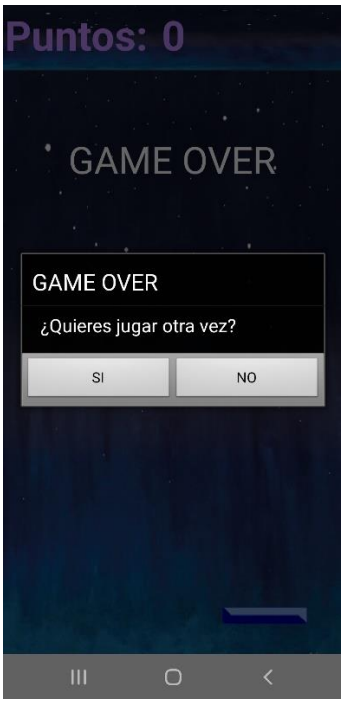




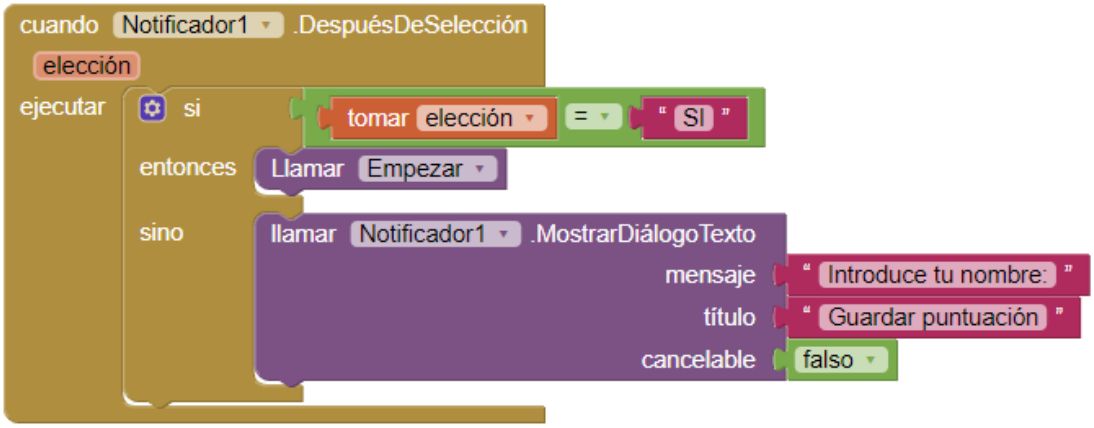
Agregamos un bloque de mostrar diálogo donde el titulo ser “GAME OVER” el mensaje “¿Quieres jugar otra vez? Con dos botones “SI” y “NO” y desactivamos la opción de Cancelar.



Hemos cambiado la coordenada, ya que el notificador me tapaba el mensaje “GAME OVER”



Esto permite ahora llamar el bloque DespuesDeSelección.



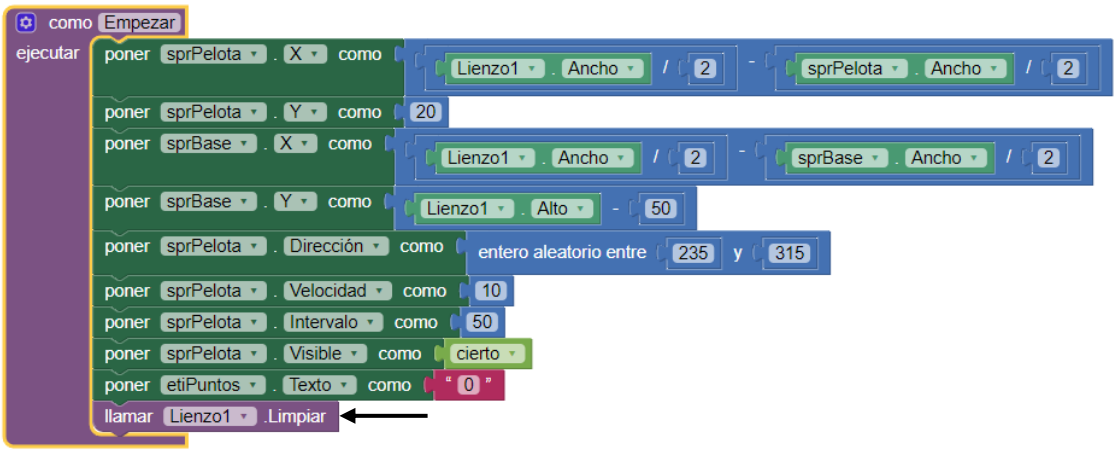
Si la respuesta es Sí llama al procedimiento para inicializar los valores, de lo contrario que nos pregunte por nuestro nombre par guardar la puntuación que hemos obtenido.

Después de la entrada de texto:

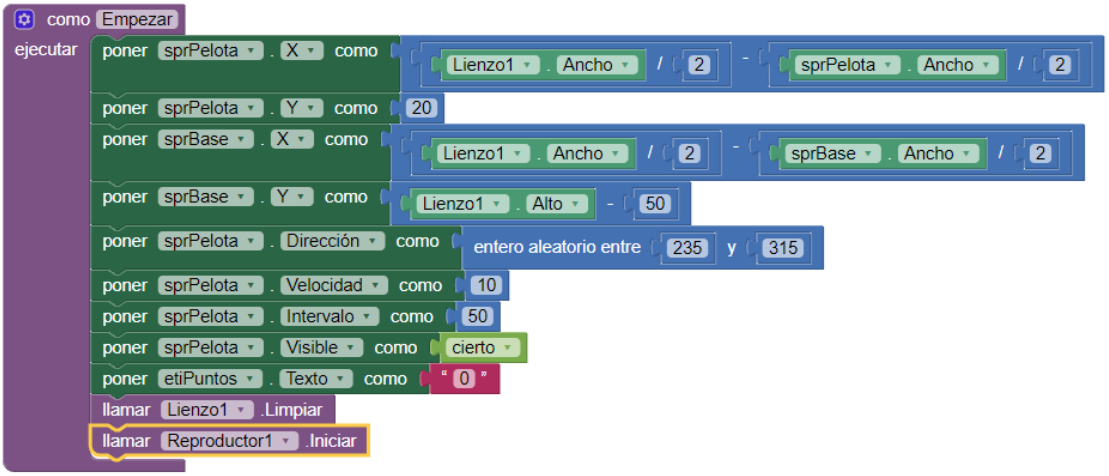
Guardamos la información del nombre del jugador con su puntuación.

Cada jugador solo podrá tener una puntuación.

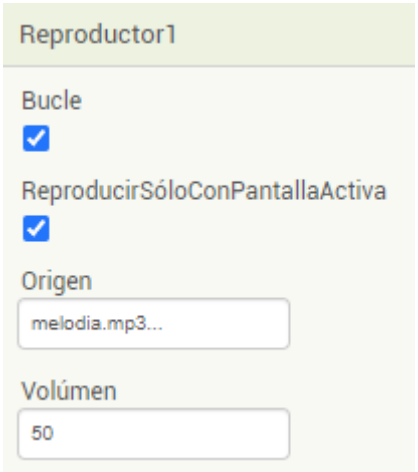
Cerramos la aplicación.

