

# Fundamentos de VR-AR mediante aplicaciones Gráficas 3D en Android utilizando OpenGL- Parte I

## 3er Foro Nacional de Tecnologías de la Información y Sistemas Computacionales

Dr. Marco Aurelio Nuño Maganda

Universidad Politécnica de Victoria  
Laboratorio de Sistemas Inteligentes  
[mnunom@upv.edu.mx](mailto:mnunom@upv.edu.mx)

28 y 29 de Septiembre de 2023

# Contenido

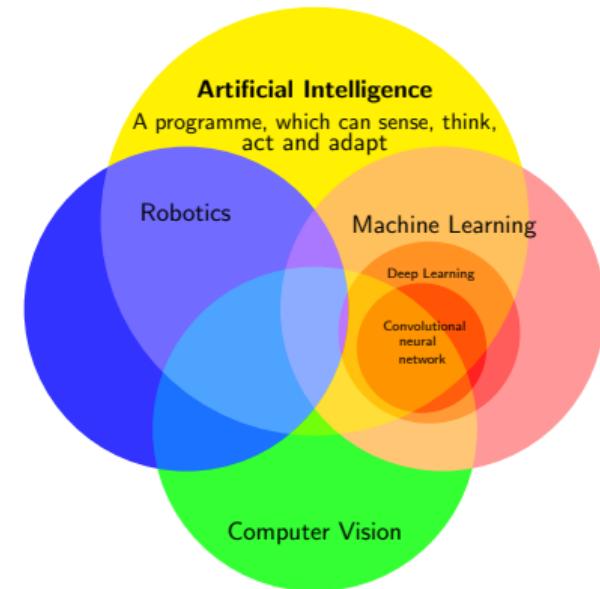
- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

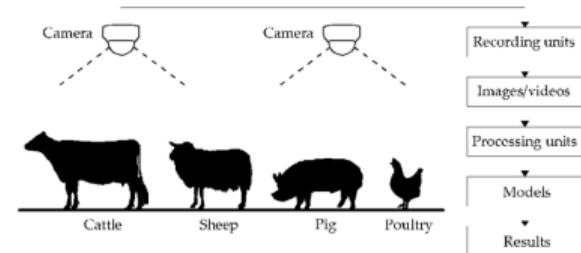
# Inteligencia Artificial

- Es un subconjunto de las ciencias de la computación (CC) en donde las máquinas aparentan inteligencia mediante la ejecución de programas.
- Las computadoras intentan resolver tareas que solo son posibles mediante inteligencia humana.
- Es una rama muy amplia que cubre muchos aspectos de la vida cotidiana.



# Visión por Computadora

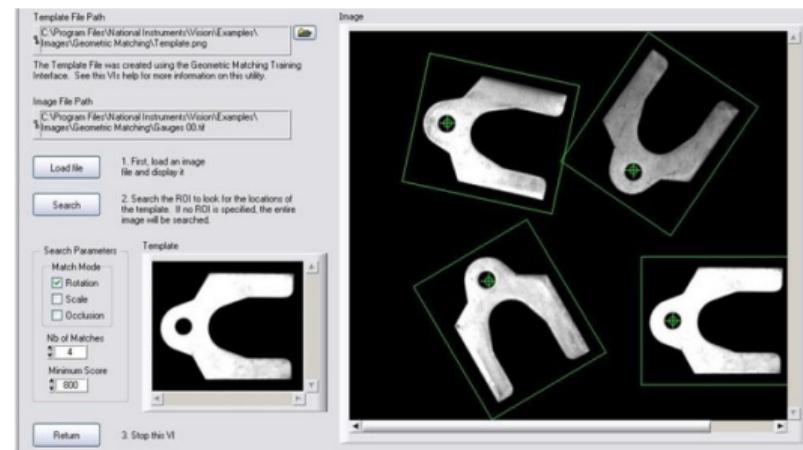
- La visión por computadora imita la percepción humana y las capacidades de razonamiento.
- Se intenta inferir conocimiento a partir de imágenes o videos.
- Procesamiento de imágenes es una fase temprana de la VC. La entrada es una imagen y la salida es otra imagen.
- En la visión por computadora, la entrada es una imagen pero la salida son datos.



