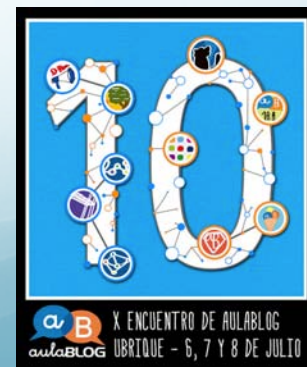


Taller de **AppInventor**

Iván Ruiz Rube



Materiales taller

- ✓ Esta presentación
- ✓ Tutorial paso a paso
- ✓ “Código fuente” del proyecto
- ✓ Aplicación (fichero APK)
- ✓ Recursos (imágenes, sonidos, certificado) asociados al proyecto

Agenda

- Contexto
- Programación en Android
- AppInventor
- Demo:
 - *Retocar* y compartir fotos
 - Participar en cuestionarios

Tecnología everywhere



Aprendizaje de la programación

- Permite entender mejor la realidad que nos rodea y nos prepara para aportar valor a la sociedad.
- Conocer la programación no para que seamos programadores en el futuro, sino porque vamos a pasar la vida rodeados de objetos programables.
- El objetivo es formar a creadores y no sólo a meros consumidores.
- La programación potencia la creatividad, el pensamiento lógico y nos aporta habilidades para la resolución de problemas.

Clases aplicadas

- Clases aplicadas en centros públicos y concertados en la provincia de Cádiz durante 2015
 - 4 charlas en Cádiz
 - 2 charlas en Chiclana de la Frontera
 - 1 charlas en El Puerto de Santa María
 - 2 charlas en Jerez de la frontera
 - 1 charlas en San Fernando
- Perfil del alumnado
 - Alumnos de Bachillerato
 - Ciclos formativos
 - Alumnos de 4º de ESO



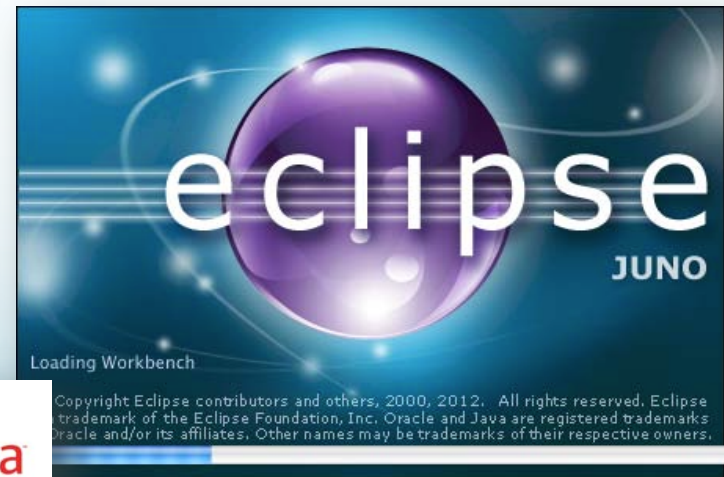
Android



- Sistema operativo para móviles más extendido en la actualidad
- Basado en el sistema Linux y desarrollado por Google
- Actualmente en su versión 5.0 (Lollipop) aunque la versión con mayor cuota de mercado es la 4.4 (KitKat)
- Gran comunidad desarrolladores: +1M apps en Google Play

Programación en Android

- Los programas están escritos en el lenguaje de programación Java.
- Requiere conocimientos de Ingeniería del Software: análisis, algoritmia, usabilidad, pruebas, etc.
- Existen entornos especializados de desarrollo:



Programación en Android

The screenshot displays the Android Studio IDE with the `MainActivity.java` file open. The code defines a `MainActivity` class that extends `Activity` and implements `onCreate`, `onOptionsItemSelected`, and `onCreateOptionsMenu` methods. The `onCreate` method initializes the UI, sets up button click listeners for `submitButton` and `streamButton`, and registers a device. The `onOptionsItemSelected` method handles menu item selections, and the `onCreateOptionsMenu` method inflates the menu.

```
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import co.infinum.streamoplayer.utils.JSONParser;

public class MainActivity extends Activity {

    private static String NEW_DEVICE_URL_EXTENSION = "/api/new_device";
    private static final String TAG_RESULT = "result";
    private static final String TAG_MESSAGE = "message";
    private static final String PARAMETER_NAME = "name";
    private static final int RESULT_SETTINGS = 1;
    private static final int RESULT_STREAM = 2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button submitButton = (Button) findViewById(R.id.submitButton);
        Button streamButton = (Button) findViewById(R.id.streamButton);
        showUserSettings();

        submitButton.setOnClickListener((v) -> {
            registerDevice();
            // new RegisterDevice().execute(newDeviceURL(), null, null);
        });

        streamButton.setOnClickListener((v) -> {
            Intent streamIntent = new Intent(getBaseContext(), StreamActivity.class);
            startActivityForResult(streamIntent, RESULT_STREAM);
        });

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.main, menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            Intent intent;
            switch (item.getItemId()) {

                case R.id.menu_settings:
                    intent = new Intent(this, UserSettingsActivity.class);
                    startActivityForResult(intent, RESULT_SETTINGS);
            }
        }
    }
}
```

The right pane shows a preview of the app, which is a stopwatch titled "Ultimate Stopwatch". The stopwatch display shows "00:00:00.000". Below the display are three buttons: "START", "RESET", and "LAP". The bottom of the screen shows the Android navigation bar.

