



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**Dirección General de Educación Tecnológica
Industrial y de Servicios**

Dirección Académica e Innovación Educativa

Subdirección de Innovación Académica

Departamento de Planes, Programas y Superación Académica

Cuadernillo de Aprendizajes Esenciales

Anexos del Módulo V, Submódulo 2

Ofimática



ANEXOS

ANEXO 1

HISTORIA DE LA TECNOLOGÍA MÓVIL Y DEFINICIÓN, CARACTERÍSTICAS Y CLASIFICACIÓN DE LOS DISPOSITIVOS MÓVILES.

El teléfono móvil se remota a los inicios de la segunda guerra mundial donde ya se veía que era necesario la comunicación a distancia es por eso por lo que la compañía creo un equipo llamado handie Talkie itt2-16 que es un equipo que permite el contacto con las tropas vía ondas.

- Generación en 1979 se dio a los países asiáticos el nacimiento de la primera generación de celulares con tecnologías analógicas que utiliza ondas de radios.
- Generacion en 1990 cuando se consolidaron las computadoras y las redes informáticas.

Las tecnologías móviles son un medio de comunicación que a superado en la telefonía móvil son más fáciles y baratas de desplegar. El uso de las tecnologías móviles entre los habitantes de una población ayuda a disminuir la brecha digital existente. Las redes actuales de telefonía móvil permite velocidades medias competitiva en relación con las de banda ancha en redes fijas : 183 kops GSM.

La evolución de teléfono móvil de móvil ha perdido disminuir su tamaño y peso que nos hace la vida más fácil y que nos permite comunicarnos desde cualquier lugar.

Características de las tecnologías móviles.

El primer Smartphone de la historia fue 18 m simori fabricada en 1992 distribuido por E.E.U.U. entre agosto de 1994 y febrero de 1995.

Clasificación de los dispositivos móviles.

Debido a la gran variedad existentes de dispositivos móviles esto ha sido clasificado de la siguiente manera.

Dispositivos de propósito general de trabajo.

Actualmente dominados por las computadoras de escritorio computadoras portátiles, tablets y similares. Estos dispositivos pueden trabajar una variedad de información centradas en las tareas generales.

-Dispositivos con propósito de entretenimiento. Tales como el pad, PlayStation, Nintendo entre otros. Estos dispositivos tienen el principal propósito de entretenimiento (especialmente por medio de video juego) pero tiene como objetivo apoyar fácilmente el uso de otros medios.

VENTAJAS	DESVENTAJAS	USOS
Movilidad: son livianos. Conectividad: la inalámbricidad facilita el acceso y uso de Internet	Tamaño: el tamaño de la pantalla por lo tanto es menor el tamaño del teclado haciéndolo incómodo para el usuario.	Llamadas: Las llamada tecnología móvil ya que es muy práctico usarlo y bastante costeable.

CUESTIONARIO EJEMPLO

INSTRUCCIONES: Con tus conocimientos previos contesta las siguientes preguntas, como ejemplo.

1) ¿Qué es un dispositivo móvil?

Es un objeto donde lo puedes transportar a todos lados, por ejemplo: celular, laptop, etc.

2) ¿Cuáles son las características de un dispositivo móvil?

Apoyan a realizar tareas, agenda, contactos, llamadas realizadas.

3) ¿Tiene ventajas un dispositivo móvil? ¿Cuáles?

Si tienen ventajas. Son pequeñas, ligeras y se les puede instalar videojuegos.

4) Menciona alguna desventaja de un dispositivo móvil

Que se descarguen, si las mojas se estropean, y no aguantan las caídas

5) ¿Qué dispositivos móviles has utilizado?

Celulares, laptop, Tablet, etc.

ANEXO 2

PRESENTACION DE LA SIGNATURA

PRESENTACION DE LA ASIGNATURA

MÓDULO V	
Información General	
INSTALA Y OPERA EQUIPOS DE CÓMPUTO Y PERIFÉRICOS UTILIZADOS EN LA OFIMÁTICA 192 horas	// SUBMÓDULO 1 Instala y configura equipos de cómputo y periféricos 112 horas
	// SUBMÓDULO 2 Establece comunicación y gestiona información mediante el uso de dispositivos móviles 80 horas

ESTABLECE COMUNICACIÓN Y GESTIONA INFORMACIÓN MEDIANTE EL USO DE DISPOSITIVOS MÓVILES



PLANTEAMIENTO



- En el mercado existen una gran cantidad de aplicaciones móviles ya desarrolladas, pero existen también clientes potenciales que tienen la idea de alguna aplicación, los técnicos en ofimática tienen los conocimientos necesarios para diseñar y desarrollar dichas aplicaciones móviles.

M. en C.C. ARMANDO RERGIS RAMIREZ

3

PROYECTO



- El alumno de ofimática tendría que encontrar un cliente (docente, alumno, familiar o empresa) que requiera de sus servicios y desarrollarle una aplicación móvil.

DISPOSITIVOS MOVILES



M. en C.C. ARMANDO RERGIS RAMIREZ

5



Actualmente existen una gran cantidad de dispositivos móviles

PULSE SMARTPEN LIVESCRIBE DESKTOP LIVESCRIBE COMMUNITY

ANEXO 3

I. Conceptos Básicos INTRODUCCION

La unidad correspondiente al manejo del entorno de desarrollo móvil, se avoca en el uso de la tecnología móvil, herramientas de la plataforma y técnicas necesarias que permitan la construcción e implementación de desarrollo de programación de aplicaciones para ejecutarse en dispositivos móviles. El desarrollo de esta unidad proporcionará al alumno elementos básicos que le permitirán desarrollar las actividades y prácticas propias de esta competencia, así como:

- Diferenciar las características de los dispositivos móviles y la plataforma de desarrollo con base en tecnologías y herramientas para aplicaciones móviles, se encuentra estrechamente vinculado con el subsecuente resultado de aprendizaje sobre el uso del entorno de desarrollo de programación móvil de acuerdo a los componentes y herramientas del lenguaje para la construcción de aplicaciones, y para lograrlo se sugiere que el docente opere con los conceptos y habilidades para al diseño e implementación de soluciones orientadas a móvil.
- Analizar la información relacionada con el manejo del entorno de desarrollo móvil, precisando la utilidad de esta fase y ventajas para el desarrollo de soluciones orientadas a móvil.
- Demostrar las distintas tecnologías orientadas a móvil mediante la ubicación del mercado actual; diferenciación de plataformas o tecnologías móvil, dispositivos disponibles y sus características; así como los beneficios de los productos móviles y sus aplicaciones.

Los teléfonos móviles son ya parte esencial en nuestra forma de vida. Cada día son más los usuarios de estos terminales, y cada vez más pequeños. Hasta ahora, nos han acompañado a todas partes y nos han permitido comunicarnos con cualquier otro terminal, ya sea fijo o móvil. Aunque la comunicación telefónica por voz es el principal uso de estos terminales, pronto se han desarrollado nuevas formas de comunicación y otras capacidades en torno a nuestros pequeños teléfonos.

El primero y quizás más lucrativo hasta la fecha, fue el uso de la mensajería SMS (Short Message Service). Nuestros pequeños terminales nos permiten enviar mensajes cortos de texto (hasta un tamaño de 160 caracteres) que son enviados desde el terminal al centro servidor de mensajes cortos o SMSC (Short Message Service Centre), que a su vez se encarga de hacer llegar el mensaje al móvil destinatario.

Más tarde, aparecieron los terminales capaces de navegar por Internet, pero las limitaciones de la pantalla y de los teclados hacían inviable su uso con páginas web normales. Así nació la tecnología WAP, que nos permiten navegar por páginas especiales creadas en WML en lugar de en HTML. Sin embargo, las limitaciones de este medio, y quizás también por el elevado precio y la baja velocidad del servicio, han hecho que la tecnología WAP no se haya extendido tanto como su hermana mayor, la WEB. Para paliar las bajas velocidades -sin contar con la baja fiabilidad- de la tecnología GSM para la transmisión de datos, apareció la tecnología GPRS (General Packet Radio Service). GPRS nos ofrece una red de transferencia de datos sin hilos a una velocidad aceptable, tanto es así, que ahora se puede enviar grandes paquetes de información, como fotografías, música, e incluso video. A partir de aquí, se hace patente la necesidad de una nueva generación de móviles capaces de reproducir músicas más complejas y mostrar gráficos y fotografías en color. A la vez que aparecen estos móviles en el mercado, aparece el nuevo servicio de mensajes cortos llamado MMS (Multimedia Message Service). Gracias a MMS, además de texto, podemos enviar fotografías, sonidos, gráficos, etc. Pero aún estaba por llegar la verdadera revolución.

INTRODUCCIÓN A TELEFONÍA MÓVIL

Los recientes avances de las tecnologías de hardware, software y telecomunicaciones están alimentando una nueva generación de sistemas o herramientas tecnológicas que están generando un gran impacto dentro del sector empresarial. Estas nuevas herramientas permiten a usuarios

con dispositivos móviles conectarse a Internet a gran velocidad a través de la propia red de comunicación móvil o a través de redes locales inalámbricas.

Dentro de este nuevo escenario, el término "Tecnología móvil" hace referencia al conjunto de dispositivos y herramientas que nos dan la posibilidad de realizar una actividad determinada de manera ubicua, que normalmente se inscribe en un espacio físico, en otro lugar. Estos avances tecnológicos han impulsado a la Sociedad actual a inclinarse con una confianza creciente hacia las distintas posibilidades de explotar servicios antes desconocidos, como puede ser el comercio electrónico, el cual está demostrando las innumerables ventajas que este puede aportar al crecimiento económico. Los requerimientos de la población de la sociedad en cuanto al acceso remoto a la información contenida en bases de datos también han mostrado una evolución creciente, por lo tanto, es de interés estudiar y aportar soluciones a medida para brindar servicios que permitan satisfacer estos requerimientos de información.

La utilización del teléfono móvil ha traspasado fronteras inimaginables hace unos años atrás. Aportando la libertad de la "no dependencia" de cables para que las personas puedan comunicarse o acceder a Internet en casi cualquier lugar.

Dentro de un escenario corporativo, las tecnologías móviles permiten llevar a cabo tareas sin estar en la oficina, y poder efectuar múltiples actividades, favoreciendo el teletrabajo y dando solución a uno de los mayores 'canceres' de esta sociedad, la baja productividad,

EVOLUCIÓN DE LA TECNOLOGÍA MÓVIL BREVE HISTORIA DE LA TELEFONÍA CELULAR.

Martin Cooper fue el pionero en esta tecnología, a él se le considera como "el padre de la telefonía celular" al introducir el primer radioteléfono en 1973 en los Estados Unidos mientras trabajaba para Motorola; pero no fue hasta 1979 en que aparece el primer sistema comercial en Tokio Japón por la compañía NTT (Nippon Telegraph & Telephone Corp.) En 1981 en los países Nórdicos se introduce un sistema celular similar a AMPS (Advanced Mobile Phone System). Por otro lado, en los Estados Unidos gracias a que la entidad reguladora de ese país adopta reglas para la creación de un servicio comercial de telefonía celular, en octubre de 1983 se pone en operación el primer sistema comercial en la ciudad de Chicago. A partir de entonces en varios países se diseminó la telefonía celular como una alternativa a la telefonía convencional alámbrica. La tecnología inalámbrica tuvo gran aceptación, por lo que a los pocos años de implantarse se empezó a saturar el servicio, por lo que hubo la imperiosa necesidad de desarrollar e implementar otras formas de acceso múltiple al canal y transformar los sistemas analógicos a digitales para darle cabida a más usuarios. Para separar una etapa de la otra, a la telefonía celular se ha categorizado por generaciones. A continuación, se describen cada una de ellas.

LAS GENERACIONES DE LA TELEFONÍA INALÁMBRICA

La primera generación 1G.

La 1G de la telefonía móvil hizo su aparición en 1979, se caracterizó por ser analógica y estrictamente para voz. La calidad de los enlaces de voz era muy baja, baja velocidad [2400 baudios], la transferencia entre celdas era muy imprecisa, tenían baja capacidad [basadas en FDMA, Frequency Division Multiple Access] y la seguridad no existía. La tecnología predominante de esta generación es AMPS (Advanced Mobile Phone System).

La segunda generación 2G.

La 2G arribó hasta 1990 y a diferencia de la primera se caracterizó por ser digital. El sistema 2G utiliza protocolos de codificación más sofisticados y son los sistemas de telefonía celular usados en la actualidad. Las tecnologías predominantes son; GSM (Global System for Mobile Communications); IS-136 (conocido también como TIA/EIA-136 o ANSI-136) y CDMA (Code División Multiple Access) y PDC (Personal Digital Communications), éste último utilizado en Japón.