# Visión por Computadora (Template Matching)

Permite comparar imágenes aplicando operaciones de comparación entre regiones.

- Algoritmos clásicos basados en correlación: SAD, SSD, Rank.
  - Ventajas: Facil de entender, rápidos
  - Desventajas: Sensibles a la escala, a rotaciones y a cambios en la iluminación.
- Algoritmos modernos basados Aprendizaje Profundo
  - Ventajas: Buen desempeño
  - Desventajas: Requieren de entrenamiento con muchas imágenes



# Gráficos por Computadora

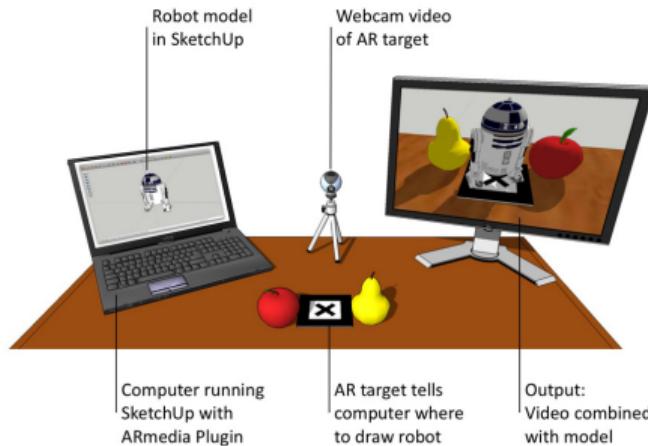
- Es la rama de las CC encargada de la producción de imágenes y animaciones empleadas en juegos de computadora y simulaciones, en algunos casos incluyendo elementos fotorealisticos.
- Se requieren conocimientos de geometría, álgebra, cálculo, física, programación (estructura de datos).
- Existen librerías de bajo nivel (OpenGL) hasta frameworks (Unity, Unreal).



# Realidad Aumentada (AR)

- La AR es una experiencia que traslapa elementos digitales (modelados por computadora) con el mundo físico del usuario (adquirido mediante una cámara).
- Los elementos digitales se combinan con las vistas del mundo real.

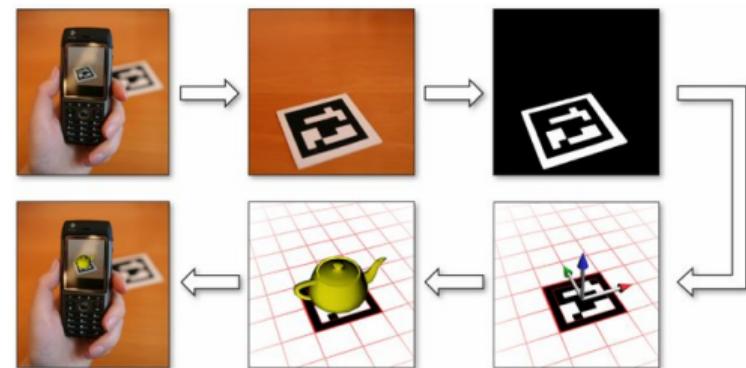
## HOW IT WORKS



\* <http://photos1.blogger.com/img/m-a310d3b7b46f285189e1d6da63a1af13be4fffc4.jpg>

# Pasos en la detección de marcadores de AR

- ① Umbralización.
- ② Detección del Marcador.
- ③ Estimación de Pose y Posición.
- ④ Empalme del modelo 3D.