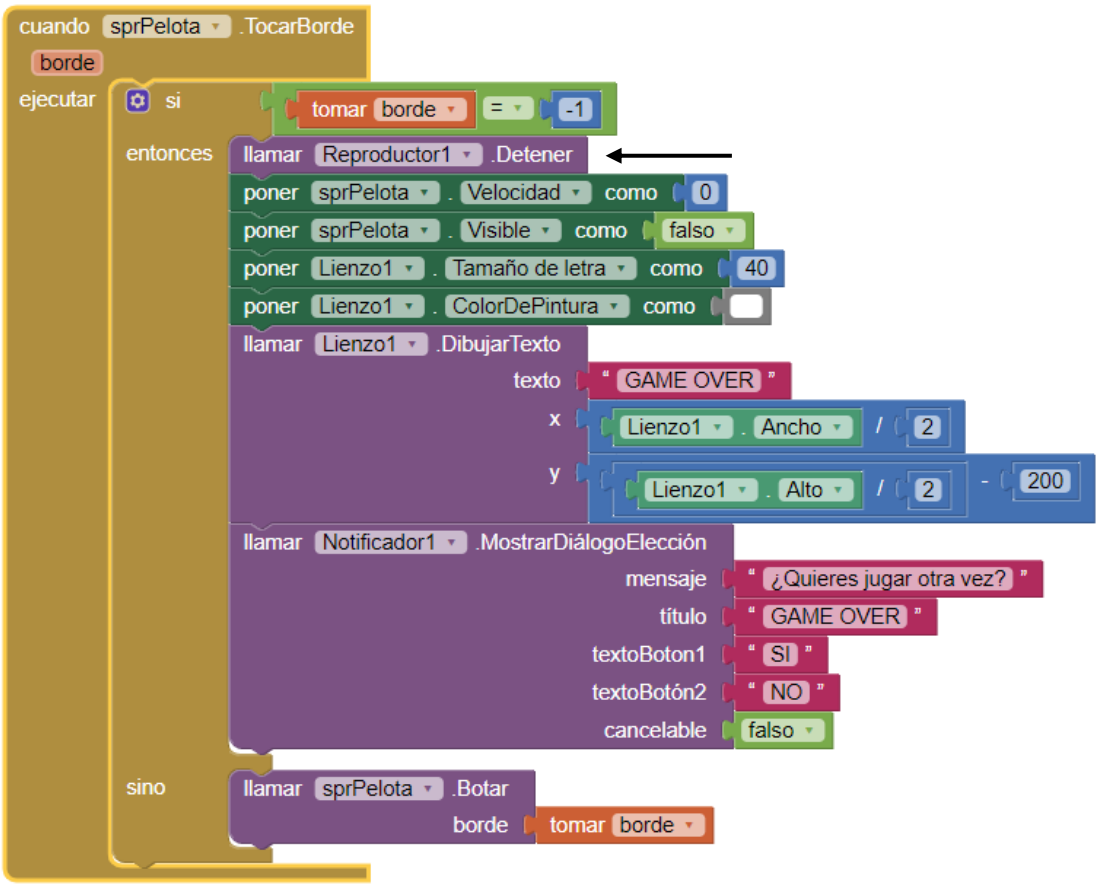


En el procedimiento hemos agregado el bloque de limpiar lienzo.
Vamos a agregar la melodía.



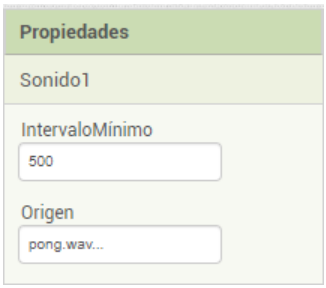
Agregamos el bloque Reproductor1.Iniciar.
Ahora nos vamos a modo de diseño y en propiedades:
En propiedades seleccionaremos la melodía:

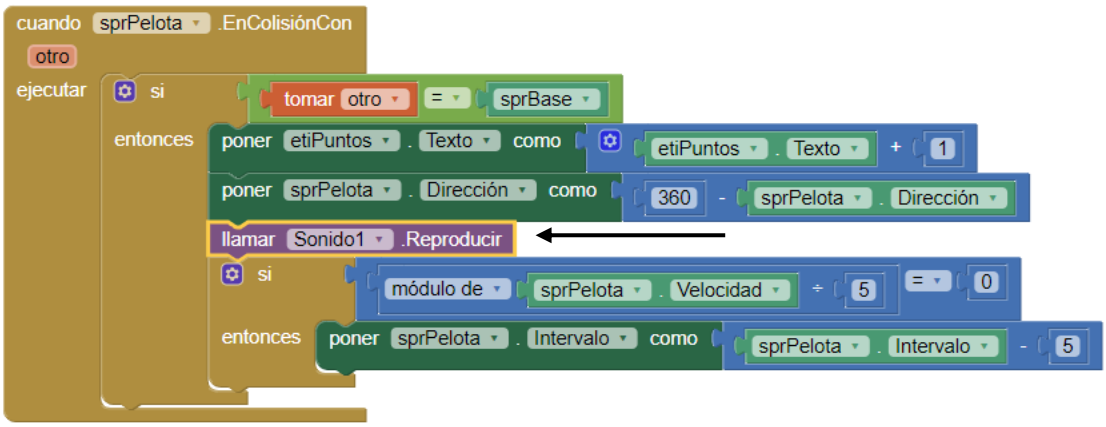




Cuando sprPelota toca el borde inferior también queremos que se detenga la música.

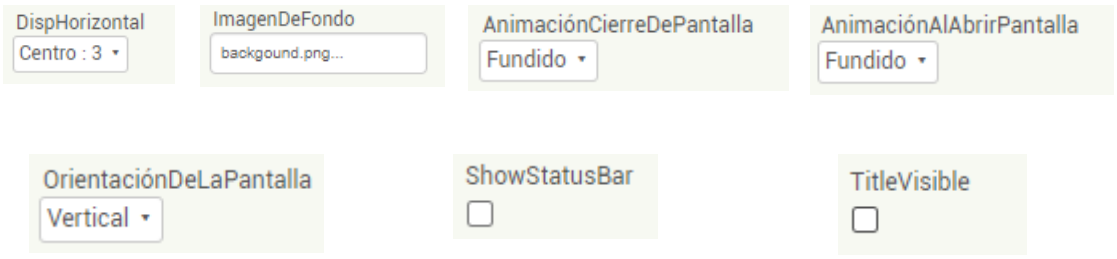
Ahora vamos a poner en propiedades el sonido para el componente Sonido1.



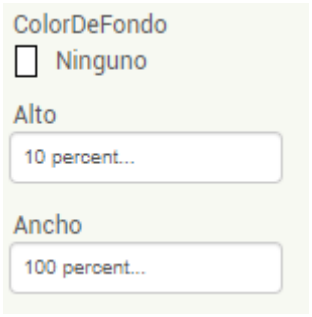


En el bloque cuando sprPelota.EnColisiónCon agregamos el bloque llamar Sonido1.Reproducir.

Vamos a la pantalla scrRecords y cambiamos las siguientes propiedades:



Agregamos una disposición horizontal y modificamos las propiedades.



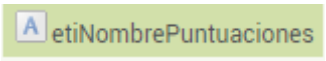
Agregamos dentro de la disposición una etiqueta que podrá puntuaciones con las siguientes propiedades:

El color Custom es #bb86fcff

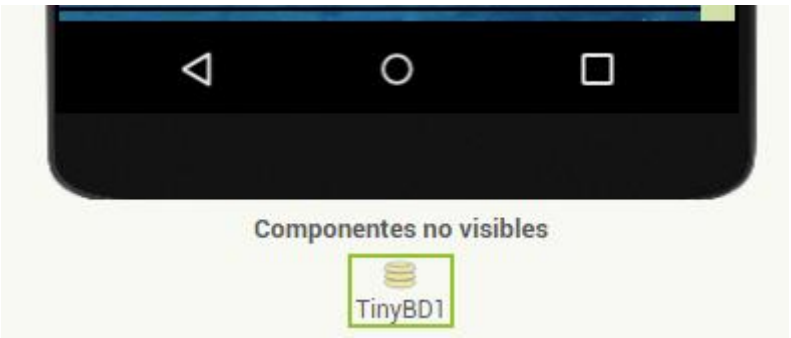
En la parte inferior agregamos otra disposición vertical, con las siguientes propiedades:

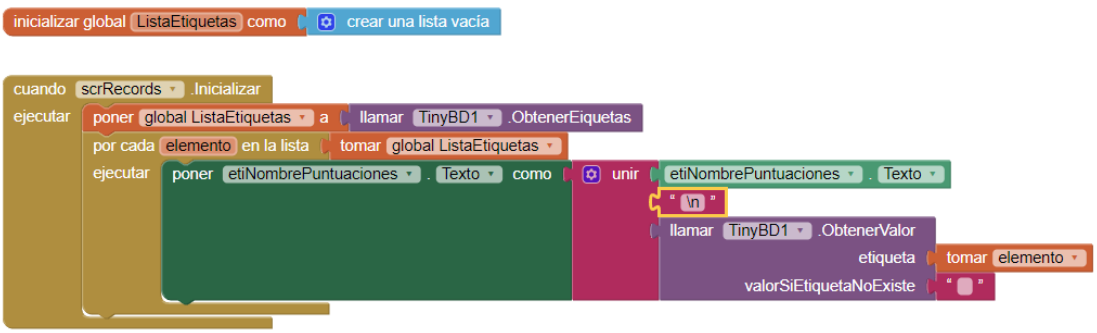
Agregamos una etiqueta con las siguientes propiedades:

Como nombre de la etiqueta:



Ahora agregamos un elemento no visible para recuperar los datos que hemos almacenado.