Los protocolos empleados en los sistemas 2G soportan velocidades de información mas altas para voz pero limitados en comunicaciones de datos. Se pueden ofrecer servicios auxiliares tales como datos, fax y SMS [Short Message Service]. La mayoría de los protocolos de 2G ofrecen diferentes niveles de encriptación. En los Estados Unidos y otros países se le conoce a 2G como PCS (Personal Communications Services),

La generación 2.5GV. Muchos de los proveedores de servicios de telecomunicaciones (carriers) se moverán a las redes 2.5G antes de entrar masivamente a 3G. La tecnología 2.5G es más rápida y más económica para actualizar a 3G, La generación 2.5G ofrece características extendidas para ofrecer capacidades adicionales que los sistemas 2G tales como GPRS (General Packet Radio System), HSCSD (High Speed Circuit Switched Data), EDGE (Enhanced Data Rates for Global Evolution), IS-136B, IS-95B, entre otros. Los carriers europeos y de Estados Unidos se moverán a 2.5G en el 2001. Mientras que Japón ira directo de 2G a 3G también en el 2001.

La tercera generación 3G

La 3G es tipificada por la convergencia de la voz y datos con acceso inalámbrico a Internet, aplicaciones multimedia y altas transmisiones de datos. Los protocolos empleados en los sistemas 3G soportan más altas velocidades de información enfocados para aplicaciones mas allá de la voz tales como audio (MP3), video en movimiento, video conferencia y acceso rápido a Internet, sólo por nombrar algunos, Se espera que las redes 3G empiecen a operar en el 2001 en Japón por NTT DoCoMo, en Europa y parte de Asia en el 2002, posteriormente en Estados Unidos y otros países. Los sistemas 3G alcanzaran velocidades de hasta 384 Kbps permitiendo una movilidad total a usuarios viajando a 120 kilómetros por hora en ambientes exteriores y alcanzará una velocidad máxima de 2 Mbps permitiendo una movilidad limitada a usuarios caminando a menos de 10 kilómetros por hora en ambientes estacionarios de corto alcance o en interiores. Entre las tecnologías contendientes de la tercera generación se encuentran UMTS (Universal Mobile Telephone Service), cdma2000, IMT-2000, ARIB[3GPP], UWC-136, entre otras. El impulso de los estándares de la 3G está siendo apoyado por la ITU (International Telecommunications Union) y a este esfuerzo se le conoce como IMT-2000 (International Mobile Telephone).

La cuarta generación 4G

La cuarta generación es un proyecto a largo plazo que será 50 veces más rápida en velocidad que la tercer generación. Se realizaron pruebas de esta tecnología desde el 2005 y se comercializaron la mayoría de los servicios desde el 2010.

EL ESTADO ACTUAL DE LA TELEFONÍA MÓVIL

Existen hoy en día tres tecnologías de telefonía celular predominantes en el mundo: IS-136, IS-95yGSM.

IS-136 (Interim Standard 136) fue la primer tecnología digital de telefonía celular (D-AMPS, versión la versión digital de AMPS) desarrollada en Estados Unidos, IS-136 esta basada en TOMA (Time División Múltiple Access), una técnica de acceso múltiple la cual divide los canales de radio en tres ranuras de tiempo, cada usuario recibe en una ranura diferente. Este método permite a tres usuarios en cada canal de radio comunicarse sin interferirse uno con el otro. D-AMPS (IS-54) es utilizado principalmente en Norteamérica, Latinoamérica, Australia, partes de Rusia y Asia.

Por otro lado, CDMA, tecnología desarrollada por Qualcomm, utiliza la tecnología de espectro disperso en la cual muchos usuarios comparten simultáneamente el mismo canal pero cada uno con diferente código. Lo anterior permite una mayor capacidad en usuarios por celda. A CDMA de segunda generación se le conoce como cdmaOne. Hasta diciembre del 2000 existen más de 27 millones de usuarios en más de 35 países alrededor del mundo utilizando cdmaOne. GSM (siglas derivadas originalmente de Groupe Spéciale Mobile) es tecnología celular desarrollada en Europa considerada como la tecnología celular más madura, con más de 200 millones de usuarios en mas de 100 países alrededor del mundo. GSM es un servicio de voz y datos basado en conmutación de circuitos de alta velocidad la cual combina hasta 4 ranuras de tiempo en cada canal de radio.

INTRODUCCIÓN A LAS TECNOLOGÍAS Y HERRAMIENTAS MÓVILES.

Las tecnologías móviles se han convertido en una necesidad para la vida diaria de muchas personas. Hoy más que nunca, vemos una gran cantidad de gente utilizando las aplicaciones de los dispositivos de cómputo móvil para llevar a cabo sus tareas tanto de trabajo o de estudio como de entretenimiento. El desarrollo de aplicaciones para estos dispositivos se ha convertido en un nicho de oportunidad para los profesionistas de la computación.

El uso acelerado de tecnologías móviles ha impulsado una revolución en la manera de llevar a cabo las tareas de trabajo, estudio o simplemente de entretenimiento. Hoy en día es de lo más común ver a las personas en los restaurantes, transporte público, parques, supermercados, etcétera, llevar a cabo tareas tan simples como revisar una lista de pendientes o tan complejas como realizar una reunión de trabajo virtual.

Existen tres plataformas móviles que dominan el mercado actual tanto de teléfonos inteligentes (smartphones en inglés) como de tabletas (tablets en inglés): iOS, Android y Blackberry.

iOS es el sistema operativo utilizado por los dispositivos (iPod, iPhone, iPad) creados por la compañía Apple.

Android es el sistema operativo creado por la compañía Google y utilizado en teléfonos inteligentes y tabletas de varias compañías como HTC, LG, Motorola, Samsung, Sony, entre otras. Blackberry OS es el sistema operativo utilizado por los teléfonos inteligentes de la compañía Research In Motion (RIM), la cual también es creadora de la tableta PlayBook que utiliza el sistema operativo Blackberry Tablet OS.

ANEXO 4

LENGUAJES PARA PROGRAMAR EN ANDROID

<http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android/>

La gente de Google ha escogido el lenguaje Java para dar soporte a aquellas personas y empresas que deseen realizar aplicaciones de forma “nativa” en la plataforma de Android. En su renovada página developer.android.com podrás encontrar toda la información que necesitas para poder hacer tus aplicaciones, ya que cuentas con tutoriales, foros, videos y códigos de ejemplo sobre cada API de Android; digamos que es ello máximo, para documentarse sobre programación en Android.



Existen diversos ambientes para programar rápidamente en Android diferentes a Java, incluso se pueden hacer aplicaciones sin teclear una sola línea de código. El lenguaje que ofrece oficialmente Google para sus desarrollos, es Java y debido a que es del cual existe más información, en donde las actualizaciones están al día (como por ejemplo con Jelly Bean, que ya lo puedes descargar), y con el cuál más gente en el mundo trabaja.

Basic4Android

Basic4Android es una plataforma de programación para aplicaciones Android cuyo lenguaje base de programación es VisualBasic, el eterno rival de Java, ese lenguaje que está orientado a aquellas personas que empezamos en el mundo de la programación de una manera más gráfica y no tan abstracta. No es el mismo lenguaje de Microsoft, pero su sintaxis es la misma, lo cual tiene sus mismas ventajas como algunos de sus inconvenientes.

En la página oficial se encuentra toda la documentación de este ambiente de programación para Android, <http://www.basic4ppc.com> aquí se encuentra la sección de descargas, documentación, foros, características que lo hacen sobresalir sobre otros ambientes, así como la sección de pago, esta plataforma no es gratuita, encontramos desde la versión mínima pago que es de 49 dólares hasta la versión máxima de 249 dólares americanos. ¿Cuál es la diferencia entre estas versiones? La diferencia es el tiempo que se tiene de actualizaciones, en la de menor precio existe el completo acceso a la plataforma, pero con solo 2 meses de actualizaciones, y la de mayor costo se tienen 2 años de actualizaciones, lo que quiere decir por ejemplo que si se tiene una licencia de menor precio y en 3 meses sale la actualización por ejemplo de las librerías para el uso del NFC ya no tendrá acceso a esta librería. Se puede descargar una versión completa de prueba por 30 días, y tú decides si la compras o no <http://www.basic4ppc.com/android/downloads.html>

Mono para Android

Otro de los lenguajes que Microsoft desarrollo para hacer aplicaciones fue C# y .NET, las cuales son muy usados en diferentes ambientes, por lo que no podría faltar que estos lenguajes tan comunes y opuestos a Java llegaran a Android.

Si se programa con Visual Studio lo único que debe instalar es el SDK de Android, la versión para Android de Mono y listo amigo, sigue desarrollando sin ningún inconveniente; además según Xamarin (la empresa creadora de Mono), trabajas con un lenguaje nativo para Android ya que no tiene un intérprete como lo tendría Basic4Android, y su aprendizaje es relativamente sencillo en un tiempo prudente. Por otro lado, está el tema del costo, que para la versión más económica de Mono es de 399 dólares americanos y la más cara ni les cuento. Aquí también podrás



Create amazing Android apps with C# and .NET

Bringing joy to Android development.

CROSS PLATFORM
Easily share code between iOS, Android and Windows Phone 7.

NATIVE CODE
High-performance compiled code with full access to all the native APIs.

C# & .NET
Everything you love about .NET, including LINQ, Delegates and Events.

tener 30 días con una versión libre para que pruebes y evalúes si es lo que necesitas para desarrollar ese proyecto que tienes en mente <http://xamarin.com/trial>. Este es un video de cómo es su uso tan rápido e intuitivo. Como se puede ver en el video el cual no contiene largas líneas de código para hacer algo rápido, es algo muy sencillo de programar y muy eficiente sin



**Basic
4Android**

estar pensando en Java y HTML al mismo tiempo, por eso es conveniente descargarlo en calidad de prueba en sus 30 días gratis.

App Inventor

¿No se quiere Java, ni C#, ni C, ni .NET, ni VisualBasic, en resumidas cuentas, ningún programa de desarrollo tradicional? Para diseñadores, principiantes de aplicación para Android sin tener necesariamente un background técnico, para ellos está dirigido el App Inventor. Esta plataforma de desarrollo está basada en un lenguaje de desarrollo gráfico en donde no escribes ni una sola línea de código, tan solo arrastras bloques identificados con la acción que necesitas hacer y listo. Esta plataforma de desarrollo fue impulsada por Google hace un tiempo con el fin de que más personas se unieran a la familia de Android; esta genial herramienta usa el navegador como centro principal de trabajo, y almacena todo esto en servidores que están disponibles cada vez que entres a internet. Intentar explicarlo con palabras es más difícil que usar App Inventor, mejor te invito a que veas como se hace una app en menos de 5 minutos en el siguiente video. Si no se quiere realizar algún pago, solo se descarga lo necesario y listo, se arrastran bloques y ya. ¿Se quieres conectar la aplicación con Twitter? Ok, se arrastra el bloque de Twitter. ¿Usar el Bluetooth? Ok, entonces se toma el bloque de Bluetooth y se conecta de una vez. Así de sencillo.

Es muy fácil de empezar, la página oficial es <http://explore.appinventor.mit.edu/> en donde se encuentra toda la documentación y tutoriales respectivos para empezar, y con un móvil Android podrá ver como al arrastrar un elemento al entorno de trabajo se puede ver al mismo tiempo en el dispositivo, algo así como “programación instantánea”.

LiveCode

Una plataforma en la que puede programar tanto para Android, iOS, Windows, Linux, iPhone, iPad, Web y para Servidores con una sola plataforma de trabajo. Esta es la plataforma de LiveCode. Cualquier aplicación, diseñada, trabajada, montada en el servidor y con miles de visitas diarias. Ahora se quiere llevar esa misma aplicación a Windows y a iOS para cubrir un mercado más grande; en ese momento se debe pensar ya en por lo menos un programador para cada plataforma. En vez de pensar en 5 entornos de desarrollo, porque no, mejor en una plataforma de desarrollo, que haga todo para todas las plataformas con unos sencillos clicks, eso en resumidas cuentas es LiveCode. El lenguaje de programación que usa LiveCode se llama “Programación Orientada a Eventos”, y se basa en arrastrar elementos a un área de trabajo y programar los eventos que estén vinculados a este elemento; por ejemplo, si arrastras un botón al área de trabajo, el evento que se tiene vinculado es el click sobre ese botón, o por ejemplo si arrastra una imagen, el evento vinculado es por ejemplo moverlo sobre la pantalla. Este programa también es de pago, dependiendo de que alcance se requiere tener en el desarrollo del proyecto y si se quiere llegar a los sistemas operativos más populares, la licencia va a ser mucho más costosa que si solo se necesita programar por ejemplo solo para Android; la licencia más económica es de 99 dólares americanos, pero existe una versión de prueba de 30 días, para una evaluación o pruebas, la página oficial es <http://www.runrev.com/home/> en donde se encuentra mucha más información al respecto.