\* Daniel Wagner and Dieter Schmalstieg. "ARToolKitPlus for Pose Tracking on Mobile Devices". In: *Proceedings of 12th Computer Vision Winter Workshop CVWW07*. Jan. 2007

# Tipos de Aplicación de AR

- Basadas en Localización. Están basadas en sensores GPS para determinar la ubicación del dispositivo para crear objetos AR
- Basadas en Visión – Utilizan una cámara, aunque también es posible incorporar sensores (compass, acelerómetros, giroscopios, etc).
  - Requieren Marcadores (Marker) – Localizan un patrón o marcador QR y renderizan un objeto 3D basado en su localización en el espacio real.
  - No requieren marcadores (Markerless)– Se emplean esquinas y puntos característicos del espacio real

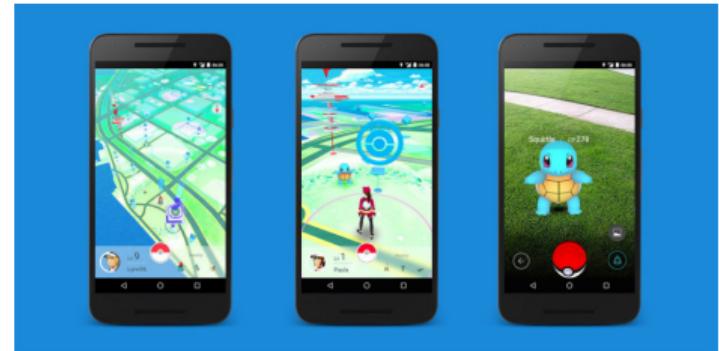
# Aplicaciones de RA

- Aplicaciones principales: Arquitectura, Cosméticos, Contenido social, Marketing, Juegos, etc

- Houzz



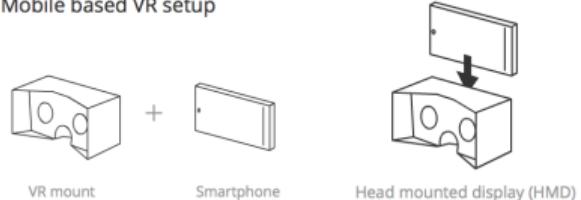
- Pokemon Go



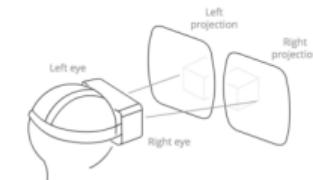
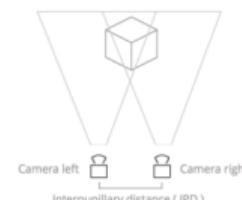
# Realidad Virtual (VR)

- La Realidad virtual (RV), el uso de modelos y simulaciones por ordenador que permite a una persona interactuar con un entorno artificial tridimensional (3-D) visual
- En un formato típico de RV, un usuario lleva un casco con una pantalla estereoscópica para ver imágenes animadas de un entorno simulado
- Reconstrucción de 3D es realizada por nuestro cerebro.

Mobile based VR setup



How to create stereoscopic 3D images



\* [https://reference.codeproject.com/book/dom/webvr\\_api/webvr\\_concepts](https://reference.codeproject.com/book/dom/webvr_api/webvr_concepts)

# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Computadora y Programabilidad

## Computadora

Es una máquina (electrónica) programable\* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

# Computadora y Programabilidad

## Computadora

Es una máquina (electrónica) programable\* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

## Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

# Computadora y Programabilidad

## Computadora

Es una máquina (electrónica) programable\* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

## Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

## Programación

El proceso de crear un software utilizando un lenguaje de programacion (C, C++, Java, Python, Kotlin, etc)

# Computadora y Programabilidad

## Computadora

Es una máquina (electrónica) programable\* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

## Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

## Programación

El proceso de crear un software utilizando un lenguaje de programacion (C, C++, Java, Python, Kotlin, etc)

## Programa

Un conjunto de instrucciones que una computadora interpreta en una secuencia logica para llevar a cabo una tarea en particular

# Computadora y Programabilidad

## Computadora

Es una máquina (electrónica) programable\* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

## Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

## Programación

El proceso de crear un software utilizando un lenguaje de programacion (C, C++, Java, Python, Kotlin, etc)