Inicializamos una variable llamada ListaEtiquetas de tipo lista vacía.

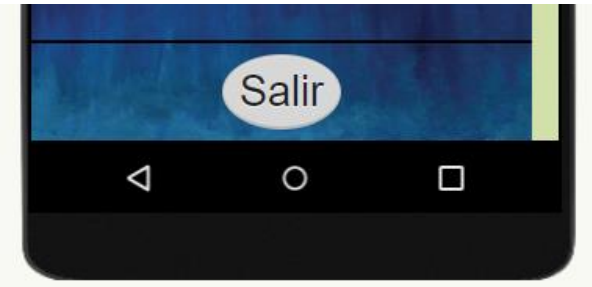
Cuando de abre la pantalla scrRecords al inicializar.

La variable ListaEtiquetas obtiene las etiquetas de la base de datos.

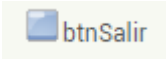
Por cada elemento de la lista ListaEtiquetas

A la etiqueta etiNombrePuntuaciones.texto concatenamos su propia etiqueta con un \n (salto de línea) seguido elemento de cada etiqueta.

En la pantalla srcRecords vamos a agregar un botón:



Lo llamaremos:



Esta será la programación en bloque:

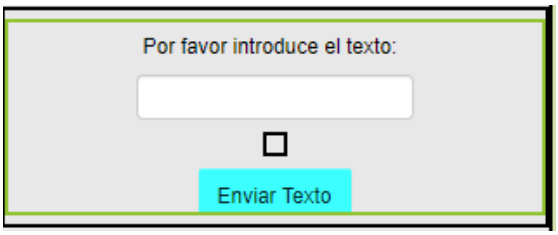
Ya lo puedes instalar en tu móvil y a jugar.

18.- Compartir archivos y texto con el componente Sharing

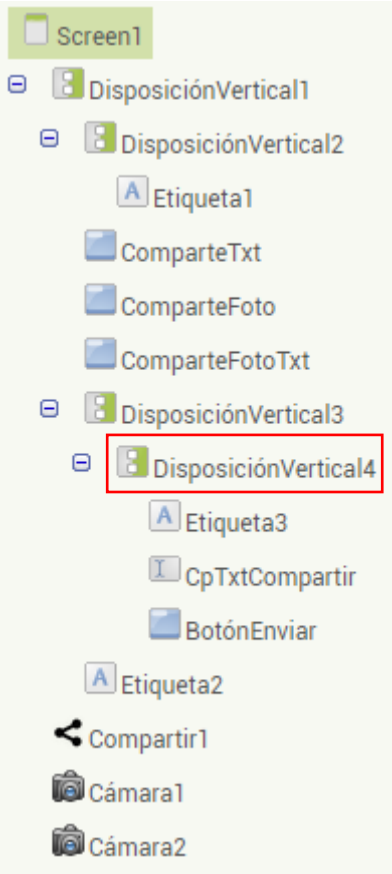
Hay muchas aplicaciones que nos permiten compartir textos y fotografías a otras aplicaciones de nuestro móvil.

Para esto vamos a realizar el correspondiente diseño:

En el recuadro que tenemos seleccionado tenemos una disposición vertical que además tiene 3 elementos, la tenemos como no visible, aquí tienen que ir los siguientes elementos:



No tienen que estar visibles.



Recuerda que la DisposiciónVertical4 no está visible.

Vamos a programar los bloques:

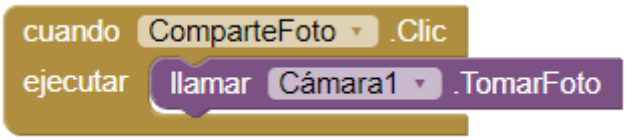
Definimos una variable global llamada Texto y asignándole el valor de texto vacío.

Cuando hacemos clic en el botón CompárteTxt:

Cambiamos el texto del BotónEnviar a “Enviar texto”

El DispositivoVertical4 lo hacemos visible.

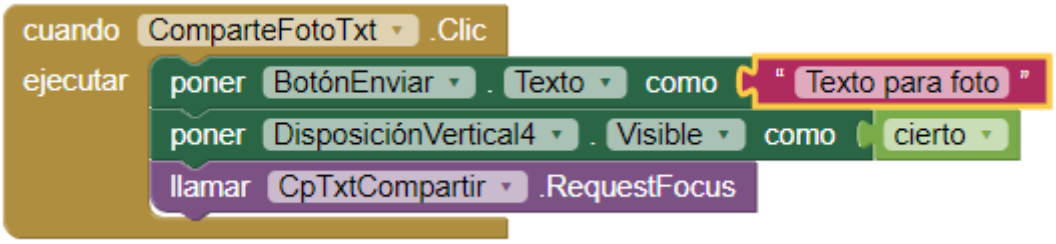
Ponemos el foco en el campo de texto CpTxtCompartir, para poder escribir el texto.



Al hacer clic en el botón CompartеFoto llamaremos a la Cámara1.TomarFoto.

Después de tomar la foto ya podemos compartir la imagen.

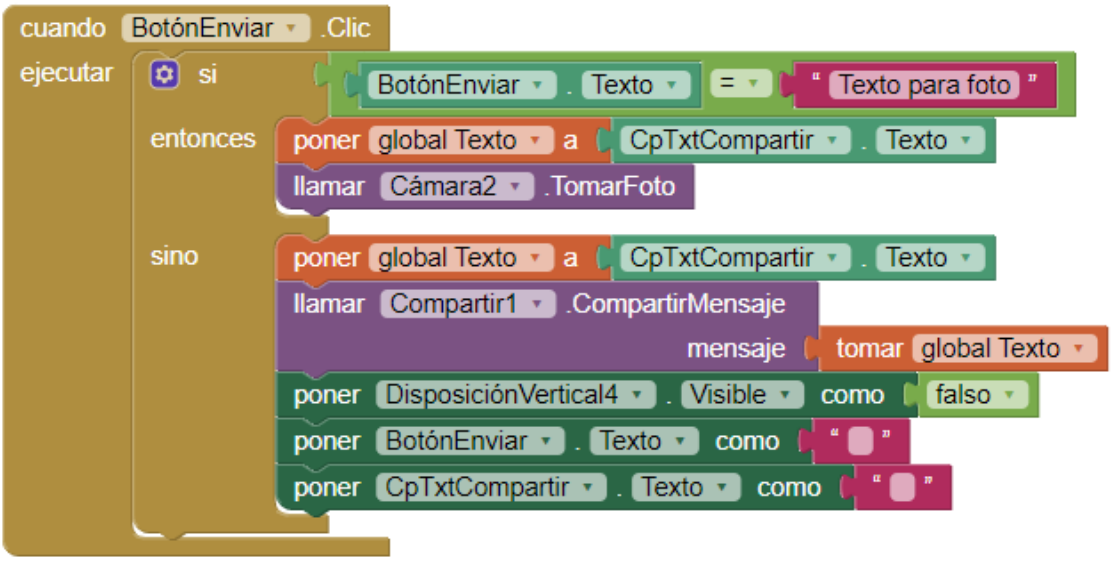
Ahora vamos a programar el botón para compartir texto e imagen.



Cambiamos el texto dele botón a “Texto para foto”

Hacemos visible la DisposiciónVertical4.