In Design CS6



Para las personas que escribe libros, revistas, comics, catálogos, periódicos, y requieren que sus publicaciones estén de una manera rápida, limpia en los dispositivos móviles Android o iOS. Entonces la solución, desde CS6 se podrán hacer las aplicaciones que se deseen sin escribir una sola línea de código. Para esto Adobe da una solución muy sencilla, solo se debe hacer 3 cosas:

1. Tener InDesign CS6 instalado en la computadora.
2. Descargar Adobe Content View para administrar el trabajo apropiado y organizado para llevarlo a las tablets.
3. Comprar una licencia dependiendo de cuántas publicaciones vaya a tener la revista o periódico o cómic que desee realizar.

El resto es por arte de magia, solo hacer trabajo de diseño y en unos sencillos pasos, la publicación ya está lista para que las personas la compren en la tienda digital ya sea para Android o iPad. Para ver más información al respecto, entrar a:

<http://blogs.adobe.com/indesigndocs/2011/02/design-decisions-for-digital-publishing-apps.html>

O en: http://help.adobe.com/en_US/digitalpubsuite/using/WS9293e1fb3b977c5c-925f33a13240a89d94-8000.html

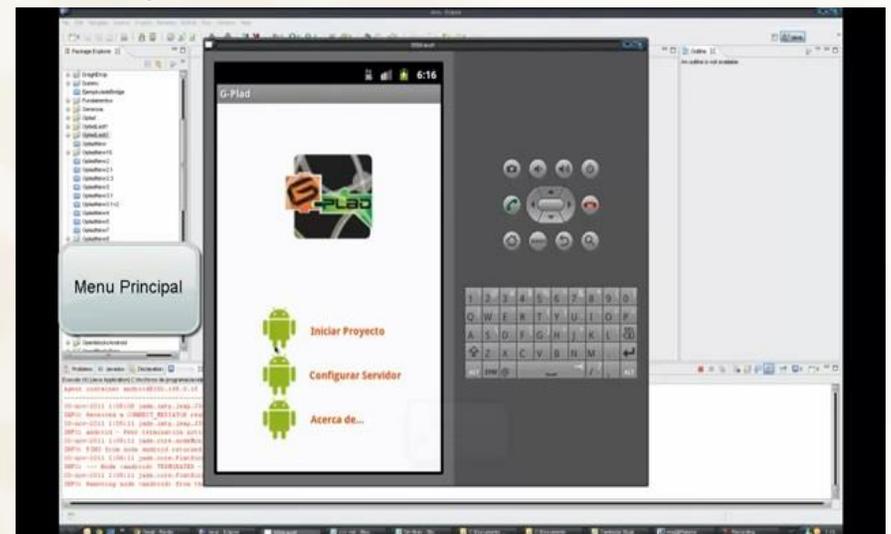
Gplad

Este es un proyecto desarrollado en la Fundación Universitaria San Martín. Se trata de un lenguaje de programación gráfico para dispositivos Android, que se inspira en la **StarLogo TNG**, un lenguaje de programación desarrollado por el Programa MIT Scheller.

La idea de los lenguajes gráficos, y Gplad no es la excepción, es la construcción de soluciones a los problemas de programación simples en una interfaz de programación basada en bloques que contiene todas las estructuras principales de los lenguajes de programación, como condicionales (*if, then*), las definiciones de las variables (*string, integer*) y bucles (*for, while*).

Aparte de escribir en este lenguaje gráfico los algoritmos, una vez creada la solución a un problema en particular, se puede ver el código en un lenguaje como Java. Dicho de otra manera, el programa en gráficas tiene una representación en modo texto, como cualquier lenguaje de programación.

Toda la programación gráfica se basa en una interfaz en donde las instrucciones del lenguaje son bloques. Hay una pantalla principal que contiene el espacio de trabajo, donde las formas (ventanas y bloques) se pueden arrastrar para empezar a crear nuestra solución al problema. Esta pantalla contendrá un menú que contendrá todas las formas posibles para



comenzar, en este caso, las principales estructuras del lenguaje. Se pueden entonces ir pegando los bloques con las instrucciones. Ver el siguiente video:

http://www.youtube.com/watch?feature=player_embedded&v=qn6EEE3W7Iq

En el fondo la idea de todo proyecto de programación gráfica se basa en la problemática que tienen las personas para atacar el aprendizaje de los lenguajes de programación. Una dificultad común es la necesidad de aprender la sintaxis específica de los lenguajes (¿acaba cada línea con ';'? , por ejemplo, o bien, cuando omitimos cerrar paréntesis). Por una parte, los aprendices se frustran de que un detalle impida que el programa funcione; por otra, muchas veces se requiere aprender una lógica, una manera de pensar para resolver los problemas de programación, por lo que, además de la sintaxis del lenguaje en particular, se requiere aprender un tipo de lógica muy particular.

Cabe señalar que Gplad y este **sitio web** se encuentran en fase de desarrollo pre-alpha. Los autores han prometido añadir más información, así como el código del proyecto tan pronto esté listo para ser lanzado. Habrá que ver si este esfuerzo prospera.

Referencias: [Sitio oficial de Gplad](#) y [All things now](#)

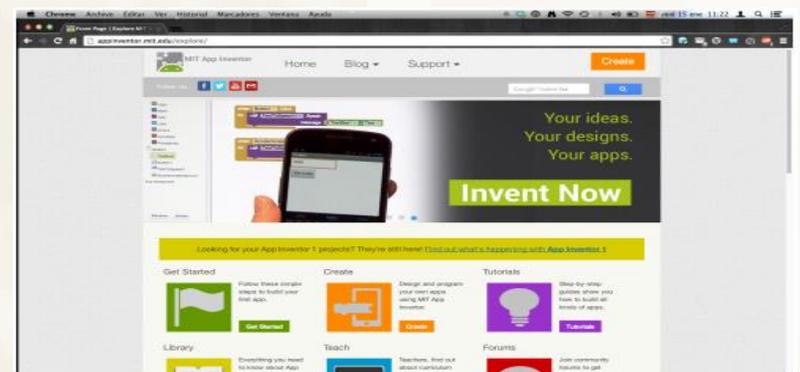
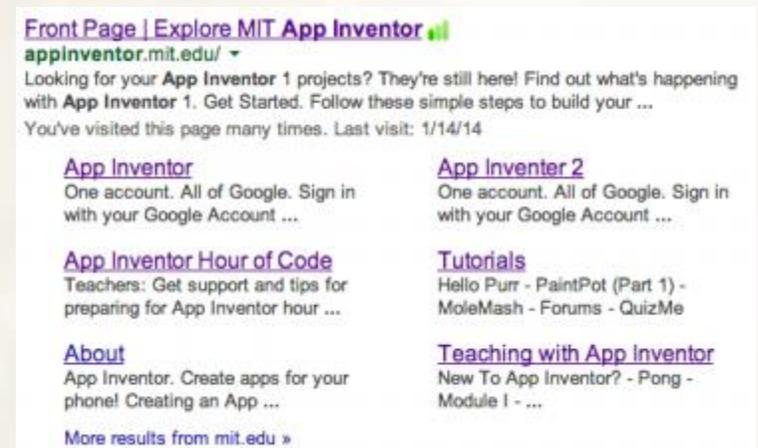
ANEXO 5

¿QUÉ ES APP INVENTOR?

App Inventor parte de una idea conjunta del Instituto Tecnológico de Massachusetts y de un equipo de Google Education. Se trata de una herramienta web de desarrollo para iniciarse en el mundo de la programación. Con él pueden hacerse aplicaciones muy simples, y también muy elaboradas, que se ejecutarán en los dispositivos móviles con sistema operativo Android. App Inventor es un lenguaje de programación basado en bloques (como piezas de un juego de construcción), y orientado a eventos. Sirve para indicarle al “cerebro” del dispositivo móvil qué queremos que haga, y cómo. Es por supuesto muy conveniente disponer de un dispositivo Android donde probar los programas según los vamos escribiendo.

Objetivos 1. Crear una cuenta Google 2. Instalar App Inventor 2 3. Instalar MIT AI2 Companion en el dispositivo Android, o conocer el emulador 4. Crear una aplicación e instalarla en el móvil 5. Conocer la página web de App Inventor (tutoriales, ejemplos, guías etc.)

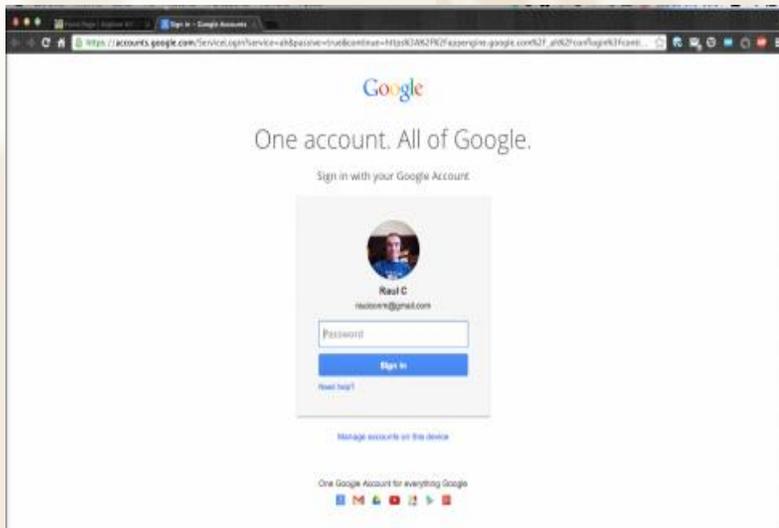
Crear una cuenta Google Es necesario crear una cuenta Google porque App Inventor es un trabajo conjunto entre Google y el MIT (<http://web.mit.edu/>). Recuérdese que Android es de Google. Abrir el navegador. OJO, tiene que ser Google Chrome, Safari o Firefox. Internet Explorer aún no es compatible con App Inventor. Ir a la página <https://accounts.google.com/> Utilizar una cuenta ya existente o crear una nueva. Nos hará falta para usar App Inventor.



Seguir las instrucciones de la página de Google para crear una nueva cuenta. Instalar App Inventor 2 Buscamos “App Inventor” en el buscador de Google. Hacer clic sobre Front Page | Explore MIT App Inventor.

Se abrirá la siguiente página.

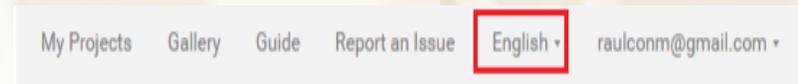
Hacemos clic sobre el botón Create. Si no hemos abierto sesión en Google, el navegador nos pedirá que lo hagamos ahora.



Al acceder a la cuenta de Google veremos nuestra página en App Inventor.

Configurar el idioma App Inventor nos muestra inicialmente el interfaz en inglés, sin embargo, podemos utilizarlo también en castellano. La elección del idioma se hace pulsando sobre el icono de la bola del mundo situado en la esquina superior derecha de la página de App Inventor.

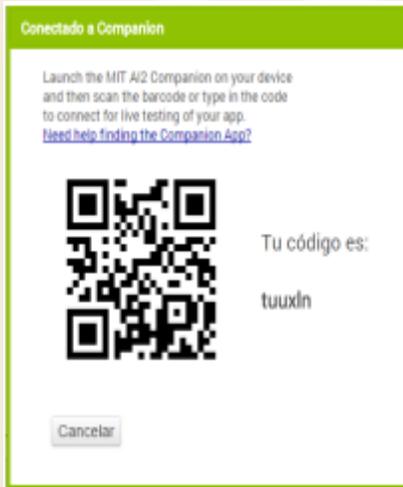
Una vez configurado en castellano pulsamos el botón Comenzar un proyecto nuevo... y le damos a nuestro proyecto el nombre.