## Programa

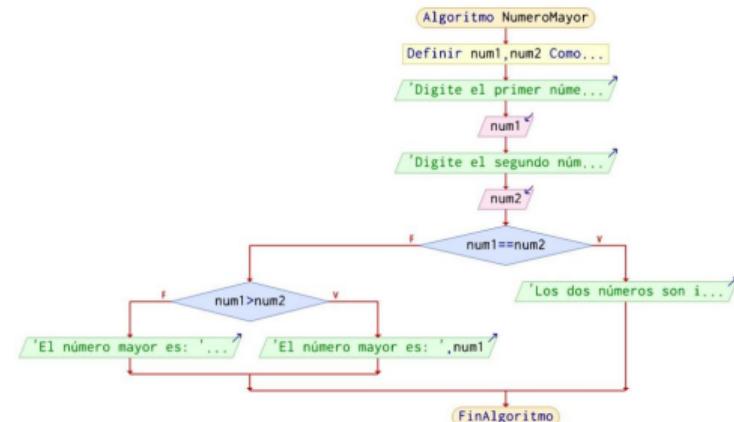
Un conjunto de instrucciones que una computadora interpreta en una secuencia logica para llevar a cabo una tarea en particular

# Diagrama de Flujo

Un diagrama de flujo es una representación gráfica de un algoritmo, los pasos que los componen y la secuencia de ejecución de sus instrucciones

## Algoritmo

Diseñar un algoritmo para comparar dos números



# Codificación de un algoritmo en varios lenguajes de programación (Python - Kotlin)

## Codificación del Algoritmo en Python

```

1 def CompararNumeros(A,B):
2     if (A==B):
3         print ("Son Iguales ")
4     else:
5         if (A>B):
6             print ("A es mayor que B ")
7         else:
8             print ("B es mayor que A ")
9
10 As = input ("Escribe el primer numero (A): ")
11 A = int(As)
12 Bs = input ("Escribe el segundo numero (B): ")
13 B = int(Bs)
14 CompararNumeros(A,B)

```

## Codificación del Algoritmo en Kotlin

```

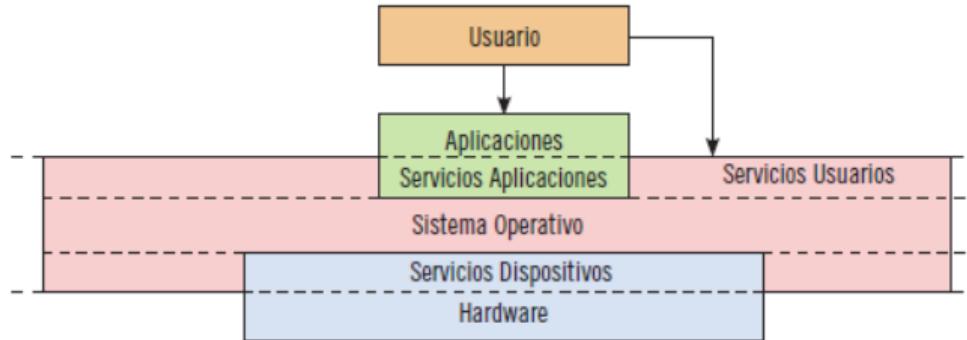
1 package com.example.myapplication
2 fun CompararNumeros (A : Int, B: Int) {
3     if (A==B) {
4         print ("Son Iguales")
5     } else {
6         if (A > B) {
7             print ("A es mayor que B")
8         } else {
9             print ("B es mayor que A")
10        }
11    }
12 }
13 fun main() {
14     print("Escribe el primer numero (A):")
15     val As = readLine()!!
16     val A= Integer.parseInt(As)
17     print("Escribe el segundo numero (B):")
18     val Bs = readLine()!!
19     val B= Integer.parseInt(Bs)
20     CompararNumeros(A,B)
21 }