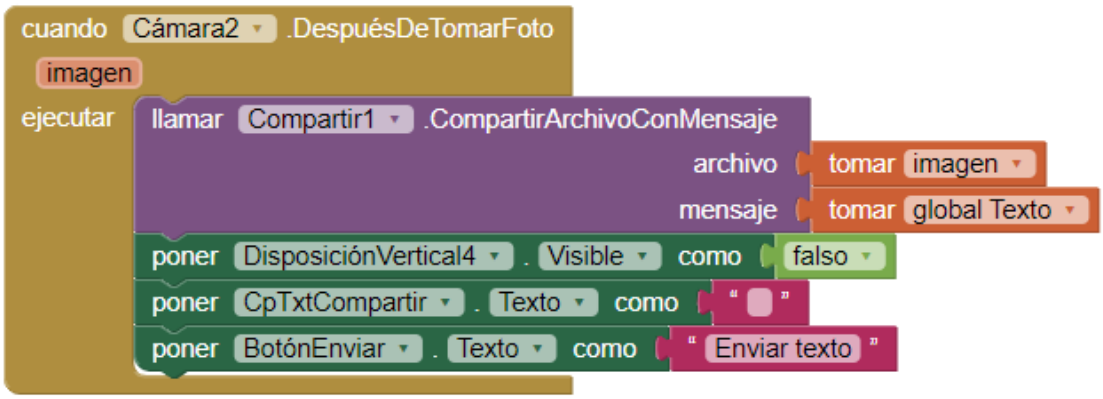
Ponemos el foco en el campo de texto CpTxtCompartir para escribir el texto para la foto.



Al seleccionar el botón BotónEnviar realizamos la siguiente comparación:

Si el botón BotónEnviar contiene el texto “Texto para foto” asignamos a la variable Texto el texto que hemos escrito en la caja de texto y tomamos la fotografía con la Cámara2.

Si es así igualmente a la variable Texto le asignamos el texto que hemos escrito en la caja de texto, compartimos el mensaje, dejamos no visible la DisposiciónVertical4 y borramos el texto del botón BotónEnviar.



Después de tomar foto con la Cámara2 compartimos la imagen y el texto.

La visibilidad de DisposiciónVertical4 la ponemos a falso.

Vaciamos el contenido de la caja de texto CpTxtCompartir.

Cambiamos el texto del BotónEnviar por “Enviar texto”.

19.- PRÁCTICA FINAL

Ahora que ya somos unos programadores profesionales, vamos a crear una aplicación que será la lista de la compra.

Primero te voy a explicar el funcionamiento del programa, por sin te atreves a realizarla sin mirar los pasos que hay que realizar, una vez terminado o con dudas que no te permiten avanzar podrás consultar los pasos para realizar esta aplicación.

Figura 1

Figura 2

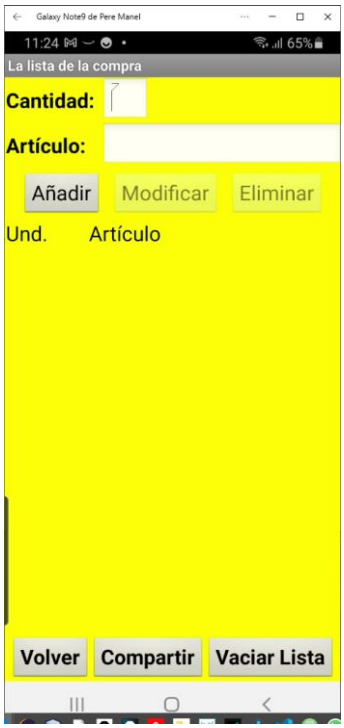


Figura 3

Figura 1: Cuando ejecutamos la aplicación se mostrará esta figura, si seleccionamos el botón de Ayuda mostrará la pantalla que se muestra en la Figura 2.

Figura 2: Al seleccionar el botón “Cerrar Ayuda” vuelve a mostrar la pantalla que se muestra en la Figura 1.

Al seleccionar el botón Entrar muestra la pantalla que se muestra en la Figura 3.

Figura 3: Cuando estamos en la pantalla de la Figura 3, con el botón Añadir podemos añadir elementos a la lista de la compra.

Podrás observar que los botones de Modificar y Eliminar los tengo inhabilitados.

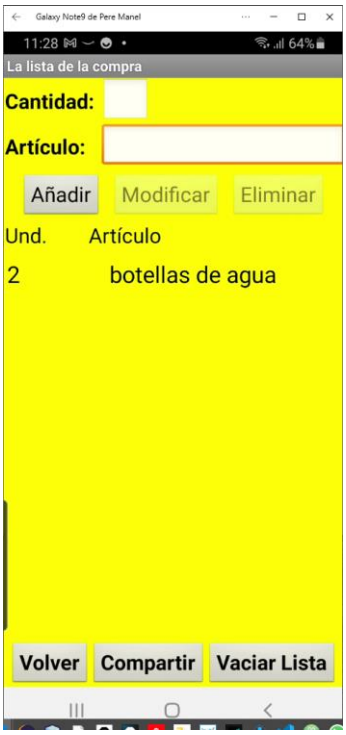


Figura 4

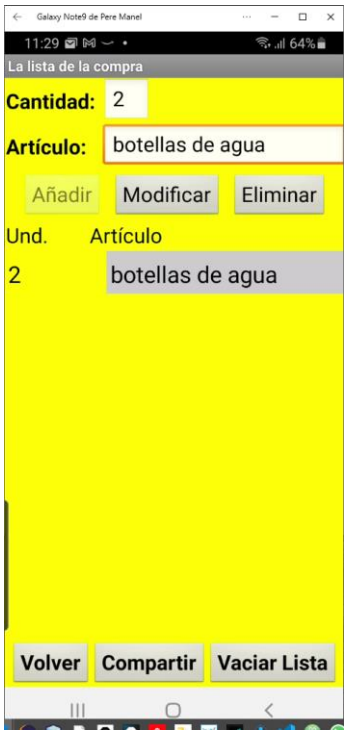


Figura 5

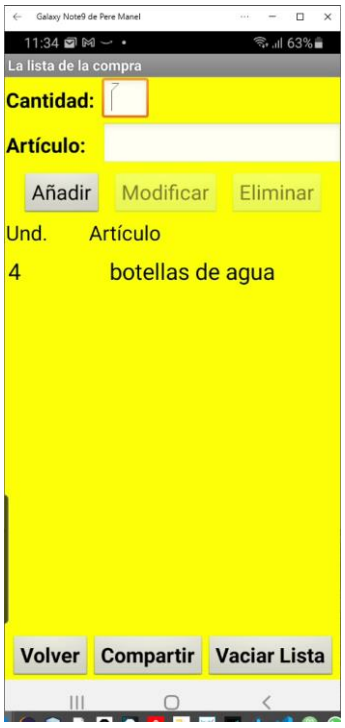


Figura 6

Figura 4: una vez agregado el elemento este se muestra en dos 2 VisorDeLista.

Figura 5: Si seleccionas algún elemento del VisorDeLista, automáticamente se muestra de nuevo en los correspondientes cuadros de texto.

El botón Añadir lo ponemos deshabilitado y habilitamos los botones Modificar y Eliminar, ya que en estos momentos es lo que estamos haciendo.

Vamos a modificar la cantidad de botellas ya que queremos comparar 4, el cambio está reflejado en la Figura 6.