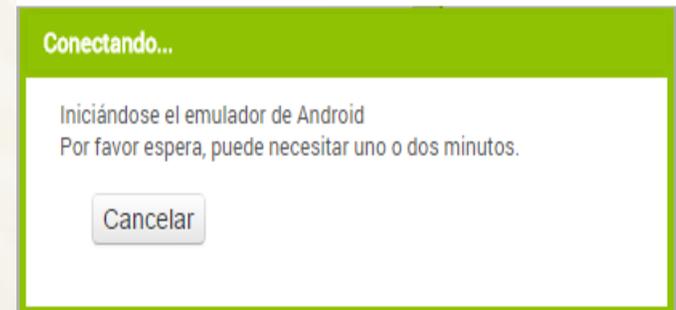
La pantalla que estamos viendo se divide en partes: - A la izquierda están los objetos que vamos a usar para diseñar la pantalla de nuestra aplicación. Botones, imágenes, dibujos, etc.

Es como la paleta de un pintor - Después explicaremos la parte derecha - ¿Qué es la pantalla del centro? Representa la pantalla del móvil, y sirve para DISEÑAR el aspecto de la aplicación. La llamaremos Visor Arriba a la izquierda hay un botón importante: Conectar. Para poder probar cómo funciona lo que vamos haciendo necesitamos transferirlo a un móvil, o usar el emulador incluido en App Inventor (esta opción es menos recomendable).

Para conectar App Inventor con el móvil hacer clic en Conectar, y elegir la opción AI COMPANION. Se abrirá una pantalla como esta:

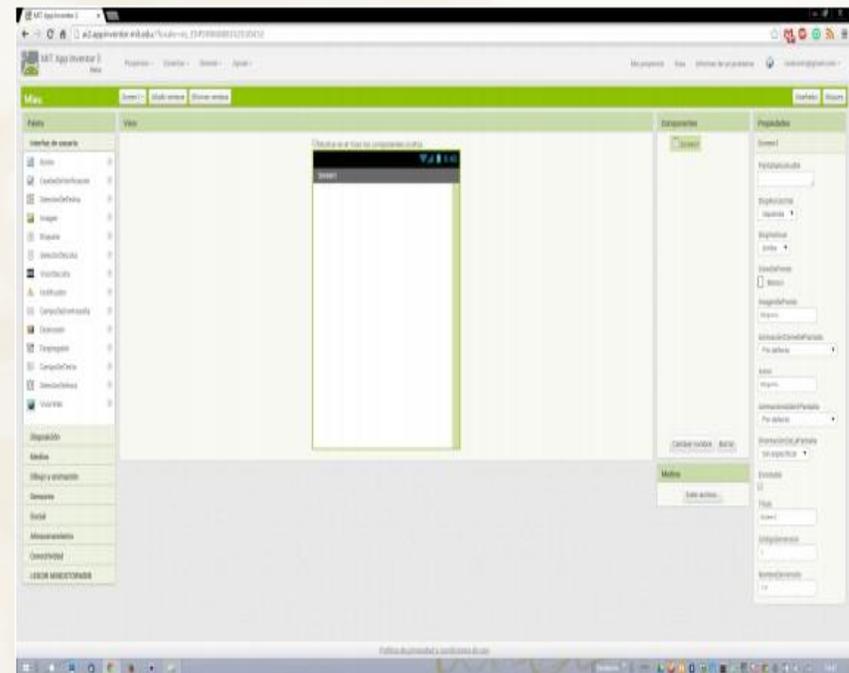


Instalar MIT AI2 Companion en el dispositivo Android, o conocer el emulador Para poder utilizar nuestro dispositivo como banco de pruebas tenemos que descargarnos de Google Play Store una aplicación que se llama "AI2 Companion". Buscarla en Google Play Store con ese nombre y descargarla ahora. Ocupa poco, y sólo hará falta descargarla una vez. Abriremos ahora en el móvil la aplicación que hemos descargado, haciendo clic sobre el icono MIT AI2 Companion. Puede estar en la página principal o dentro del grupo de Aplicaciones. Cuando se abra, elegiremos Connect with code (color naranja), y escribiremos el código de letras y números (alfanumérico) que aparece en la pantalla del ordenador. Quien pueda leer códigos QR puede hacerlo desde la pantalla



ahora pulsando en el botón azul scan QR code. Para que esto sea posible el ordenador y el dispositivo deben estar en la misma red, es decir, que deben tomar la IP del mismo rango, típicamente del mismo enrutador (router). Si no disponemos de WIFI podremos utilizar el emulador, o una conexión USB (ver detalles para USB en la web de App Inventor). Para abrir el emulador, haremos clic sobre Conectar y elegiremos la opción Emulador. No hay que hacer nada más, tarda un poco, pero una vez que cargue se verá la pantalla en blanco de nuestra aplicación.

Una vez establecida la conexión entre App Inventor y el móvil veremos una pantalla en blanco con el título Screen1.



ANEXO 6

ENTORNO GRAFICO DE APP INVENTOR

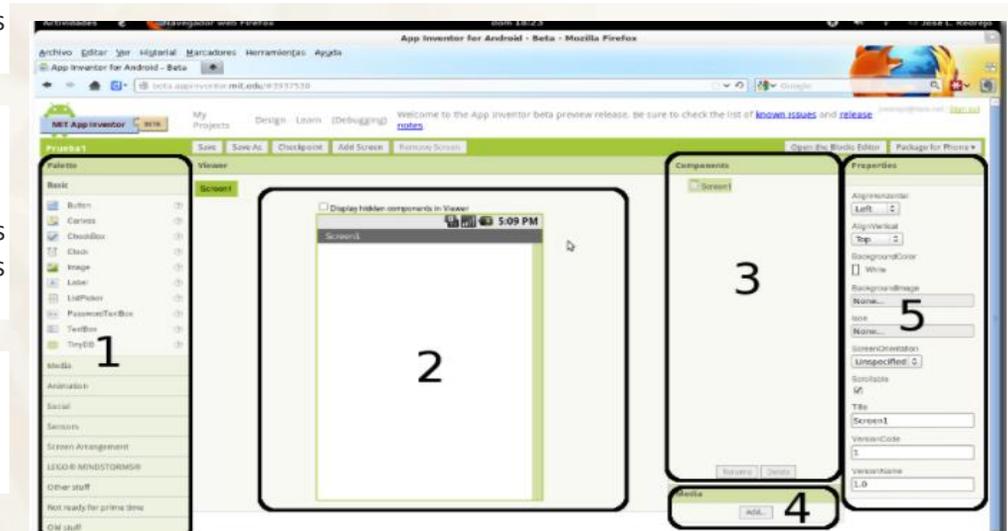
Se trata de la ventana del diseñador en la que se construye, mediante el ratón, sin necesidad de usar ningún lenguaje de programación, todo el interfaz gráfico que va a tener la aplicación. En ella se han marcado con números cada una de sus partes principales:

1. La *paleta* contiene todos los elementos que podemos insertar en nuestra aplicación. Hay elementos gráficos como cuadros de texto, botones, lienzo de dibujo (Canvas) y elementos que no se ven en la pantalla del móvil, como base de datos (TinyDB), acelerómetro, cámara de vídeo, etc.
2. *Viewer*: El visor de la pantalla, simula la apariencia visual que tendrá la aplicación en el móvil. Para añadir un elemento a la pantalla hay que arrastrarlo desde la paleta y soltarlo en el visor. Los elementos que no tengan visibilidad hay que arrastrarlos también al visor y automáticamente se desplazarán debajo de él bajo el epígrafe “Non-visible components”
3. *Components* muestra la lista de los componentes que se han colocado en el proyecto. Cualquier componente que haya sido arrastrado y soltado desde la paleta al visor aparecerá ahí. Si se quiere borrar alguno es en la lista de componentes donde está el botón que permite borrarlo.
4. *Media* muestra las distintas imágenes y sonidos que estarán disponibles para el proyecto. Cualquier archivo de imagen o audio que se quiera usar en la aplicación hay que insertarlo usando este apartado para que esté disponible.
5. *Properties*: cada vez que en el Viewer se seleccione un componente, en Properties aparecerán todos los detalles que se puedan cambiar de ese componente. Por ejemplo, al hacer clic sobre un componente TextBox se podrá cambiar en Properties su color, texto, fuente, etc. Para aquellos que hayan usado antes entornos de desarrollo del tipo de Visual Studio en Windows o Gambas en Linux le será muy familiar esta forma de trabajar.

COMO USAR APP INVENTOR

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

- o GNU/Linux: Ubuntu 8+, Debian 5+
 - o Macintosh (con procesador Intel): Mac OS X 10.5, 10.6
 - o Windows: Windows XP, Windows Vista, Windows 7
- Necesita también de conexión a Internet (los programas y las aplicaciones se cargan siempre desde Internet) y uno de los siguientes navegadores web con estas versiones mínimas:
- o Mozilla Firefox 3.6, sin la extensión NoScript instalada
 - o Apple Safari 5.0
 - o Google Chrome 4.0
 - o Microsoft Internet Explorer 7



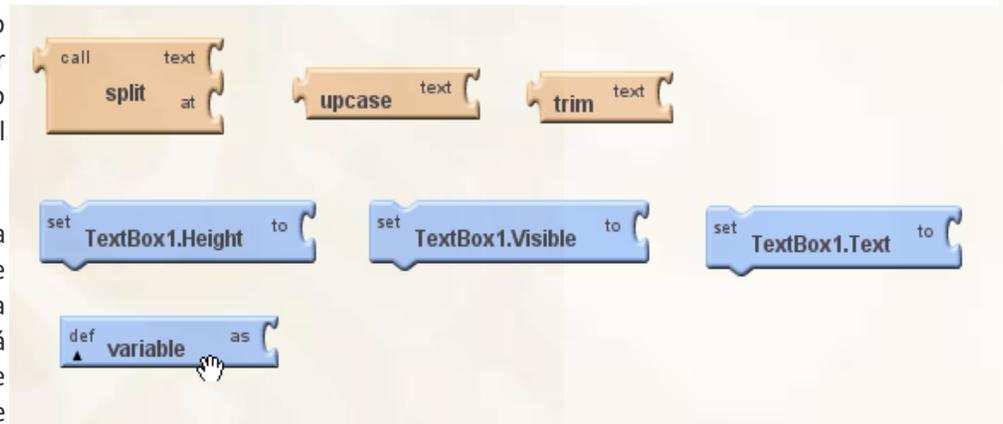
Se necesita también una cuenta de correo electrónico de Gmail y Java Web Start instalado en el ordenador.

El software para ejecutar los programas se descarga de <http://appinventor.mit.edu/explore/content/install-app-inventor-software.html> , donde se elegirá una opción u otra dependiendo del sistema operativo que se esté usando. En el caso de Macintosh o Windows se dispone del típico instalador para estos entornos. En sistemas Gnu/Linux Debian o derivados se proporcionan los paquetes deb de instalación⁴ o la opción de instalar directamente desde un archivo comprimido tar.gz⁵. Nota importante: En el caso de instalar en un sistema Linux de 64 bits es necesario asegurarse de que están instalados algunos paquetes para que el software funcione. En ⁶ se puede ver la lista de paquetes necesarios.

Este software permitirá no solo ejecutar los programas en nuestro móvil o en el emulador, sino que dispone además de un intuitivo interfaz de depuración que ayuda a encontrar errores de programación. Para que App Inventor pueda comunicarse desde el ordenador con el móvil usando el cable USB es necesario ir a los ajustes del móvil ->Opciones de desarrollador y activar la opción “Depuración de USB”.

Importante: Es conveniente instalar este software, pero no necesario para hacer los programas. Si no se instala saldrá un mensaje de error cada vez que se carga la aplicación de desarrollo avisando de que no encuentra el entorno de emulación, pero se puede realizar el programa igualmente.

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



ANEXO 7

Elementos de programación en App Inventor

Veamos algunos de esos bloques de App Inventor:

1.- **Sentencias** Al programar necesitamos a menudo decirle al ordenador que haga algo. En App Inventor existen numerosos bloques que son sentencias de programación. Se distinguen rápidamente porque expresan una acción a realizar con un verbo en imperativo. Los más habituales son *call*, *set* y *def* (abreviatura de define). Además, los bloques que son sentencias sólo están disponibles en dos colores: azul para modificar variables o propiedades de un objeto y naranja para llamar a funciones. En la imagen siguiente se ven algunos ejemplos de las sentencias para dividir un texto en partes, convertirlo a mayúsculas, cambiar la altura, texto o visibilidad de un recuadro de texto o definir una variable.