```

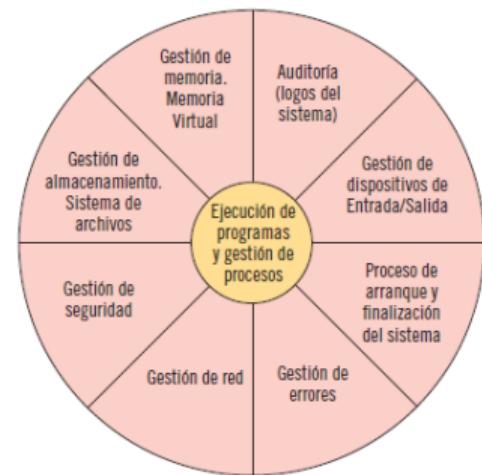
# Sistema Operativo

Un Sistema Operativo (SO) es un programa (software) que al arrancar la computadora\*\* se encarga de gestionar todos los recursos del sistema informático permitiendo así la comunicación entre el usuario y la computadora.

Estructuración de los servicios del sistema operativo



Bloques funcionales de un sistema operativo



# Sistemas Operativos para PCs



# Telefono Celular No-inteligente vs Telefono Celular Inteligente

## Teléfono No-inteligente

- Su funcionalidad principal era la comunicación (llamadas o mensajes) a través de la red celular (GSM)

## Teléfono inteligente

- Interfaz de entrada: Pantalla Touch (a color, de alta definición)
- Conexión a Internet: WiFi, GSM (4G o 5G)
- Comunicación con otros dispositivos: Bluetooth, NFC
- Cámaras (Frontal y Posterior)



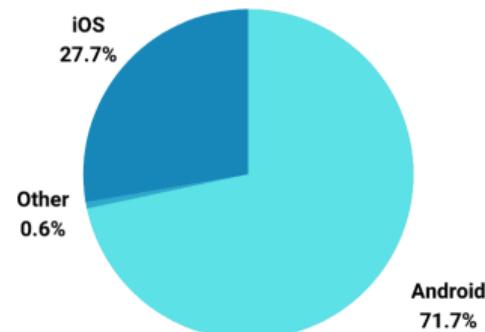
# Sistemas Operativos para Teléfonos Inteligentes



Windows®  
phone

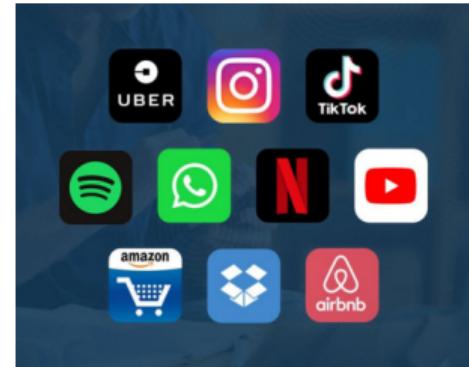
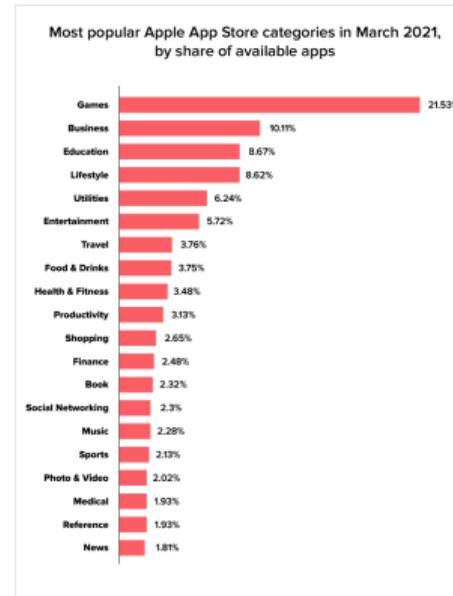
# Android

- Android es un sistema operativo móvil basado en Linux
- Principalmente orientado a dispositivos de pantalla táctil (Smartphone, tablets, smartwatches, etc)
- Fue desarrollado por Android Inc (Adquirida por Google en 2005)
- Vinculado con un grupo de empresas (HTC, Sony, Motorola, Samsung, LG, Lenovo, entre otras) para la creación de un SO común para sus dispositivos
- A la fecha (Q1 2023), los teléfonos con SO Android concentran mas del 70% del mercado global.