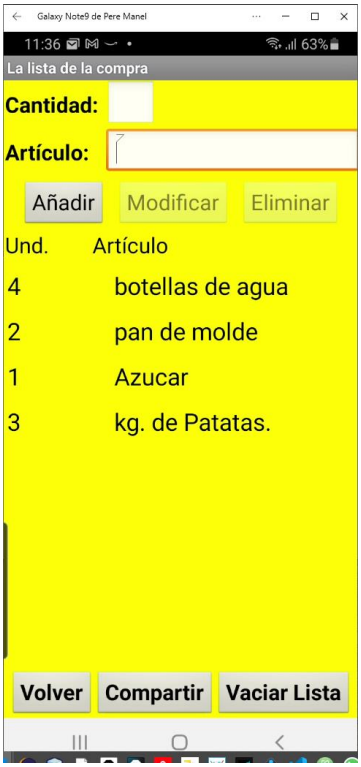


Figura 7

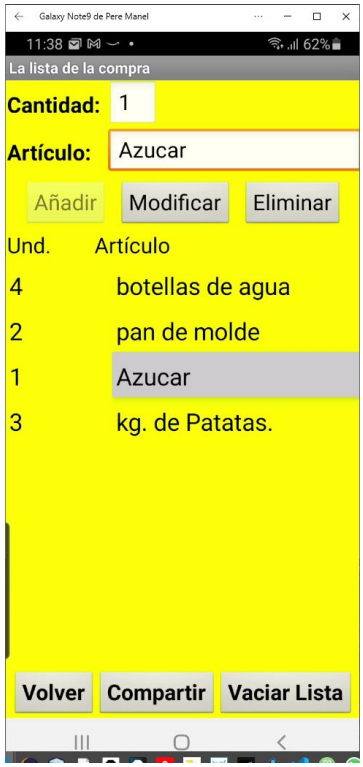


Figura 8

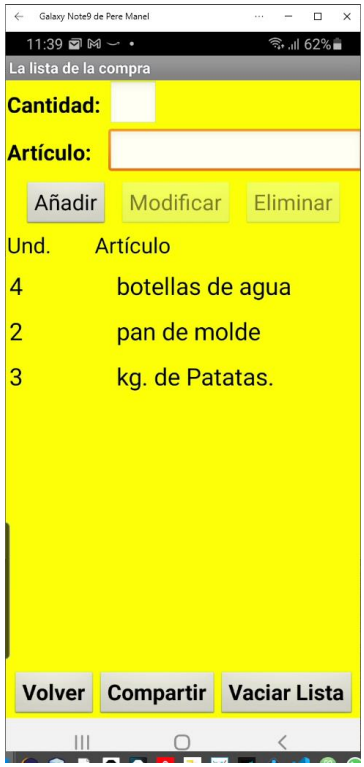


Figura 9

Figura 7: Vamos a seleccionar desde el visor el artículo azúcar ya que lo queremos eliminar.

Figura 8: Seleccionaremos el botón de Eliminar.

Figura 9: Observamos como este elemento ya no está en la lista.

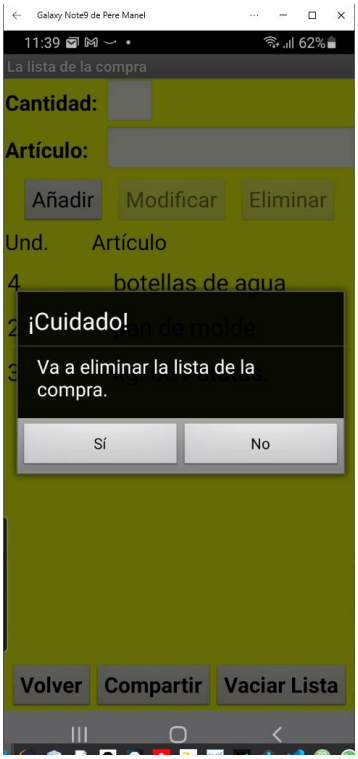


Figura 10

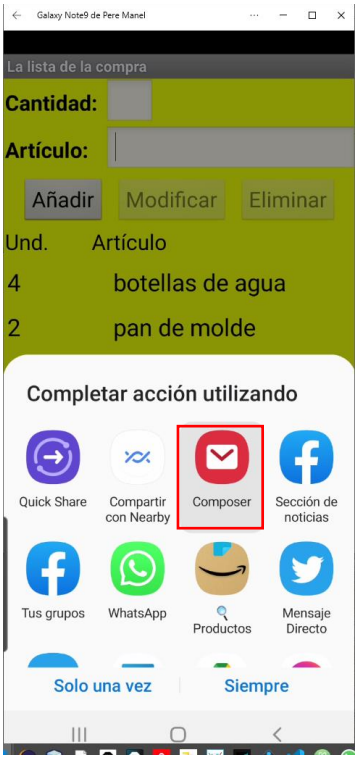


Figura 11

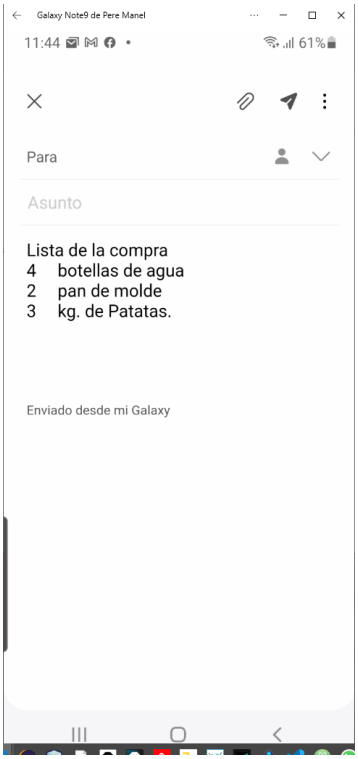


Figura 12

Figura 10: De la parte inferior hemos seleccionado el botón Vaciar Lista, observamos un mensaje avisándonos de que se va a eliminar la lista de la compra, si contestamos afirmativamente con un Si esta se borrará y si contestamos negativamente con un No la lista permanecerá.

Figura 11: la parte inferior hemos seleccionado el botón Compartir, Seleccionaremos Correo electrónico.

Figura 12: Solo queda añadir el destinatario, asunto y darle al botón de Enviar.

En la parte inferior con el botón volver volvemos a la pantalla incial.

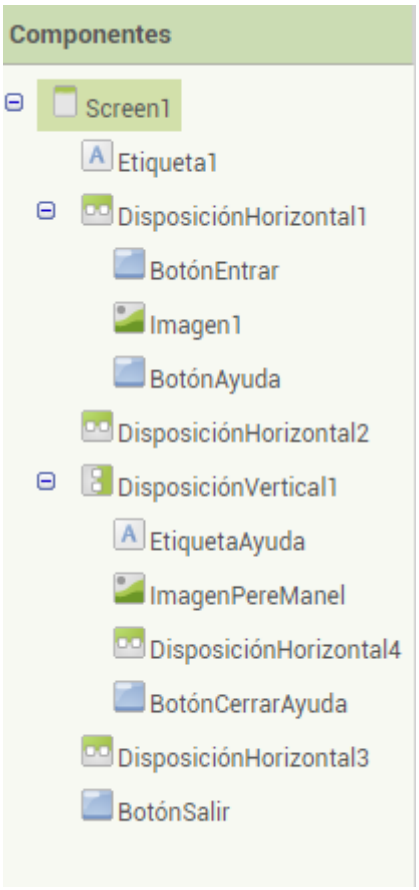
Desde la pantalla principal en la parte inferior tenemos el botón salir, este nos permitirá salir de la aplicación.

¡BUENA SUERTE!

Explicación de paso a paso de como se ha realizado este proyecto.

Pantalla principal:

