

Entendiendo la Arquitectura de una Aplicación

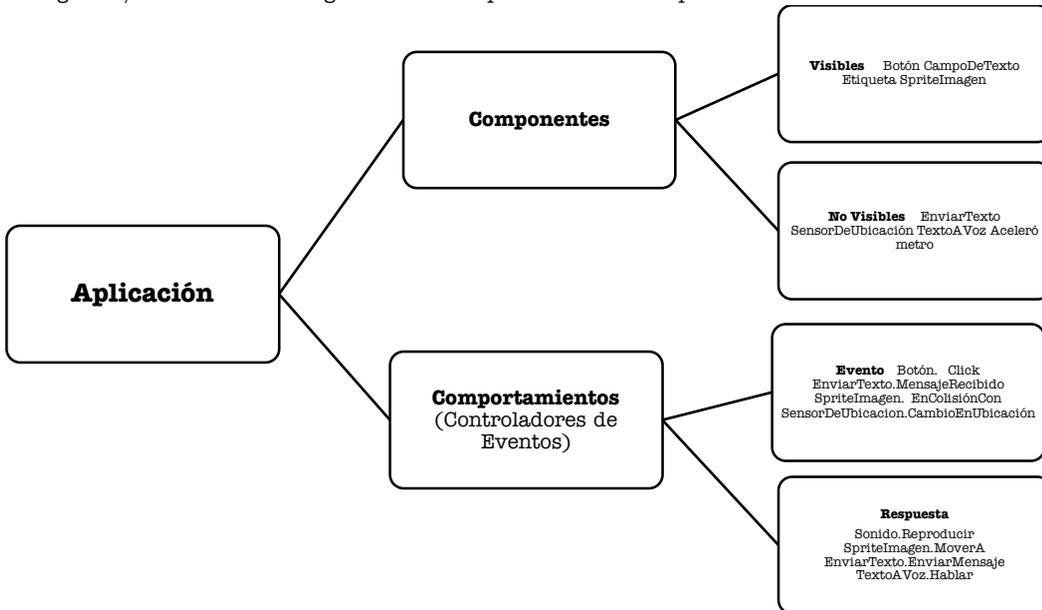
Muchas personas pueden decir que es lo que es una aplicación, desde la perspectiva del usuario, pero entender que es una aplicación desde la perspectiva de un programador es más complicado. Las aplicaciones tienen una estructura interna, lo que se conoce como la arquitectura de la aplicación, que se debe entender para poder crear aplicaciones de manera efectiva.

Una manera de describir el interior de una aplicación es separarla en dos partes: sus componentes, y sus comportamientos. En general, estos conceptos corresponden a las dos

ventanas principales de AppInventor. Por un lado, se usa el Diseñador de Componentes para especificar los componentes (u objetos) de la aplicación, y por otro lado se usa el Editor de Bloques para programar como la aplicación responde al usuario y a otros eventos externos.

O sea, el Editor de Bloques se usa para programar el comportamiento de la aplicación.

La Figura 1, muestra una vista general de la arquitectura de una aplicación.



Componentes

Existen dos tipos principales de componentes en una aplicación: visibles y no-visibles.

Los **componentes visibles** de una aplicación son aquellos que se pueden ver cuando la aplicación se ejecuta, por ejemplo, los botones, cajas de texto y etiquetas. Al conjunto de componentes visibles de una aplicación también se le conoce como la interfaz de usuario.

Los **componentes no-visibles** son aquellos que no se pueden ver, y que por lo tanto no son parte de la interfaz de usuario. Su propósito es proveer acceso a las funcionalidades preexistentes de un dispositivo.

Por ejemplo, el componente `EnviarTexto` envía y procesa los mensajes de texto (SMS), el componente `SensorDeUbicación` determina la ubicación del dispositivo, y el componente `TextoAVoz` habla un mensaje escrito como texto. Los componentes no-visibles representan la tecnología del dispositivo que está a disposición del programador.

Tanto los componentes visibles como no-visibles se definen por un conjunto de **propiedades**.

Las **propiedades** son espacios de memoria para almacenar información sobre el componente.

Los componentes visibles, tales como botones o etiquetas, tienen propiedades como su anchura, altura y alineamiento, los que en conjunto definen cómo luce el componente. Las propiedades de un componente son como celdas de una hoja de cálculo.

Una Aplicación es Como una Receta de Cocina

Tradicionalmente se ha comparado al software (programas, aplicaciones) con una receta de cocina. Como en una receta, una aplicación tradicional sigue una secuencia lineal de instrucciones, como las de la Figura 2, que el computador debiera ejecutar.

