

# **Guion para el taller de adolescentes**

1. Introducción a la programación (5-10 min) (D)
  - a. Qué es (preguntar si sabrían dar una definición y luego aportar nosotros una, probablemente de Wikipedia)
  - b. Lenguajes (preguntar si les suenan algunos, y luego listar los que se nos vengan a la cabeza)
  - c. Ejemplos de programas (meternos a GitHub y poner un par de ejemplos de código, preferiblemente en diferentes lenguajes)
2. Introducción a App Inventor (5-10 min) (D)
  - a. Interfaz
  - b. Widgets disponibles para colocar en la vista de diseño
  - c. Vista rápida de las cajas de código, sin meterse a explicar qué hacen
3. Etiquetas, campos de texto y botones (R)
  - a. Enseñar estos tres widgets y mostrar por encima las distintas cosas que se pueden cambiar en la parte derecha
  - b. Ejemplo: Campo de texto con botón que al pulsarlo muestra lo que hayas escrito en una etiqueta (5 min)
  - c. Ejercicio: Dos campos de texto, donde se van a introducir números, con un botón que al pulsarlo muestra el resultado de sumar ambos números. Este ejercicio se irá ampliando en los siguientes apartados (10 min)
4. Variables (R)
  - a. Contar la importancia de usar variables para evitar redundancias en el código de un programa real (5 minutos o menos)
  - b. Ejemplo: Sobre el ejemplo anterior, guardar en una variable lo que se escribió en el campo de texto en lugar de usar el texto directamente. (5 min o menos)
  - c. Ejercicio: Guardar en variables los números introducidos y el resultado. Comentar que en este caso es bastante innecesario esto ya que es lo mismo arrastrar una caja de variable que la de "Textbox.Text" (5 minutos o menos)
5. Notifiers (D)
  - a. Ejemplo: Campo de texto con un botón que al pulsarlo muestra una notificación con lo que se ha introducido (5 min o menos)
  - b. Ejercicio: Mostrar en una notificación el resultado de la suma (5 min o menos)
6. Condicionales (D)
  - a. Explicar el funcionamiento de las sentencias condicionales: if, else, else if (5 min)
  - b. Ejemplo: Campo de texto con un botón que al pulsarlo muestra una notificación si el texto tiene alguna particularidad (lo que se nos ocurra:

que tenga una longitud determinada, que sea igual a algo...).

Alternativamente, dos campos de texto con un botón que al pulsarlo notifique si ambos textos son iguales (5 min)

- c. Ejercicio: Eliminar la notificación del ejercicio anterior (la del resultado). Mostrar qué pasa actualmente cuando se intenta sumar y algún campo está en blanco. Añadir comprobaciones a ambos campos para mostrar una notificación si alguno está vacío. (5-10 min)

## 7. Listviews (R)

- a. Explicar lo que es una listview (una lista, vamos), bastante típico en Android por lo que tengo entendido. (5 min o menos)
- b. Ejemplo: Campo de texto con un botón que cada vez que se pulsa introduce en una listview lo que se ha escrito (5 min)
- c. Ejercicio: En lugar de mostrar el resultado en una etiqueta y que se borre cada vez que se haga otra suma, guardar los resultados en una listview (alternativamente, que guarde algún texto como “El resultado de  $x + y$  es  $z$ ”). Añadir también un botón de borrar que vacíe la listview (10 min)

## 8. Cambio de pantalla (D)

- a. Ejemplo: Dos pantallas con un botón cada una, que lo único que hacen es cambiar entre una pantalla y otra (5 min o menos)
- b. Ejercicio: Añadir un menú a la aplicación con dos botones: “Sumas” y “Factoriales”, y dos pantallas. Cada botón llevará a una pantalla, y el botón correspondiente a la pantalla actual (por ejemplo, “Sumas” si estamos en la pantalla de sumas) deberá estar desactivado (5 min)

## 9. Bucles (R)

- a. Explicar los bucles: for, while. Comentar que no se suelen recomendar pero que no hay más remedio que usarlos de vez en cuando (5 min o menos)
- b. Ejemplo: Campo de texto, donde se introduce un número, con un botón que al pulsarlo muestre en una etiqueta los números desde el 1 hasta el que se ha introducido (5 min)
- c. Ejercicio: Campo de texto, donde se introduce un número, con un botón que al pulsarlo meta en una listview si el número introducido es primo o no. Añadir también un botón para vaciar la listview. Hablar luego sobre la poca eficiencia del algoritmo, y mejorarlo haciendo que el bucle llegue solo hasta el suelo de la raíz cuadrada del número, y que el bucle tenga un “break” en cuando encuentre algún divisor (10 min o más)
- d. (Extra, si da tiempo) Preguntar si conocen los números factoriales. Como es 99.9% seguro que no, se les explica qué son, cómo se calculan, y su aplicación en el cálculo de probabilidades (típico ejemplo

de una bolsa con bolas numeradas, y calcular la probabilidad de que se saquen las bolas en un orden determinado) (5 min)

- e. (Extra, si da tiempo) Ejercicio: Campo de texto con un botón que al pulsarlo te muestre en una etiqueta el factorial de número que se haya introducido. Sería conveniente escribir primero un pseudocódigo del algoritmo, desarrollándolo poco a poco y en conjunto con los alumnos, porque no creo que solo con la explicación anterior se les vaya a ocurrir un algoritmo para calcular factoriales. (10 min o más)

#### 10. Mostrar aplicaciones que hayamos hecho (R)

- a. Enseñamos lo que hayamos hecho nosotros en App Inventor con cierta complejidad. Mostramos las cajas para que vean que lo que hemos dado en el taller es algo muy básico (5-10 min)

#### 11. Tiempo restante

- a. En el tiempo que quede, proponemos tres cosas: ampliar la aplicación que hemos hecho en los ejercicios con algo que se les ocurra a ellos, hacer algo de cero que ellos propongan, o dejarlos libres para que experimenten con lo que quieran (que estará bien si de verdad experimentan y no se ponen a hacer otras cosas).
- b. Podríamos explicar cómo exportar la aplicación a apk por si a alguno le hiciera ilusión llevársela.