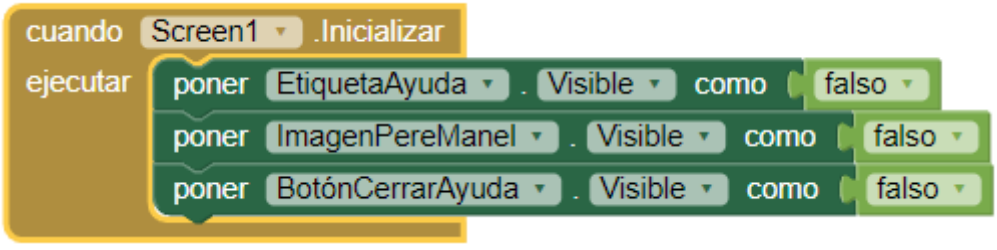
Diseño de la pantalla principal



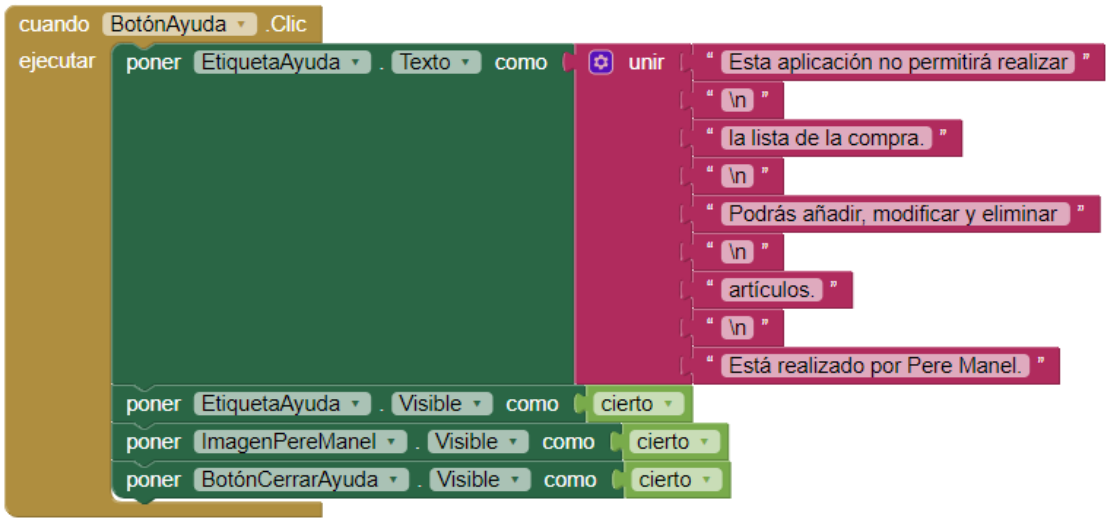
Estructura de los componente

En la estructura podrás saber el nombre que le hemos dado a cada componente.

Los bloques serán los siguientes:

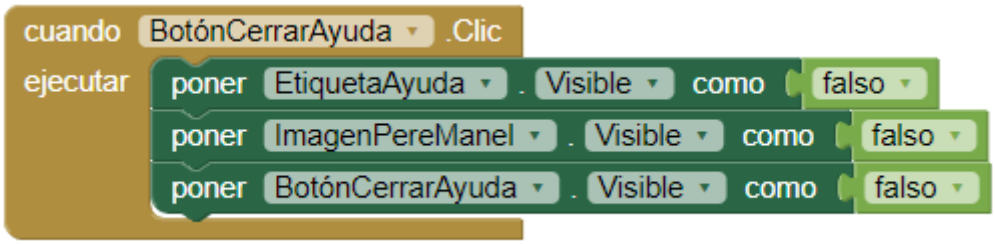


Cuando se inicializa la pantalla Screen1 ocultamos los siguientes elementos.

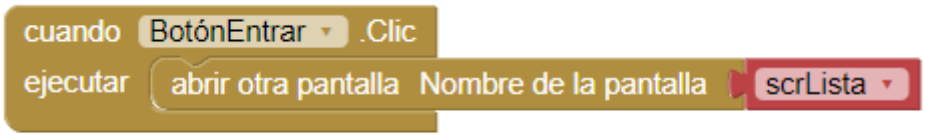


Cuando hacemos clic en el BotónAyuda a la EtiquetaAyuda le agregamos el correspondiente texto que está concatenado por varias cadenas de texto.

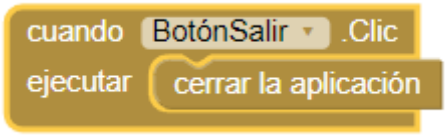
Hacemos visible EtiquetaAyuda, ImagenPereManel y BotónCerrarAyuda.



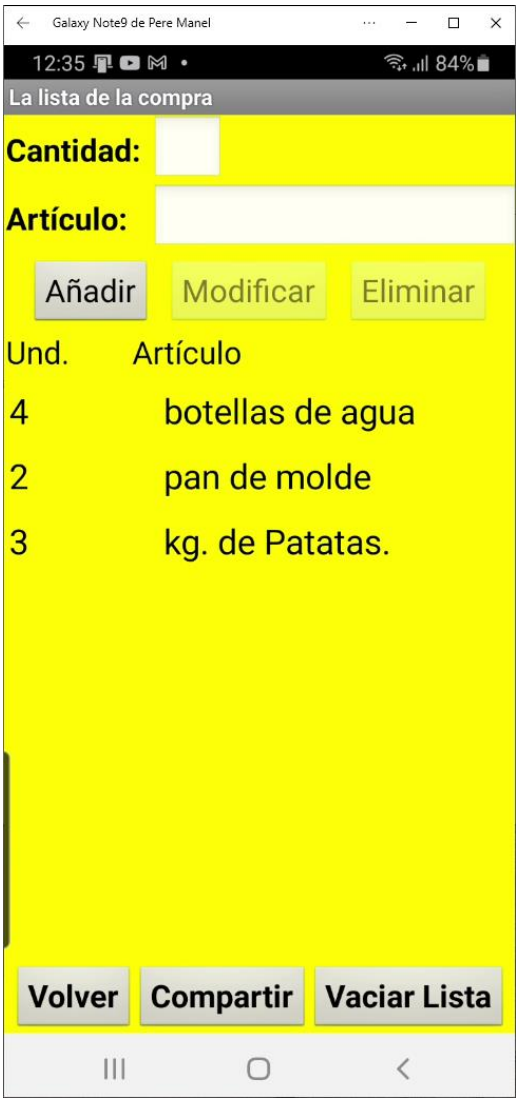
Cuando hacemos clic en el BotónCerrarAyuda volvemos a ocultar los objetos EtiquetaAyuda, ImagenPereManel y BotónCerrarAyuda.



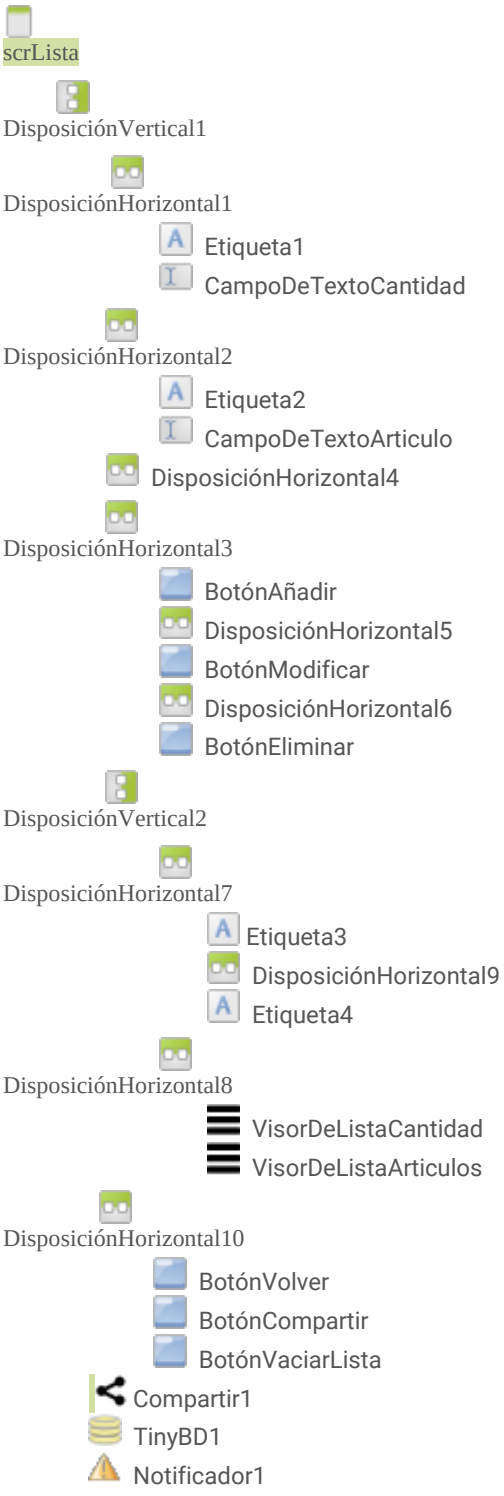
Con el BotónEntrar nos vamos a la pantalla scrLista que a continuación la crearemos.



Con el BotónSalir salimos de la aplicación.