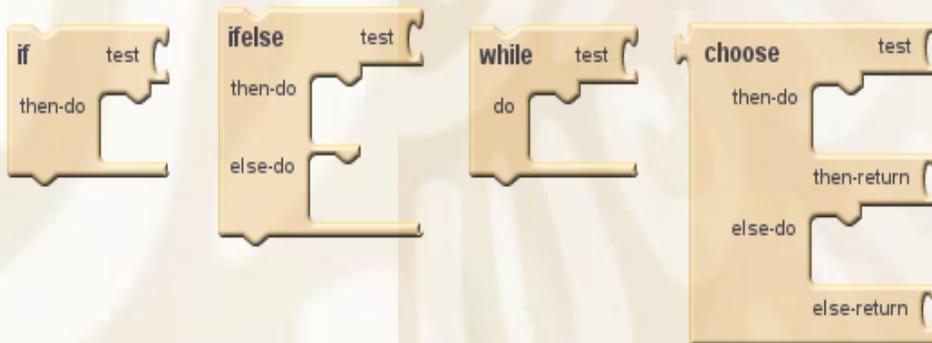
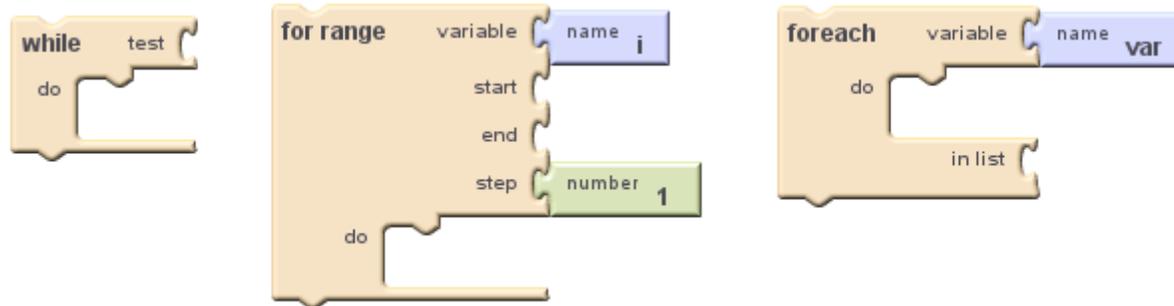
Condiciones

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

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

Bucles

Un bucle provoca la ejecución repetida de varias sentencias. En App Inventor disponemos de los bucles *while*, *for range* y *foreach*.



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

Variables

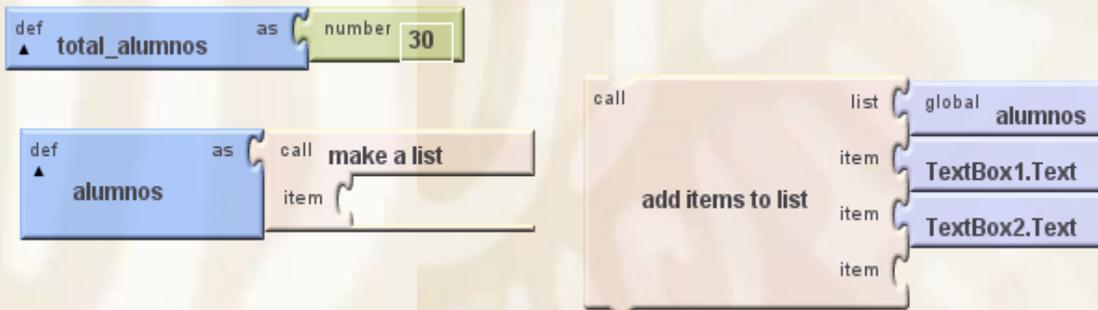
Disponemos de las herramientas para definir y cambiar variables. El tipo puede ser numérico o un texto y “se declara” al asignarle un valor por primera vez.

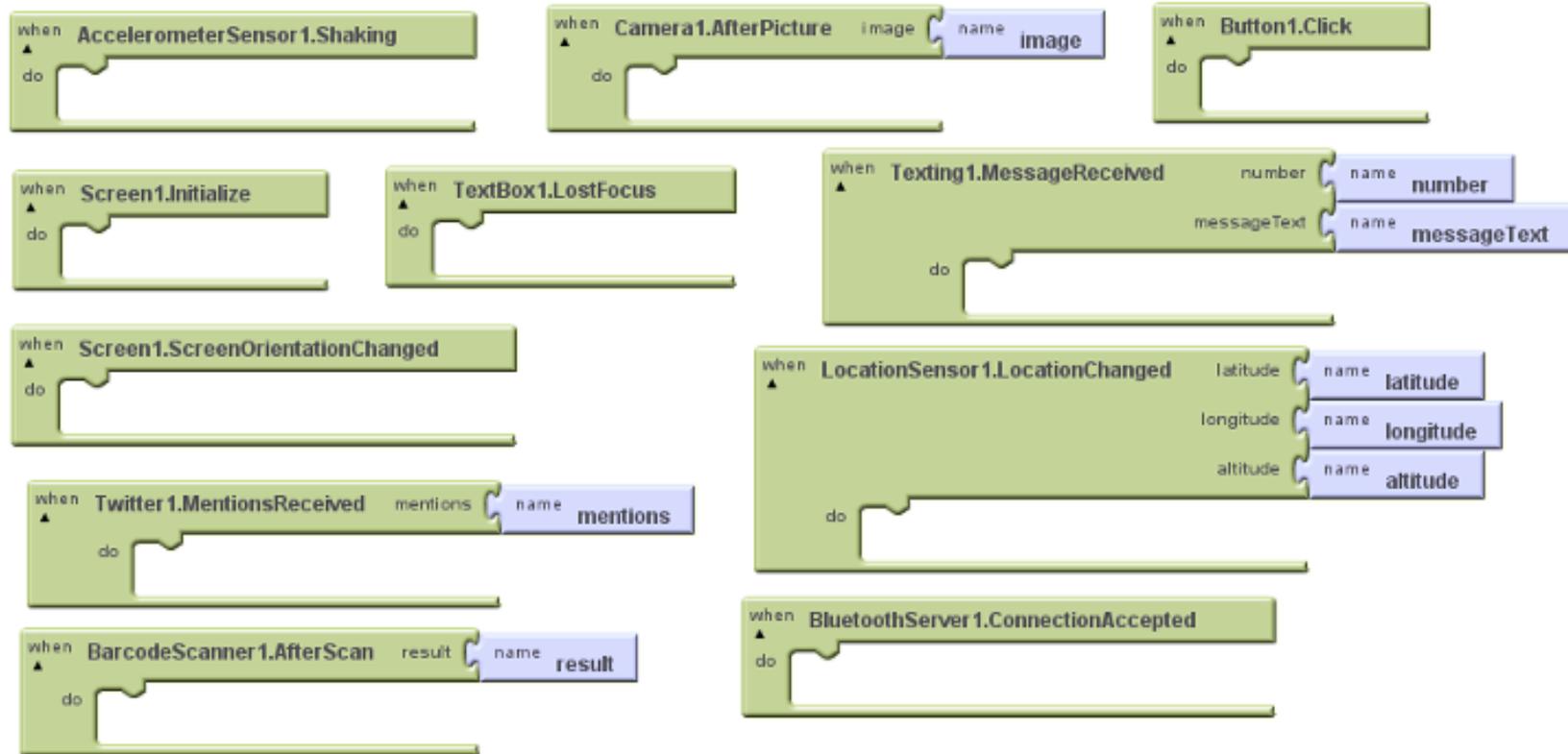
En el gráfico anterior se ve como se declara una variable llamada “total_alumnos” y se le da un valor numérico de 30. Además se ven dos bloques muy importante en App Inventor. Su lenguaje cuenta con una estructura de datos llamada *list* (lista) que es similar a los Arrays de otros lenguajes de programación.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

Eventos

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

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



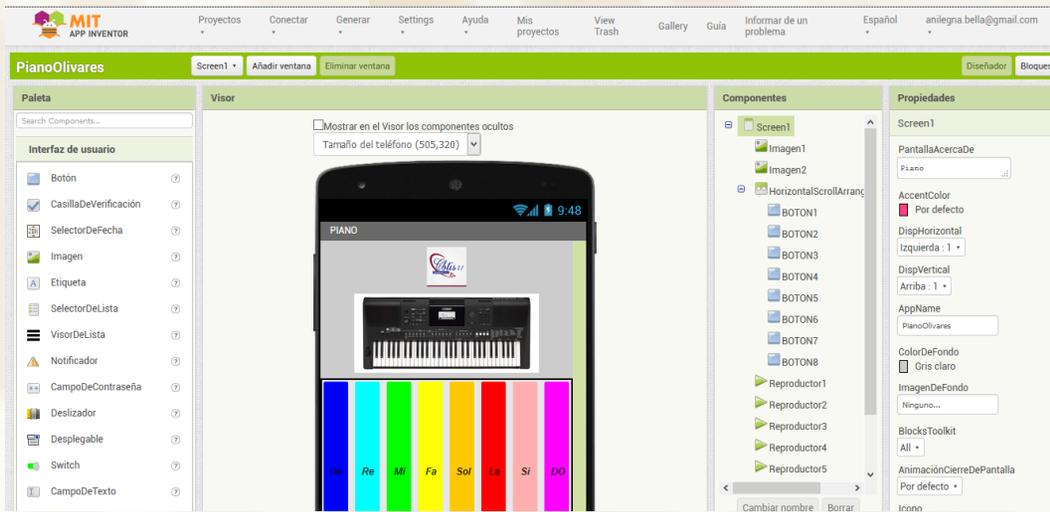


En la imagen anterior se pueden ver algunos de estos eventos. De esta selección se aprecia que con App Inventor no sólo se dispone de objetos que se pueden colocar en la pantalla como un botón o imagen, sino también de otros muchos objetos que nos dan acceso a las funcionalidades que tienen los teléfonos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

También se puede apreciar como todos los eventos tienen la estructura “When evento do”, y son de color verde. Tienen este color porque son parte de la sección “My blocks” en el entorno de desarrollo. “My blocks” es donde se leen o manipulan las propiedades de los distintos elementos que hayan sido colocados manualmente en el entorno de diseño: botones, brújula, etiquetas, cuadros de texto, etc.

ANEXO 8

Practica 1: CREAR UN PIANO



Diseño. Al terminar, cada color tiene un sonido.

cuando BOTON1 .Clic
ejecutar llamar Reproductor1 .Iniciar

cuando BOTON2 .Clic
ejecutar llamar Reproductor2 .Iniciar

cuando BOTON3 .Clic
ejecutar llamar Reproductor3 .Iniciar

cuando BOTON4 .Clic
ejecutar llamar Reproductor4 .Iniciar

cuando BOTON5 .Clic
ejecutar llamar Reproductor5 .Iniciar

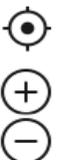
cuando BOTON6 .Clic
ejecutar llamar Reproductor6 .Iniciar

cuando BOTON7 .Clic
ejecutar llamar Reproductor7 .Iniciar

cuando BOTON8 .Clic
ejecutar llamar Reproductor8 .Iniciar

Bloques requeridos para el Diseño.

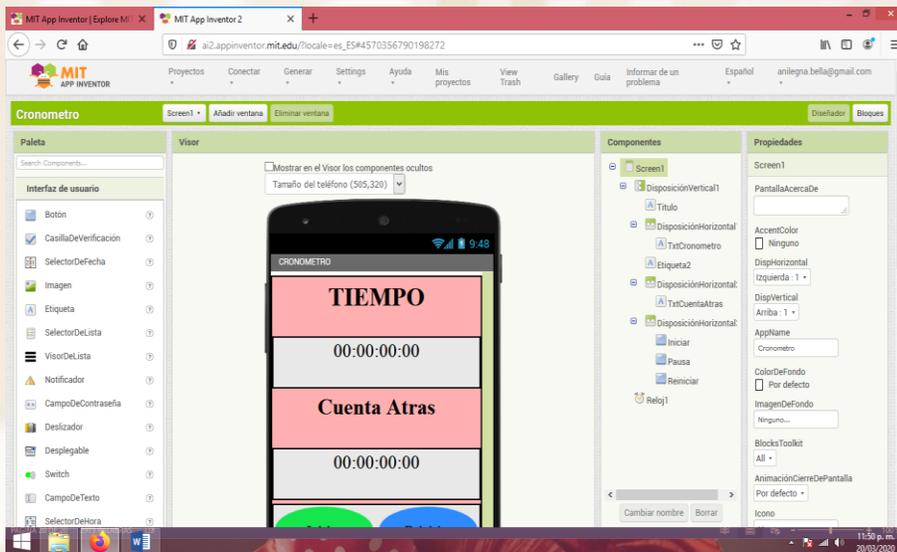
Instrucciones:



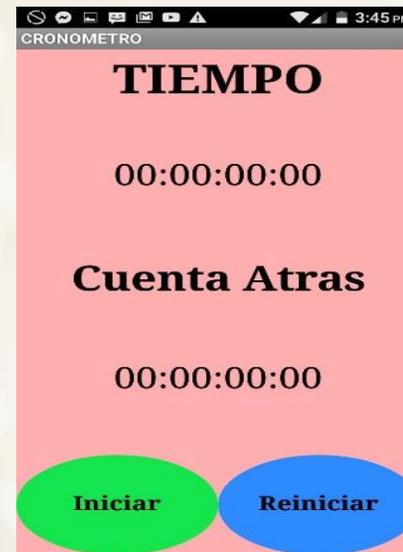
- Primero iniciamos creando un proyecto nuevo, al estar en nuestra plantilla o Screen como se le llama al programa, lo primero que aremos es adjuntar los archivos que usaremos en nuestro proyecto, en este caso, las imágenes y sonidos que usaremos.
- Una vez que lo hayamos conseguido, arrastraremos la disposición que usaremos, es mi caso, la horizontal, para a continuación arrastrar botones a la pantalla, ahí acomodaremos las propiedades de los botones, como medidas, texto, y color.
- Así mismo arrastraremos los reproductores necesarios a la pantalla (un reproductor por sonido), en propiedades adjuntaremos el sonido correspondiente a cada reproductor.
- Bueno hasta aquí terminamos el arte del diseño, ahora nos transportaremos a la parte de Bloques, para iniciar con la parte de programación de la app.
- En la parte de bloques, arrastraremos un bloque de control, que nos indique que cuando se hace clic en el botón seleccionado se ejecute la acción deseada.
- Para posteriormente, arrastrar un bloque de procedimiento, que nos indique que cuando se llama al reproductor, se inicialice el mismo, introduciremos es te bloque de procedimiento al bloque de control.