# Aplicaciones Móviles

- Ejecutadas en el teléfono
- La entrada de datos es mediante un teclado “virtual”
- El apuntador del ratón es la pantalla
- Incluyen una interfaz de usuario gráfica (GUI)
- Es posible descargar miles de éstas en nuestros dispositivos



<https://www.netsolutions.com/insights/top-10-most-popular-apps-2018/>

# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 **Herramientas Requeridas**
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Android Studio

- Android Studio es un entorno oficial de desarrollo integrado (IDE) para el sistema operativo Android de Google
- La primera versión se libera en el año 2013, siendo el lenguaje de programacion Java
- En 2019, se reemplaza el lenguaje oficial de desarrollo por Kotlin, aunque Java todavía es soportado
- Es gratis, se puede descargar e instalar en cualquier computadora sin importar el sistema operativo (Windows, Linux y MacOS)  
<https://developer.android.com>

# OpenCV para Android

- OpenCV es una librería que incluye operadores de vision por computadora en tiempo real
- Originalmente creada por Intel en 1999
- Es multiplataforma y open-source (Licencia Apache)
  - Se descarga la versión para Android del siguiente enlace:  
<https://opencv.org/releases/>
  - Es necesario descomprimir, ya que se va a utilizar para algunos demos que seran vistos mas adelante

# Otras herramientas software

Son opcionales, pero se recomienda instalarlas para una exploracion mas profunda de los conceptos que seran vistos en este curso. Todas son multiplataforma.

- Blender. Es un programa para modelar, renderizar y crear gráficos, pudienro ser exportados a formatos OBJ y GLB.
- MeshViewer. Programa para visualizar archivos de gráficos en formato GLB.  
(<https://github.com/RBFraphael/meshviewer/releases>)
- Scrcpy. Programa que permite duplicar en una computadora el escritorio de un teléfono Android conectado mediante USB

## Paginas de Internet

- SketchFab (<https://sketchfab.com/>). Permite obtener modelos 3D en diferentes formatos (particularmente en GLB) – Requiere Login con cuenta de Google,
- Free online OBJ to GLB converter.  
(<https://products.aspose.app/3d/conversion/obj-to-glb>) (En caso de no contar con Blender).

# Hardware requerido

- PC con Android Studio instalado
- Teléfono Inteligente con SO Android
- Cable USB
- Adaptador de Realidad Virtual (Opcional)
- Control de Videojuego Bluetooth (Opcional)