Galaxy Note9 de Pere Manel
12:35
84%
La lista de la compra
Cantidad:
Artículo:
Añadir
Modificar
Eliminar
Und.
Artículo
4
botellas de agua
2
pan de molde
3
kg. de Patatas.
Volver
Compartir
Vaciar Lista


Componentes no visibles
Compartir1
TinyBD1
Notificador1


scrLista
DisposiciónVertical1
DisposiciónHorizontal1
Etiqueta1
CampoDeTextoCantidad
DisposiciónHorizontal2
Etiqueta2
CampoDeTextoArticulo
DisposiciónHorizontal4
DisposiciónHorizontal3
BotónAñadir
DisposiciónHorizontal5
BotónModificar
DisposiciónHorizontal6
BotónEliminar
DisposiciónVertical2
DisposiciónHorizontal7
Etiqueta3
DisposiciónHorizontal9
Etiqueta4
DisposiciónHorizontal8
VisorDeListaCantidad
VisorDeListaArticulos
DisposiciónHorizontal10
BotónVolver
BotónCompartir
BotónVaciarLista
Compartir1
TinyBD1
Notificador1


inicializar global Articulos como crear una lista vacía
inicializar global Unidades como crear una lista vacía
inicializar global CantidadArticulos como crear una lista vacía
inicializar global Indice como 0

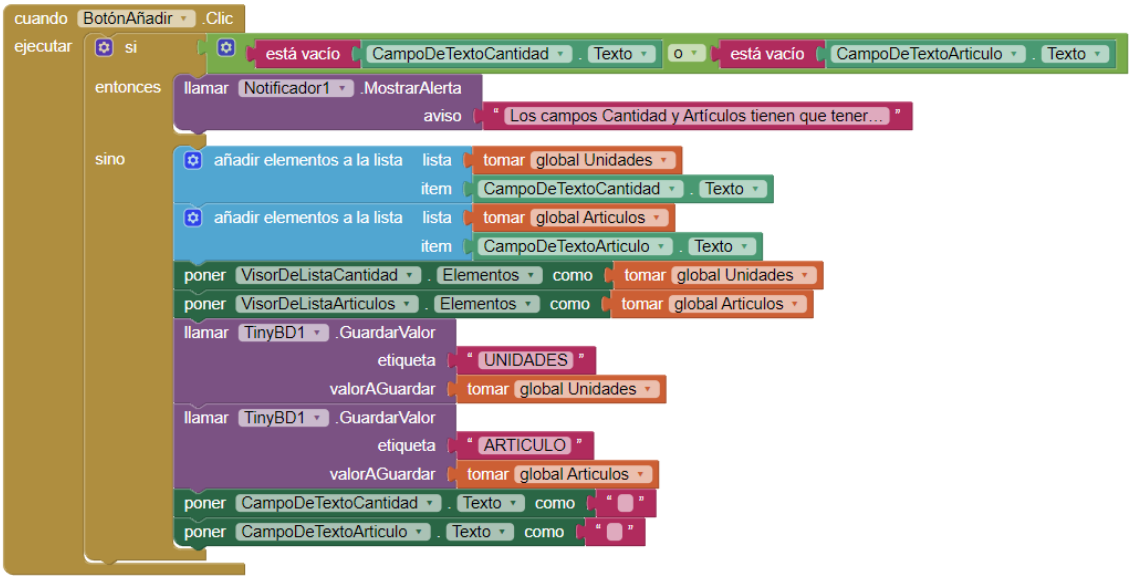
Al inicializar esta segunda pantalla los campos de texto CampoDeTextoCantidad y CampoDeTextoArticulo lo dejamos vacío.

El BotónModificar y BotónEliminar los ponemos inhabilitados.

Recuperamos de la base de datos las etiquetas UNIDADES y ARTICULO para asignárselo a sus respectivas variables de tipo lista.

Así mismo hacemos que dichos valores se visualicen en sus respectivos VisorDeListaCantidad y VisorDeListaArticulos.

Definimos los siguientes variables de tipo global.

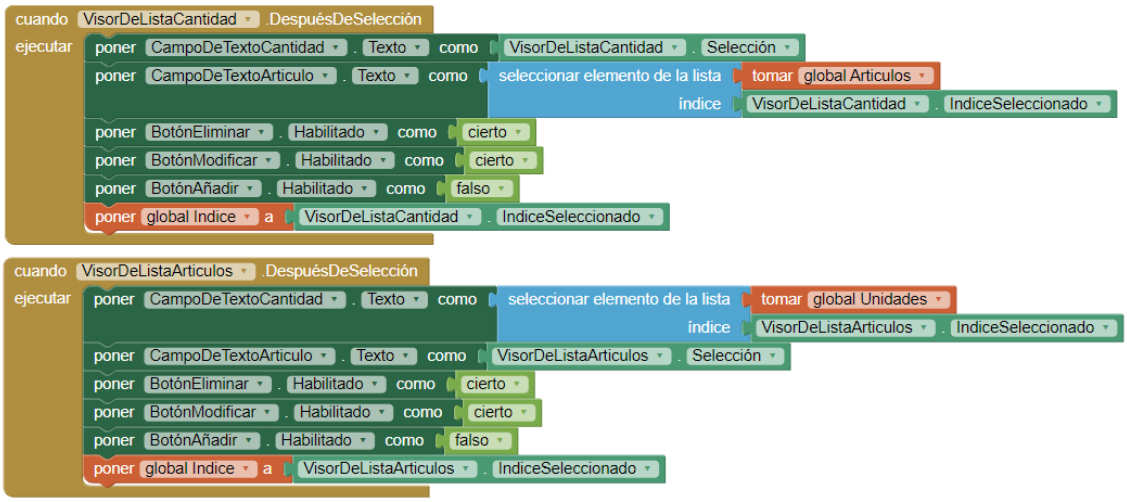


Con el BotónAñadir al hacer Clic comparamos que CampoDeTextoCantidad y CampoDeTextoArticulo no estén vacíos.

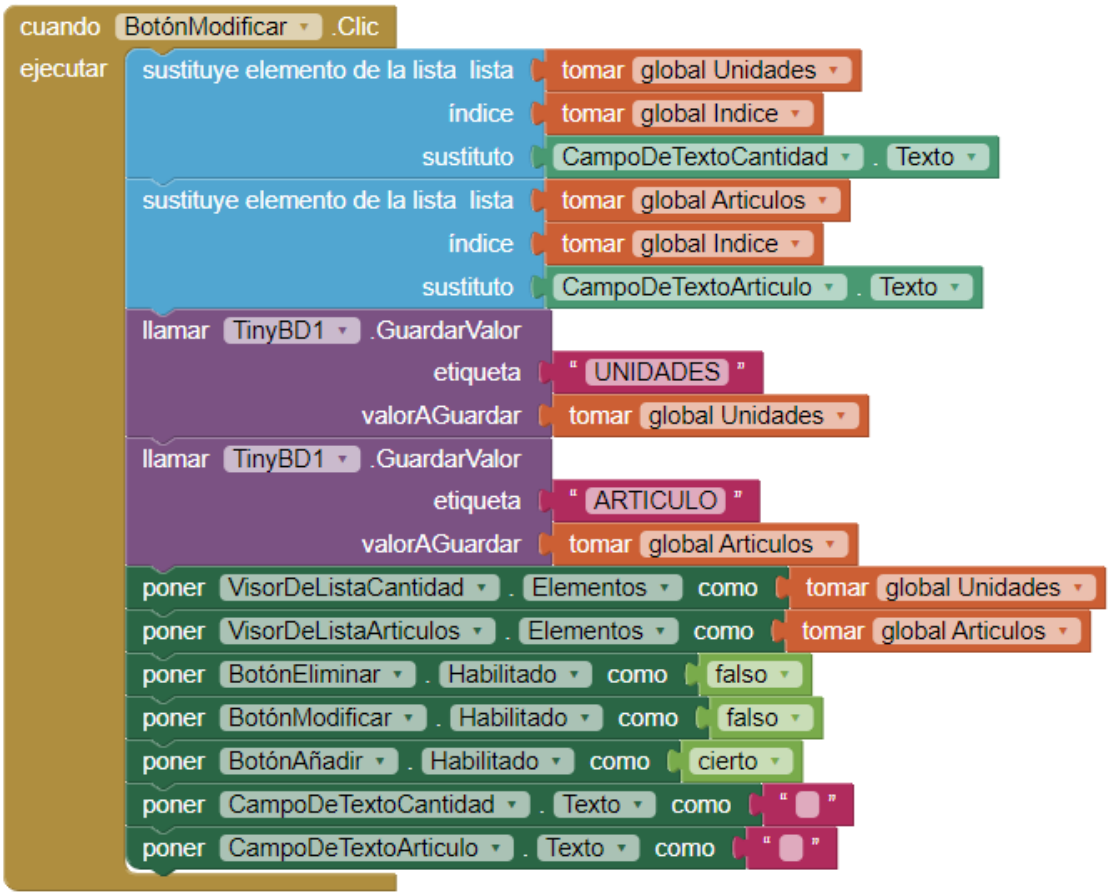
Si están vacíos nos avisará con una notificación de arleta.