ANEXO 9

Practica 2: CREAR UN CRONOMETRO



Diseño



Resultado.

```

inicializar global Contador como 0
inicializar global Milisegundos como 0
inicializar global Cronometro como falso

cuando Screen1 Inicializar
ejecutar poner global Milisegundos a llamar Reloj1 Ahora

cuando Iniciar Clic
ejecutar poner global Cronometro a cierto
poner Reiniciar Visible como falso
poner Pausa Visible como cierto
llamar Sonido1 Reproducir

cuando Pausa Clic
ejecutar poner global Cronometro a falso
poner Reiniciar Visible como cierto
poner Pausa Visible como falso
llamar Sonido1 Reproducir
    
```

1.

```

cuando Reiniciar Clic
ejecutar poner global Cronometro a falso
poner TxtCronometro Texto como "00:00:00.00"
poner TxtCuentaAtras Texto como "00:00:00.00"
poner global Contador a 0

cuando Reloj1 Temporizador
ejecutar si tomar global Cronometro
entonces poner global Contador a tomar global Contador + 100
poner TxtCronometro Texto como llamar Reloj1 FormatoDeFecha
instante llamar Reloj1 HacerInstanteDesdeMilisegundos
pattern "HH:mm:ss"
    
```

2.

```

cuando Reloj1 Temporizador
ejecutar si tomar global Cronometro
entonces poner global Contador a tomar global Contador + 100
poner TxtCronometro Texto como llamar Reloj1 FormatoDeFecha
instante llamar Reloj1 HacerInstanteDesdeMilisegundos
pattern "HH:mm:ss"
poner TxtCuentaAtras Texto como llamar Reloj1 FormatoDeFecha
instante llamar Reloj1 HacerInstanteDesdeMilisegundos
pattern "HH:mm:ss"
    
```

3.

```

instanteDesdeMilisegundos
milisegundos llamar Reloj1 Duración + tomar global Contador - 3600000000
comienzo tomar global Milisegundos
fin tomar global Milisegundos

instanteDesdeMilisegundos
milisegundos llamar Reloj1 Duración - tomar global Contador - 3600000000
comienzo tomar global Milisegundos
fin tomar global Milisegundos
    
```

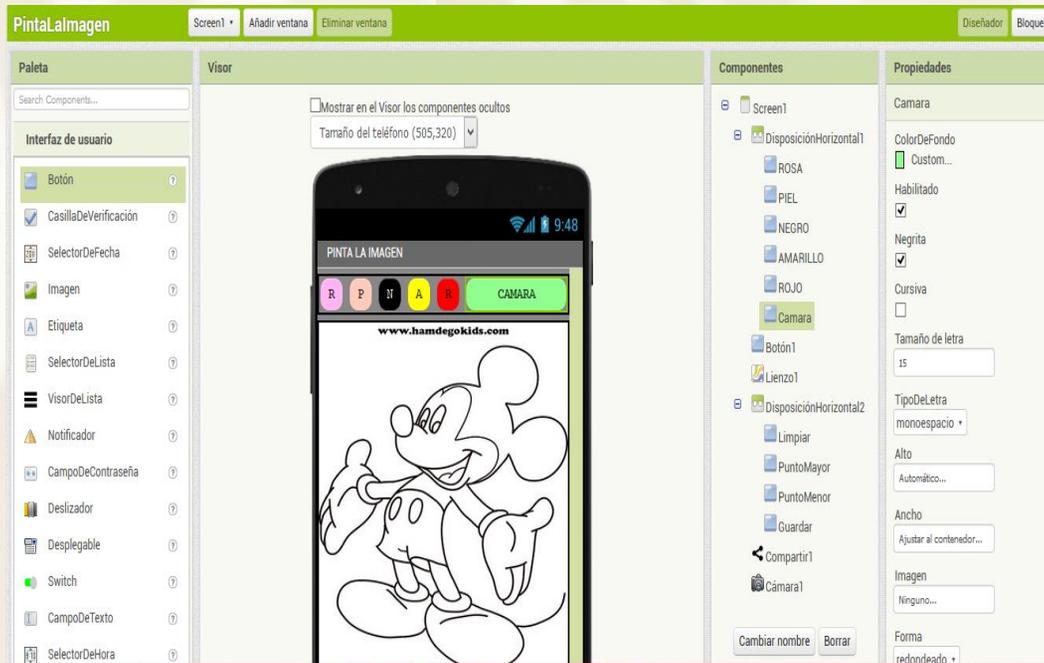
4.

Instrucciones que se siguen.

- Abriremos un proyecto nuevo, iniciando arrastraremos una disposición vertical a nuestra pantalla, de la misma forma arrastramos a nuestra disposición una etiqueta, a la que llamaremos título, de la misma forma dentro de esta disposición arrastraremos otra disposición, pero esta vez será horizontal, e la cual arrastraremos una etiqueta a la que llamaremos TxtCXronometro, a la disposición vertical le anexaremos una etiqueta (que llamaremos etiqueta 2) justo debajo de la disposición horizontal. De la misma forma anexaremos nuevamente otra disposición horizontal debajo de la etiqueta y de la misma manera arrastraremos una etiqueta la cual llamaremos TxtCuentaAtras.; por ultimo arrastraremos una última disposición horizontal, en la cual arrastraremos tres botones, uno para iniciar, otro para pausar y por ultimo uno para reiniciar. De la misma forma arrastraremos a la pantalla un reloj.
- Una vez terminemos de acomodar las propiedades de los botones, colores y etiquetas, así como que determinemos los títulos podemos empezar con la programación yéndonos a la parte de Bloques.
- Iniciaremos creando tres variables, una de Contador con 0; otra de milisegundos con cuadro de texto y por ultimo una de cronometro con bloque lógico de falso. Posteriormente pondremos un bloque de control que nos indique que cuando la pantalla inicialice se ejecutara la variable de milisegundos la cual llamara al reloj.
- Arrastraremos otro bloque de control que nos indica que al hacer clic e iniciar se ejecutara lo siguiente, al cual añadiremos la variable de cronometro a cierto, que de igual forma nos indicara que cuando inicie el botón de reiniciar visible como falso y pausa visible como cierto; es decir cuando se inicie el botón de pausa se mostrara y el de reiniciar se esconderá.
- Volveremos a arrastrar un bloque de control que nos indica que al hacer clic en el boto de pausase ejecutara lo siguiente, la variable de cronometro en falso, anexaremos el botón de reiniciar visible como cierto y botón de pausa visible como falso; es decir que cuando se haga cli en el botón de pausa, el botón de reiniciar se hará visible y el de pausa de esconderá.
- Volveremos a arrastrar un bloque de control, que nos dice que al hacer clic en reiniciar se ejecutara el cronometro y el contador atrás se volverán a poner en ceros para volver a iniciar.
- Por ultimo pondremos un bloque de control más de reloj, que nos dice que el temporizador ejecutara lo siguiente, el cual nos dice que el cronometro y la cuenta atrás, en el cual pondremos el formato de tiempo que se manejara con un cuadro de texto, al cual le anexaremos un bloque matemático en el que indicaremos si se restara tiempo o sumara, en el caso del cronometro se sumara y en la cuenta atrás se restara; en ambas le restara a la cuenta actual 36000000, para que si el cronometro inicie en 00.

ANEXO 10

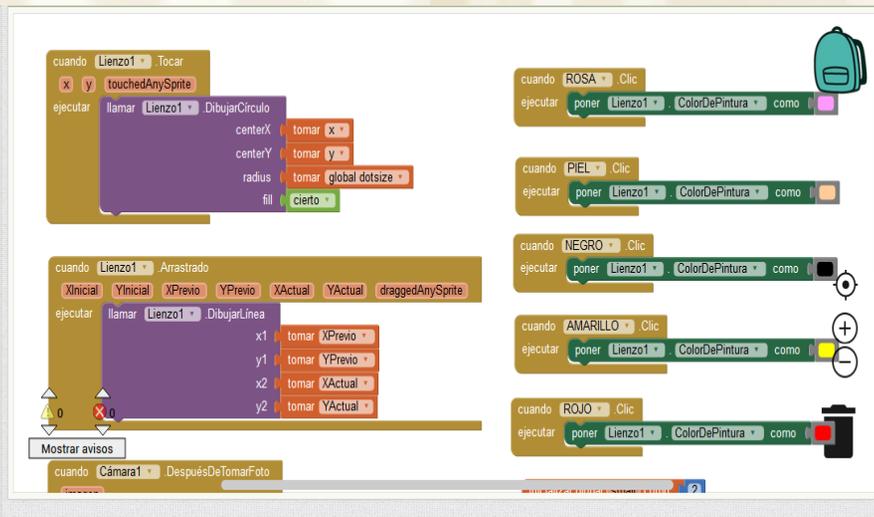
Practica 3: IMAGEN PARA COLOREAR



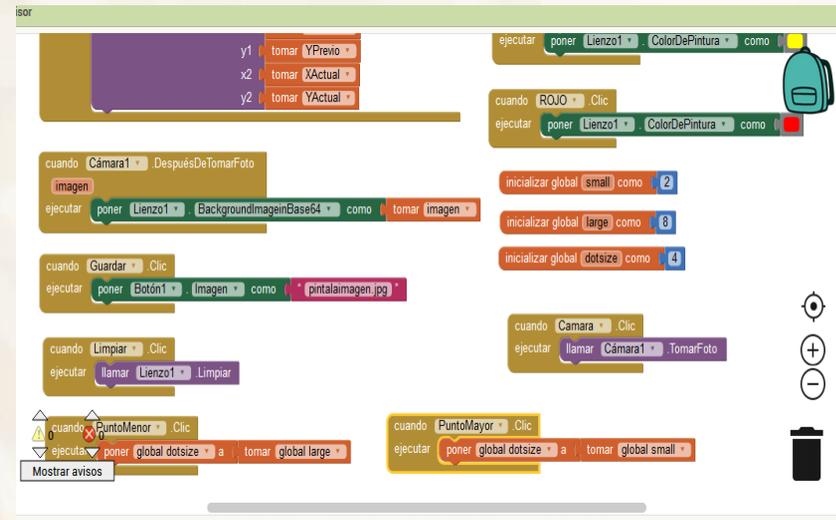
Diseño.

Seleccionas el color y con el dedo te posicionas donde quieres que aparezca el color para la imagen.

Bloques requeridos para el diseño.



1.