At the bottom of the IDE, there is a status bar with the following information:

- TODO
- 6: Android
- Unregistered Git root detected: The directory /Users/zeljokopasac/projects/android/... (moments ago)
- Loading spellchecker dictionaries...
- 19:30
- LF + UTF-8
- 117M of 711M

App Inventor

- Aplicación web desde la que podemos crear apps con apenas conocimientos de programación.
- Diseño visual de la interfaz de usuario mediante una serie de herramientas
- Programación como un puzle



<http://appinventor.mit.edu/>

Herramientas

User Interface

	Button	
	CheckBox	
	DatePicker	
	Image	
	Label	
	ListPicker	
	ListView	
	Notifier	
	PasswordTextBox	
	Slider	
	Spinner	
	TextBox	
	TimePicker	
	WebView	

Layout

	TableArrangement	
	VerticalArrangement	
	HorizontalArrangement	

Drawing and Animation

	Canvas	
	ImageSprite	
	Ball	

Media

	Camcorder	
	Camera	
	ImagePicker	
	Player	
	Sound	
	SoundRecorder	
	SpeechRecognizer	
	TextToSpeech	
	VideoPlayer	
	YandexTranslate	

Herramientas

Social

	ContactPicker	
	EmailPicker	
	PhoneCall	
	PhoneNumberPicker	
	Sharing	
	Texting	
	Twitter	

Storage

	FusionTablesControl	
	File	
	TinyWebDB	
	TinyDB	

Sensors

	Clock	
	BarcodeScanner	
	LocationSensor	
	OrientationSensor	
	NearField	
	AccelerometerSensor	
	ProximitySensor	

Connectivity

	BluetoothClient	
	BluetoothServer	
	Web	
	ActivityStarter	

LEGO® MINDSTORMS®

	NxtDirectCommands	
	NxtColorSensor	
	NxtLightSensor	
	NxtSoundSensor	
	NxtUltrasonicSensor	
	NxtTouchSensor	
	NxtDrive	

App Inventor: Diseñador

The screenshot displays the MIT App Inventor 2 Beta web interface. The top navigation bar includes the MIT App Inventor 2 Beta logo, a menu with 'Projects', 'Connect', 'Build', and 'Help', and user information for 'My Projects', 'Guide', 'Report an Issue', and 'ruizrube@gmail.com'. The main workspace is divided into four panels: 'Palette', 'Viewer', 'Components', and 'Properties'.

- Palette:** Contains categories like 'User Interface', 'Layout', 'Media', 'Drawing and Animation', 'Sensors', 'Social', 'Storage', and 'Connectivity'. Under 'User Interface', various components are listed, including 'Camcorder', 'Camera', 'ImagePicker', 'Player', 'Sound', 'SoundRecorder', 'SpeechRecognizer', 'TextToSpeech', 'VideoPlayer', and 'YandexTranslate'.
- Viewer:** Shows a preview of the app's interface. It includes a status bar at the top with a 'Screen1' dropdown, 'Add Screen ...', and 'Remove Screen' buttons. The main area displays a mobile app mockup with a 'Screen1' header, a 'Text for Button1' label, and a 'Non-visible components' section at the bottom showing 'SpeechRecognizer1'.
- Components:** Lists the components currently on the screen: 'Screen1', 'Button1', 'Canvas1', and 'SpeechRecognizer1'. It includes 'Rename' and 'Delete' buttons at the bottom.
- Properties:** Shows the properties for the selected 'Button1' component. Properties include 'BackgroundColor' (Default), 'Enabled' (checked), 'FontBold' (unchecked), 'FontItalic' (unchecked), 'FontSize' (14.0), 'FontTypeface' (default), 'Image' (None...), 'Shape' (default), 'ShowFeedback' (checked), 'Text' (Text for Button1), and 'TextAlignment' (center).

Overlaid text labels identify the main sections: 'Paleta de Herramientas' (Tools Palette) points to the Palette, 'Editor de la Interfaz de Usuario' (User Interface Editor) points to the Viewer, 'Mis componentes' (My components) points to the Components panel, and 'Propiedades' (Properties) points to the Properties panel.