De lo contrario se agregarán los valores a las variables de tipo lista Unidades y Artículos.

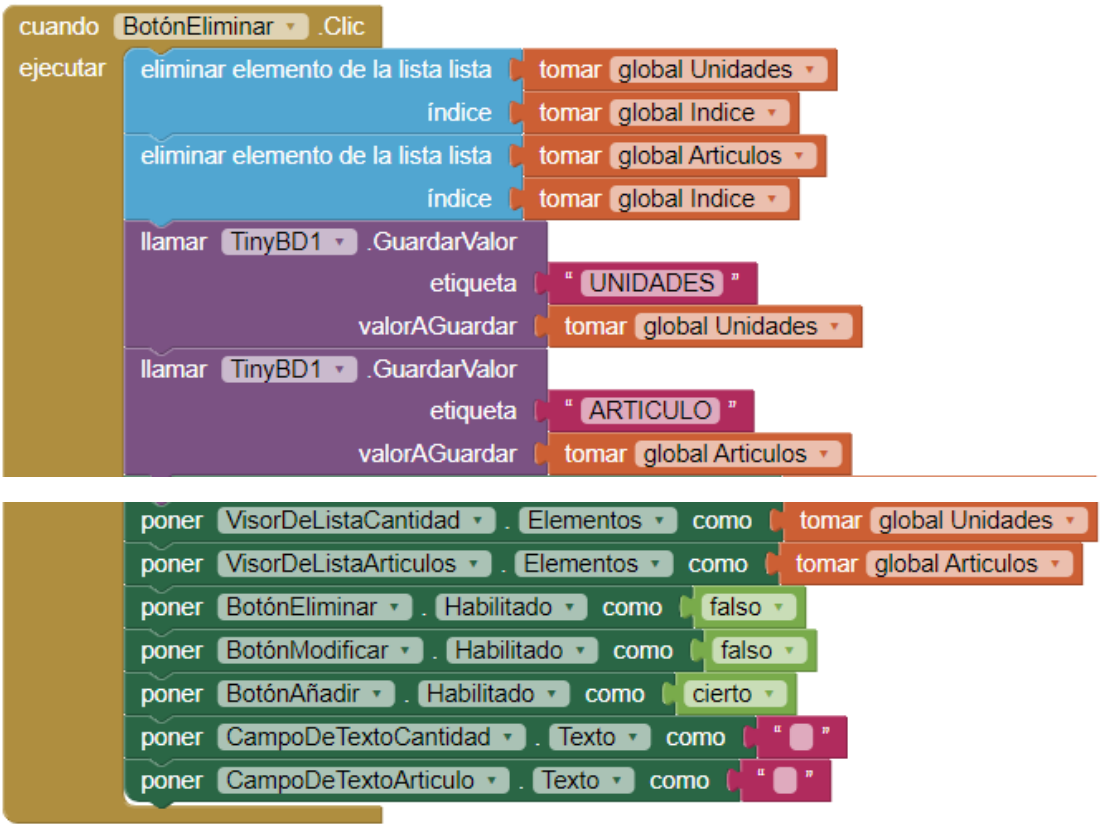
Tenemos dos Visores, uno para las Unidades y el otro para los Artículos, muestra el artículo que hemos añadido.
Los guarda en la base de datos con las etiquetas UNIDADES y ARTICULO.
Los CampoDeTextoCantidad y CampoDeTextoArticulo los deja vacíos.



Después de seleccionar alguno de los dos visores la información seleccionada se muestra en los campos CampoDeTextoCantidad y CampoDeTextoArticulo.
Se habilitan los botones BotónEliminar y BotónModificar, dejando inhabilitado el botón BotónAñadir, ya que nos encontramos en una situación que queremos modificar los datos de un elemento o eliminarlo.
Si seleccionamos el BotónModificar:



Modificamos el valor de la lista Unidades y Artículo según el índice que hemos seleccionado.
Lo grabamos en la base de datos con las etiquetas UNIDADES y ARTICULO.
Los cambios los mostramos en el VisorDeListaCantidad y VisorDeListaArticulos.
El BotónEliminar y BotónModificar los inhabilitamos.
El BotónAñadir lo habilitamos.
Los campoDeTextoCantidad y CampoDeTextoArticulo lo dejamos en vacío.
El BotónEliminar:



De la lista Unidades y Articulo eliminamos el elemento según el número de índice que adquirimos al seleccionar el visor y almacenado en la variable Indice.

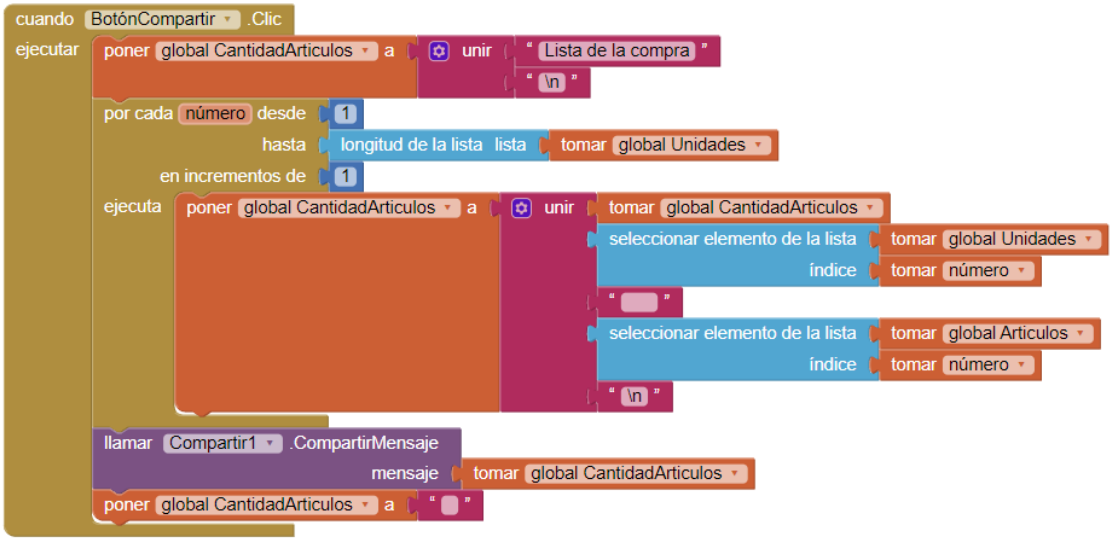
Guardamos las listas Unidades y Articulos con el elemento, sin el elemento que hemos elimiando.

Mostramos las nuevas listas al VisorDeListaCantidad y VisorDeListaArticulos.

Inhabilitamos los botones BotónEliminar y BotónModificar, para habilitar BotónAñadir.

Los campos CampoDeTextoCantidad y CampoDeTextoArticulo los dejamos como vacíos.

Cuando BotónCompartir.Clic.



La variable CantidadArticulos guardamos toda la información de las dos variables Unidades y Articulos recorriendo por todos sus elementos y con sus respectivos \n salto de línea y su cabecera de título “Lista de la compra”, que al final lo compartimos con otras aplicaciones de nuestro móvil.

Si seleccionamos el BotónVaciarLista.Clic.

Nos avisa con una notificación de que se va a eliminar la lista de la compra, esperando que el usuario selección el botón Sí o el botón No.

Si Notificador.DespuésDeSelección comprobamos si la respuesta a sido Si en este caso borramos todos los datos de la base de datos, eliminamos los elementos de la lista Unidades y Articulos, actualizamos el visor con los nuevos valores, como no tienen ninguno el VisorDeLitaCantidad y VisorDeListaArticulos se mostrará vacía.

Contenido