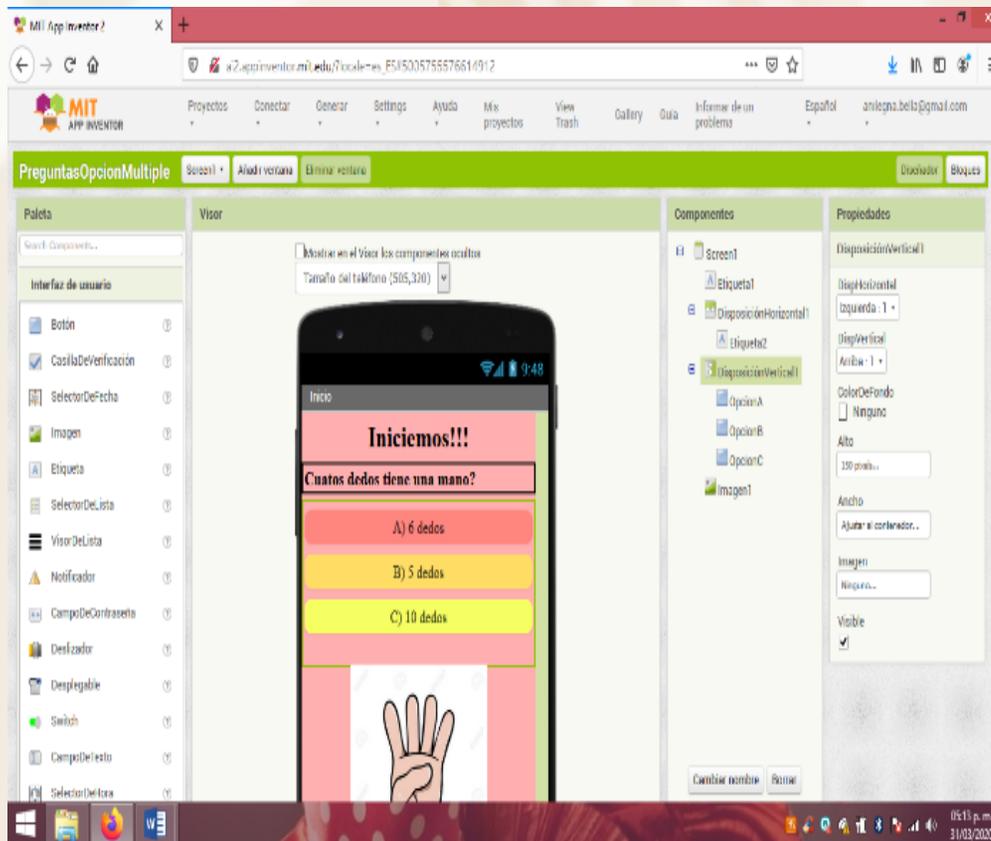
2.

Instrucciones que se siguen.

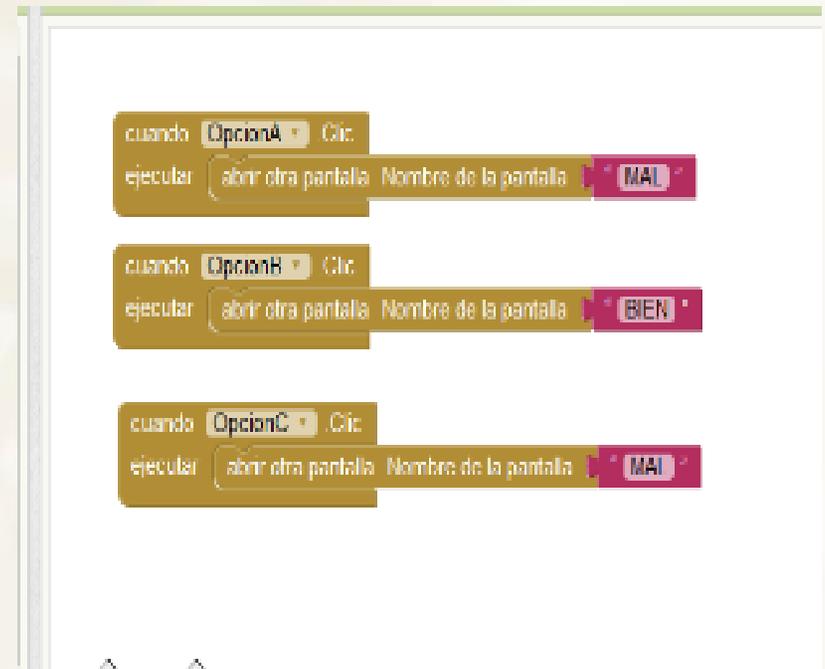
- Iniciamos arrastrando a nuestra pantalla una disposición horizontal, a la que arrastraremos los botones de los colores que usaremos y uno extra para la cámara. Arrastramos debajo de la disposición un lienzo, en el cual adjuntaremos una imagen, la cual queremos colorear. De igual manera añadimos una disposición horizontal de bajo del lienzo, a la cual le añadiremos los botones de borrar, aumentar punto y disminuir punto, así como el de guardar. Editaremos el color, tiempo de letra y texto de cada uno de los botones, así como su tamaño.
- Empezaremos con la parte de programación, para lo cual nos dirigiremos a la parte de bloques. Iniciaremos arrastrando del lienzo un bloque que nos indica que al hacer clic llamara a dibujar. Posteriormente tomaremos otro bloque de lienzo el cual nos arrastrara otras variables.
- Iniciaremos con nuestros botones de color. Traeremos un bloque de nuestro botón ros, el cual nos dice que, al hacer clic, el lienzo pintara y traeremos un bloque de color, al cual le pondremos el color indicado, en este caso el rosa. Este proceso lo repetiremos con nuestros demás botones de colores.
- Ahora crearemos nuestras variables para nuestros botones de más y menos punta. Traeremos tres bloques de variable a las cuales le pondremos small y le pondremos bloque matemático de 2, a la otra large y numero 8 y por último dotsize con número 4. Ahora empezaremos con el botón de cámara, con un bloque que nos llamara a lienzo y variable de imagen; de igual manera llamaremos otro bloque que nos llame que al hacer clic en cámara nos llamara a la cámara. Ahora el botón guardar, el cual nos indica que al hacer clic nos guardara la imagen, con un cuadro de texto pondremos el nombre de la imagen.
- Para el botón de limpiar, utilizaremos un bloque que al hacer clic nos llamara la limpieza del lienzo. Por último, programaremos nuestro botones de punta más y punta menos, para el cual utilizaremos un bloque que nos indica que al hacer clic en punta menor nos llamara a la variable de dotsize y tomara la variable de large, duplicamos todo el bloque y cambiamos punto menor por mayor y large por small.

ANEXO 11

Practica 4: CREAR UNA APP CON PREGUNTAS DE OPCION MULTIPLE



Diseño
Instrucciones que se siguen.



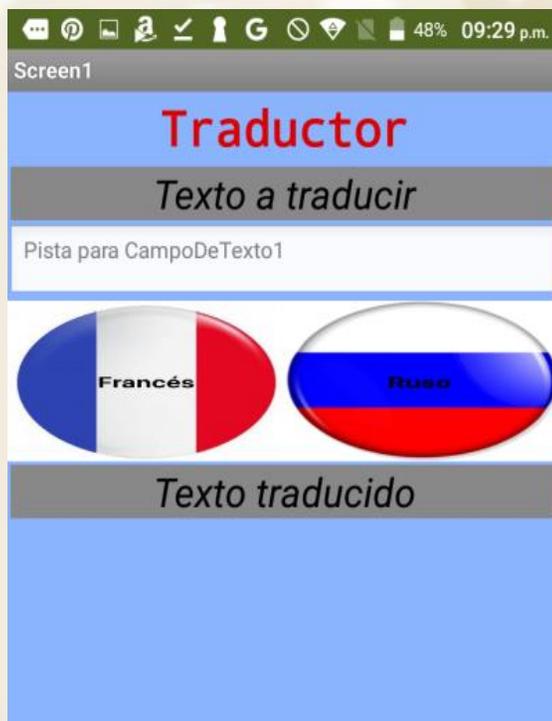
Bloques requeridos

- Iniciaremos abriendo un proyecto nuevo, a nuestra pantalla arrastraremos una etiqueta para nuestro título, posteriormente arrastramos una disposición horizontal a la cual introduciremos una etiqueta en la cual introduciremos la pregunta que deseamos realizar. Nuevamente arrastraremos una disposición, en esta ocasión vertical, en la cual introduciremos tres botones, para nuestras tres posibles respuestas, y para darle vida, arrastraremos debajo de esta disposición un elemento de imagen para adjuntar una relacionada con el tema. Una vez hallamos editado las propiedades de los botones, etiquetas y de la imagen a nuestro gusto, seguiremos con nuestra siguiente pantalla, esta pantalla la aremos nuevamente para nuestra pregunta dos, con la diferencia del título, pregunta, las respuestas y la imagen. De igual forma a esta pantalla arrastraremos un elemento de sonido, al cual adjuntaremos el que deseemos.
- Aremos clic en añadir ventana y colocaremos el nombre, esta ventana será la que nos aparezca cuando nuestra respuesta seleccionada sea correcta, para lo cual introduciremos un elemento de imagen a la cual adjuntaremos la imagen de correcto; de bajo de esta clocaremos un botón el cual nos llevará a l pregunta siguiente. Una vez terminemos de acomodar las propiedades seguiremos con la siguiente ventana.
- Abriremos una ventana nueva, la cual usaremos para que nos indique que la opción es incorrecta. Para esta de igual forma arrastraremos un elemento de imagen para adjuntarle una imagen que nos indique que es incorrecta nuestra respuesta; así mismo arrastraremos un botón debajo de esta para que con el no podamos regresar a la pregunta para así tratar de contestarla nuevamente. Acomodaremos las propiedades y finalizaremos con esta pantalla. Aremos dos pantallas iguales que nos indique que la respuesta es incorrecta ya que tendremos dos preguntas.
- Ahora empezaremos con el diseño de nuestra pantalla de correcto, pero que de igual forma nos indique que hemos finalizado la encuesta. Para esta pondremos una etiqueta, ¡¡¡para poner el mensaje "TERMINASTE!!!", posteriormente arrastraremos un elemento de imagen, ahí adjuntaremos la imagen que ya usamos anteriormente para correcto; de igual forma colocaremos un botón debajo de la imagen, pero este nos servirá para regresarnos a la primera pregunta.
- Ahora iniciaremos con la parte de programar nuestra app, para lo cual nos iremos a la parte de bloques. Donde iniciaremos con nuestra primera ventana, a la cual arrastraremos un bloque de cada botón que nos indique que al hacer clic en el botón, con otro bloque nos mande a la ventana de, y con un bloque de texto
- escribimos el nombre de la ventana a la cual deseamos que nos mande. Este proceso se repetirá con los tres botones, con la diferencia del botón y del texto, la respuesta correcta nos llevará a la ventana de BIEN y las incorrectas a la de MAL. Este proceso se repetirá con la ventana de la pregunta dos, con la defenecía de que los botones te llevaran a la otra ventana de "mal" y la respuesta correcta te llevara a la ventana de correcto finalizado.
- Ahora nos iremos a la ventana de MAL, para la cual arrastraremos un bloque del botón, que nos indique que al hacer clic, con otro bloque nos lleve a otra ventana, la cual indicaremos con un bloque de texto, al escribir el nombre de la ventana, en el caso de esta ventana nos debe regresar a la ventana anterior. Esto lo repetiremos con ambas ventanas de mal.
- Para la ventana de bien, la que nos indica que la respuesta es correcta, aremos lo mismo, del botón arrastraremos un bloque que nos indique que al hacer clic, con otro bloque nos lleve a la pregunta siguiente.