Bloques

Variables



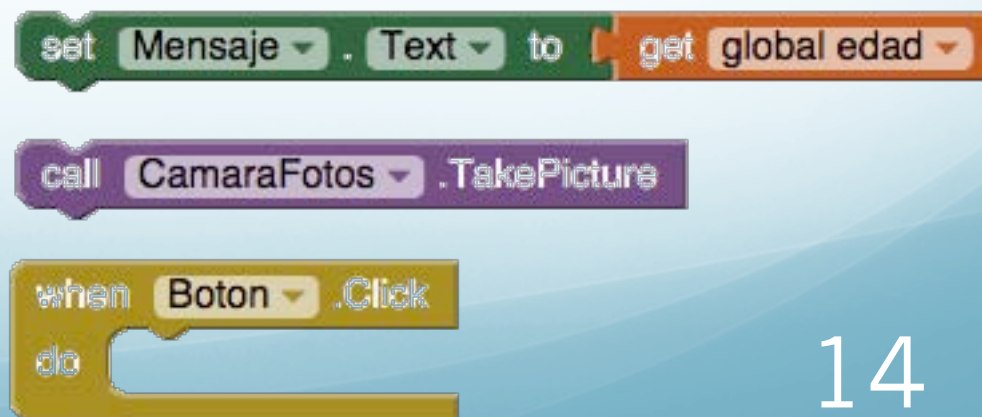
Control



Matemáticas



Específicos de cada componente



App Inventor: Editor de bloques

The screenshot displays the MIT App Inventor 2 Beta web interface. At the top, the navigation bar includes the MIT App Inventor logo, the text 'MIT App Inventor 2 Beta', and menu items: 'Projects', 'Connect', 'Build', and 'Help'. On the right side of the navigation bar are links for 'My Projects', 'Guide', 'Report an Issue', and a user profile icon with the email 'ruizrube@gmail.com'. Below the navigation bar is a green header for the project named 'TalkToMe', which contains tabs for 'Screen1', 'Add Screen ...', and 'Remove Screen'. On the far right of this header are 'Designer' and 'Blocks' tabs. The interface is divided into three main sections. On the left is the 'Blocks' panel, which lists categories: 'Built-in' (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures) and 'Screen1' (Button1, CheckBox1, TextToSpeech1, AccelerometerSensor1). At the bottom of this panel are 'Rename' and 'Delete' buttons. In the center is the 'Viewer' panel, which shows a list of available blocks for the selected screen. These include 'when Button1 .Click', 'when Button1 .GotFocus', 'when Button1 .LongClick', 'when Button1 .LostFocus', 'when Button1 .TouchDown', 'when Button1 .TouchUp', 'Button1 . BackgroundColor', 'set Button1 . BackgroundColor to', and 'Button1 . Enabled'. On the right is the 'Espacio de trabajo' (workspace), which shows the blocks being assembled into a program. The current program consists of two event-driven blocks. The first block is 'when Button1 .Click' followed by a 'do' block containing 'call TextToSpeech1 .Speak' with a message of 'TextBox1 . Text'. The second block is 'when AccelerometerSensor1 .Shaking' followed by a 'do' block containing 'call TextToSpeech1 .Speak' with a message of 'bailandoooo'. A trash can icon is located at the bottom right of the workspace.

MIT App Inventor 2
Beta

Projects Connect Build Help

My Projects Guide Report an Issue ruizrube@gmail.com

TalkToMe Screen1 Add Screen ... Remove Screen Designer Blocks

Blocks

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

Screen1

- Button1
- CheckBox1
- TextToSpeech1
- AccelerometerSensor1

Any component

Rename Delete

Media

Upload File ...

Viewer

when Button1 .Click

do

when Button1 .GotFocus

do

when Button1 .LongClick

do

when Button1 .LostFocus

do

when Button1 .TouchDown

do

when Button1 .TouchUp

do

Button1 . BackgroundColor

set Button1 . BackgroundColor to

Button1 . Enabled

when Button1 .Click

do

call TextToSpeech1 .Speak

message TextBox1 . Text

when AccelerometerSensor1 .Shaking

do

call TextToSpeech1 .Speak

message " bailandoooo "

Espacio de trabajo

Demo

- ¡Manos a la obra!



Demo 1

- APP para hacer fotos y “retocarlas” para luego poder compartirlas por redes sociales
- Componentes empleados
 - User Interface: *Button, Label, Spinner*
 - Layout: *HorizontalArrangement*
 - Media: *Camera*
 - Drawing and Animation: *Canvas*
 - Social: *Sharing*

Demo 2

- APP que permita participar en cuestionarios y almacenar los resultados de forma externa
- Herramientas empleadas
 - User Interface: *Button, Label, ListView, Notifier*
 - Layout: *HorizontalArrangement*
 - Media: *TextToSpeech, Sound*
 - Storage: *FusionTablesControl*
 - Sensor: *AccelerometerSensor, LocationSensor, Clock*

gracias

ivan.ruiz@uca.es