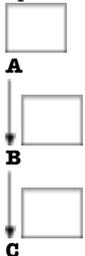


Figura 2: Una aplicación tradicional sigue una serie de pasos secuenciales, como una receta de cocina.

El programador los modifica en el Diseñador de Componentes para definir la apariencia inicial del componente. También es posible utilizar bloques para cambiar estos valores durante la ejecución de la aplicación.

Comportamiento.

Los componentes de una aplicación son generalmente sencillos de comprender, por ejemplo, un campo de texto se usa para ingresar información o un botón se usa para ser presionado.

En cambio, el comportamiento de una aplicación es conceptualmente difícil y a menudo complejo.

El comportamiento define como la aplicación debiera responder a eventos, tanto eventos iniciados por el usuario (por ejemplo, se presiona un botón) como eventos externos (por ejemplo, se recibió un mensaje de texto).

La dificultad de especificar ese comportamiento interactivo es por qué la programación es un desafío.

Afortunadamente, AppInventor provee un lenguaje de bloques para especificar estos comportamientos. Los bloques hacen que programar el comportamiento sea similar a juntar las piezas de un puzzle, en contraste a recordar y escribir código como en lenguajes de programación tradicionales. Además, AppInventor está diseñado para especificar comportamientos en respuesta a eventos de una manera sencilla y directa.

Si consideramos como ejemplo una aplicación de un cajero automático, una primera operación (A) sería iniciar una transacción bancaria, luego (B) especificar el monto que se desea retirar, y finalmente (C) modificar la cuenta del cliente, entregar el dinero y luego imprimir el saldo por pantalla.

Una Aplicación Como un Conjunto de Controladores de Eventos

La visión de una aplicación como una receta de cocina calza bien con las aplicaciones o programas que se hacían en los inicios de la computación, pero no es una gran idea para la programación de dispositivos móviles, ni en la Web, ni en la mayoría de las aplicaciones y plataformas actuales en computación.

La mayor parte del software moderno no realiza un puñado de instrucciones en un orden predeterminado. Lo que se hace es que el software reaccione ante distintos eventos la mayoría iniciados por la interacción entre el usuario y la aplicación (por ejemplo, abrir un video en YouTube).

En el caso de las aplicaciones móviles tenemos diversos eventos gatillados por el usuario.

Por ejemplo, al presionar el botón “Enviar”, la aplicación responde enviando un mensaje de texto.

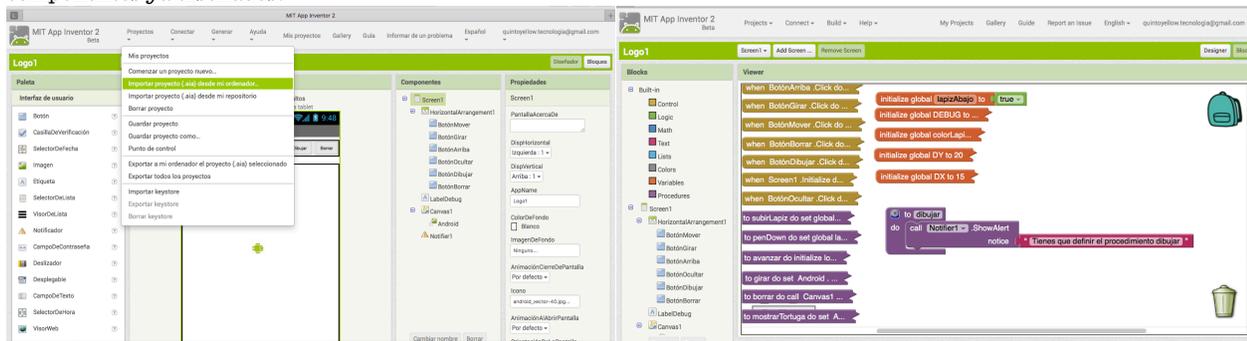
El deslizar el dedo por la pantalla táctil también es otro evento.

La aplicación podría responder dibujando una línea entre el punto donde se comenzó a deslizar el dedo y el punto donde se levantó.

Actividad :

Ingresa al siguiente link bit.ly/clubdeapps y descarga del día 5 el logo 1

Al descargar el archivo, debes abrir **importar proyecto aia desde mi ordenador** y abrir el proyecto. Donde veras una serie de componentes ya diseñados.



La idea es poder **diseñar algoritmos** (secuencia de instrucciones) para que nuestro programa logo dibuje lo que diseñemos previamente.



Para dibujar un cuadrado de 10 por 10 pixeles:

- Avanzar
- Girar
- Avanzar
- Girar
- Avanzar
- Girar
- Avanzar
- Girar

Diseñar un algoritmo y definir procedimiento.