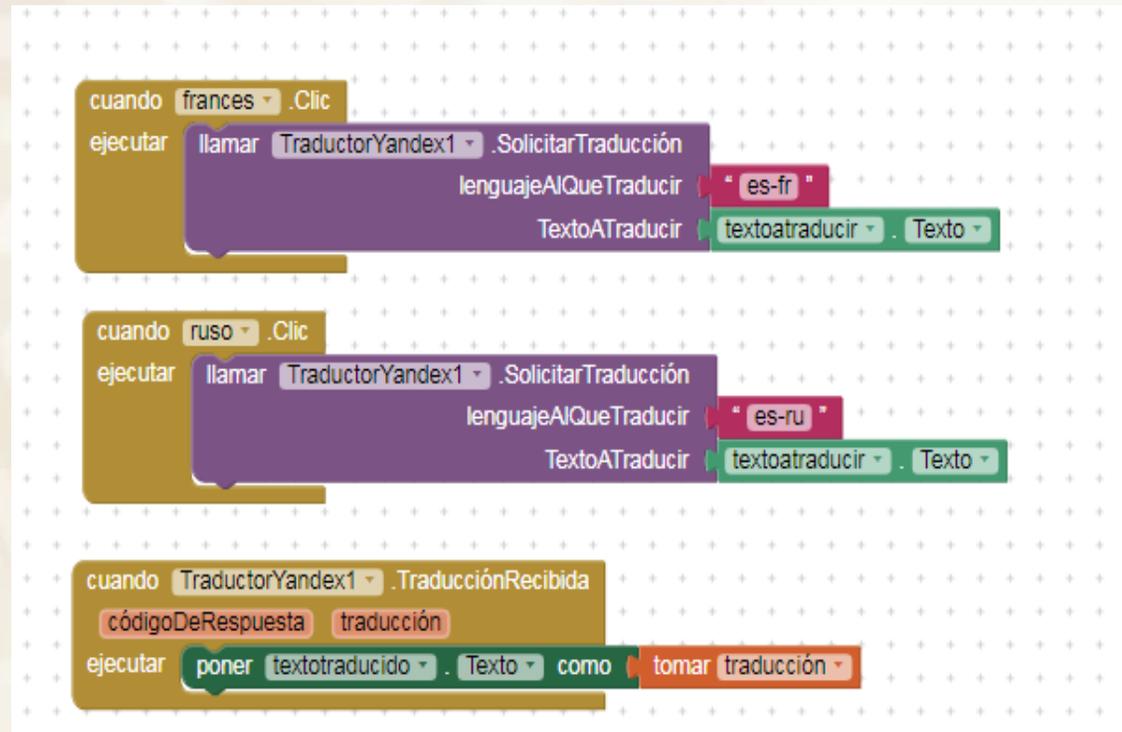
- En el caso de la última pantalla de bien, esta nos indica que es correcta y que hemos finalizado las preguntas por lo cual el botón nos llevara nuevamente a la primera pregunta. Y a esta pantalla con un bloque de sonido que al hacer clic, con otro bloque inicie el sonido que anteriormente adjuntamos.

ANEXO 12

Practica 5: CREAR UNA APLICACION DE UN TRADUCTOR



Diseño



Bloques a utilizar para su realización del diseño.

Instrucciones para su desarrollo.

Esta práctica es acerca de la realización de un traductor con mínimo 2 idiomas a traducir, para hacerlo se necesitaron varias etiquetas para poner el título, el texto a traducir y el texto traducido, una disposición horizontal para colocar los botones de los 2 idiomas, un traductor-yandex y descargar las imágenes del par de países seleccionados por sus idiomas. Finalmente, en diseñador se fueron colocando los nombres de las etiquetas, subiendo las imágenes de las banderas y por último cambiando colores, fuentes, etc.

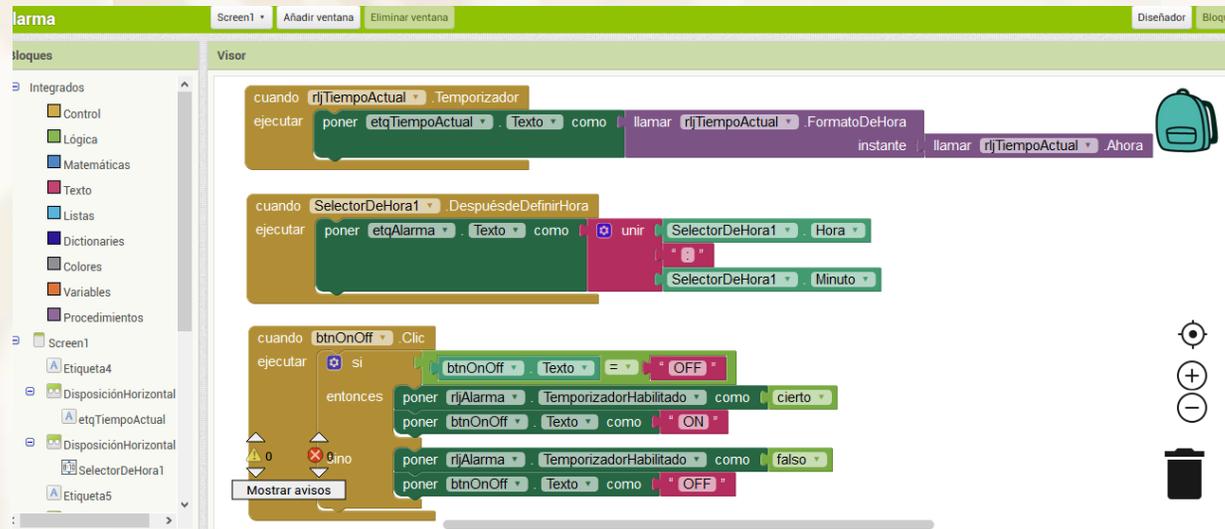
Después en bloques lo primero fue colocar el bloque para que cuando se le haga a clic a francés el texto a traducir se logre traducir al igual que con el ruso y para terminar se colocó el bloque del traductor para que se pueda traducir el texto.

ANEXO 13

Proyecto 1 CREAR UNA APLICACIÓN DE UNA ALARMA

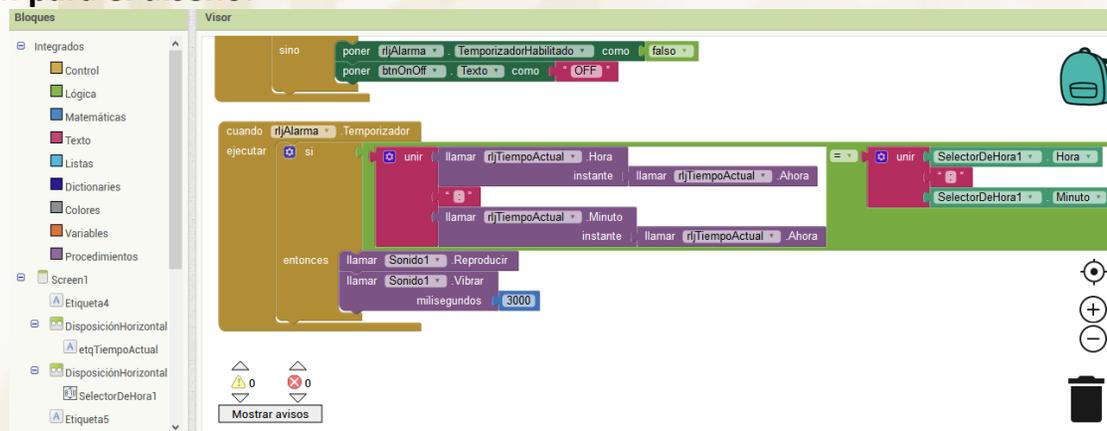


Diseño.



1.

Bloques que se requieren para el diseño.



2.

Instrucciones que se siguen.

- Abriremos un proyecto nuevo, a nuestra pantalla arrastraremos una etiqueta en la cual pondremos el título; posteriormente arrastraremos una disposición horizontal a la cual añadiremos una etiqueta la cual nos dirá el tiempo actual; nuevamente arrastraremos una disposición horizontal a la cual arrastraremos un selector de hora, es en el cual colocaremos la hora en la que deseamos que suene nuestra alarma; arrastramos una etiqueta para indicar la hora; nuevamente arrastramos una disposición horizontal a la cual arrastraremos una etiqueta para nuestra hora de la alarma; una vez más ponemos una disposición horizontal a la que introduciremos

una etiqueta y un botón, el cual será nuestro botón de encendido y apagado y la etiqueta nos indicara el modo. Por ultimo arrastraremos una disposición vertical para poner un elemento de imagen el cual adjuntaremos a nuestro gusto. Algo más que necesitaremos será, dos relojes, uno para la hora actual y otro para la alarma; así como un sonido para reproducir el sonido de nuestro despertado. Una vez modifiquemos las propiedades de nuestro botón, etiquetas e imagen a nuestro gusto, iniciaremos con la parte de programar nuestra app.

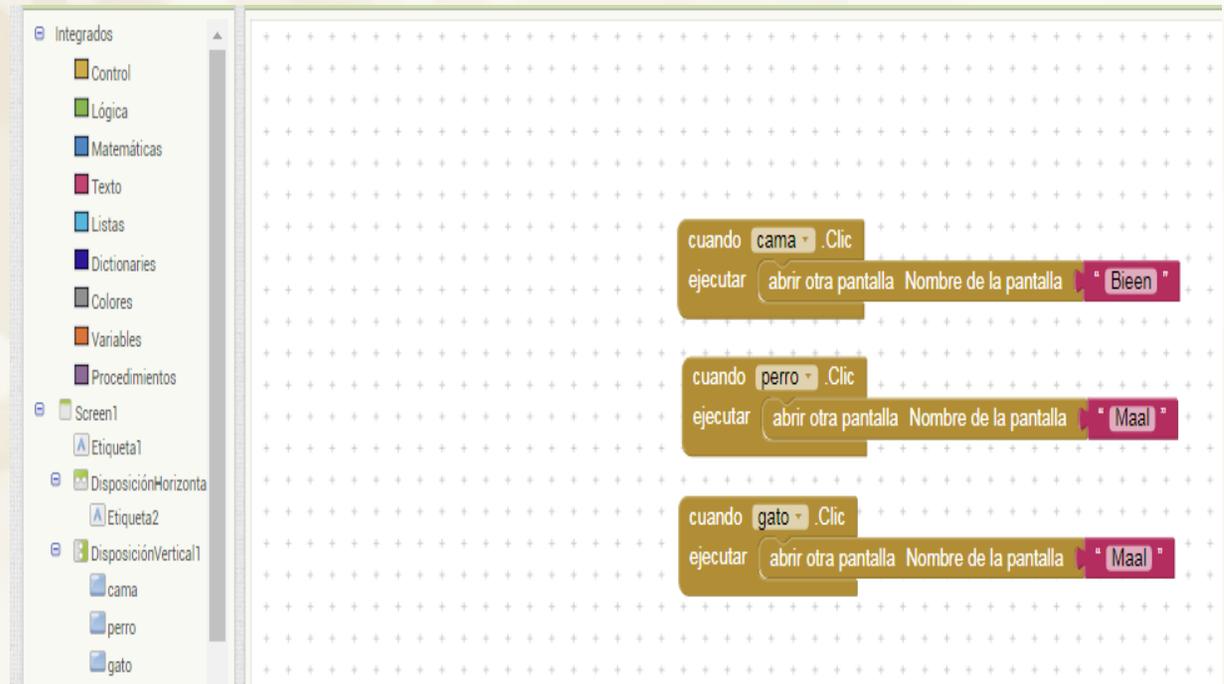
- Para esto nos iremos a la parte de bloques. Iniciaremos con un bloque del tiempo actual pone al tiempo en texto el cual llamara a la hora actual en horas. Con un bloque que nos indique que del selector de hora pondremos la alarma con texto, con un bloque de unir en el cual pondremos selector de hora en horas y un selectos de hora en minutos esto dividido por un bloque de texto en el que colocaremos dos puntos.
- Posteriormente programaremos el botón de encendido y apagado con un bloque al hacer clic nos marcara con un bloque lógico comparativo. Pondremos que cuando el temporizador este habilitado cierto con bloque de teto en ON, es decir que cuando se inicie el temporizador nos apareceré el botón de apagado. Ahí mismo anexaremos un bloque que nos dice que la alarma esta inhabilitada falso, texto OFF nos parece el botón de encendido.
- Posteriormente con un bloque del arma, temporizar nos ejecutara si, con un bloque comparativo, bloque lógico, él nos une el tiempo actual, comparando con la hora y minutos establecidos. Entonces un bloque de sonido nos llamara el sonido, de igual forma de sonido llamaremos un bloque que vibre y con un bloque matemático estableceremos el tiempo que deseamos que dure la vibración, el tiempo es en milisegundos.

ANEXO 14

Proyecto 2 CREAR UNA APLICACIÓN DE UN QUIS



Diseño



Bloques utilizados para hacer el diseño propuesto.

INSTRUCCIONES DE COMO DESARROLLARLA

Esta práctica es acerca de la realización de una aplicación donde se hace una pregunta y se presiona 1 de las 3 alternativas como respuesta, para esto se utilizó una etiqueta para el título, una disposición horizontal para colocar la etiqueta que tiene la pregunta, una disposición vertical para colocar los 3 botones que serán las respuestas para elegir, así como descargar imágenes alusivas a la pregunta que será una para el fondo, para cuando te equivoques y una para cuando aciertes. En diseñador se acomodó lo mencionado anteriormente y se modificó para que se viera mejor cambiando los colores al igual que colocando una imagen, luego en el screen 2 se colocó la imagen de mal así como un botón para volver e igual con el botón de bien en el screen 3.

Después en bloques se configuro para que cuando respondieras te llevara a saber si estaba bien o mal, esto que cuando le des clic a camaleón te lleve a bien y cuando hagas clic en perro o gato te lleve a mal.