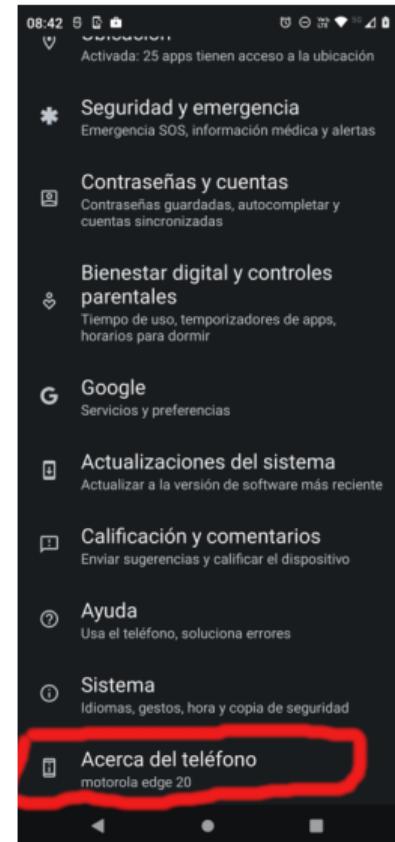
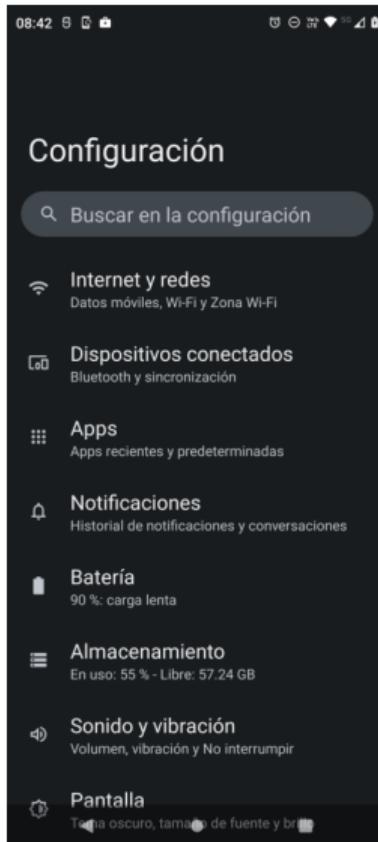
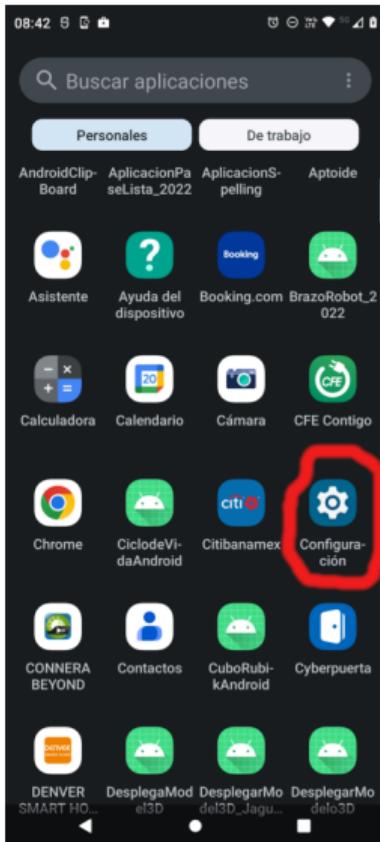
# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

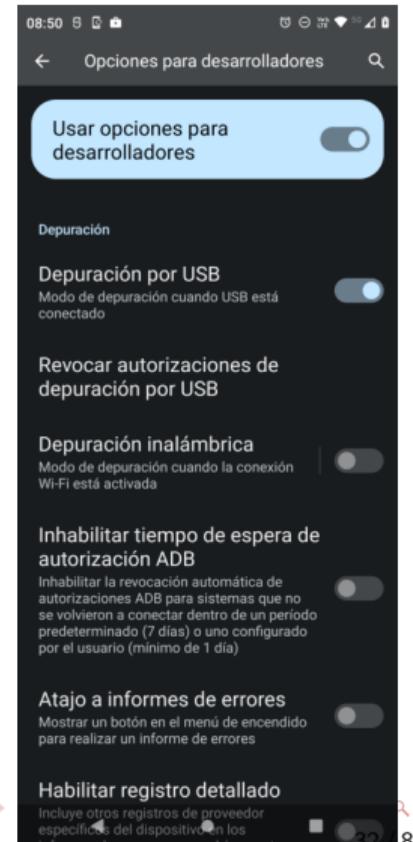
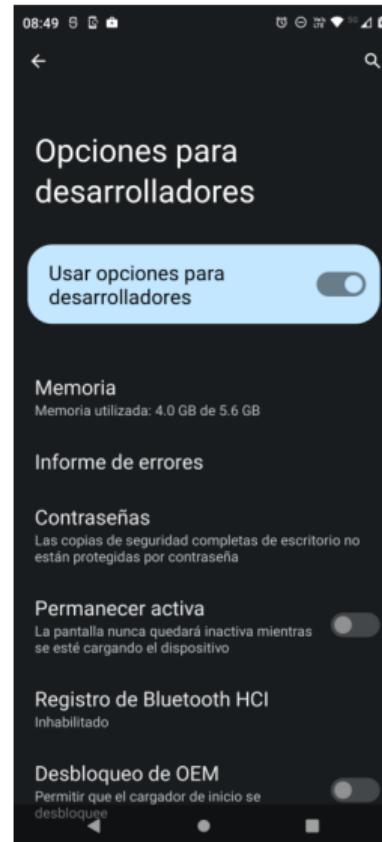
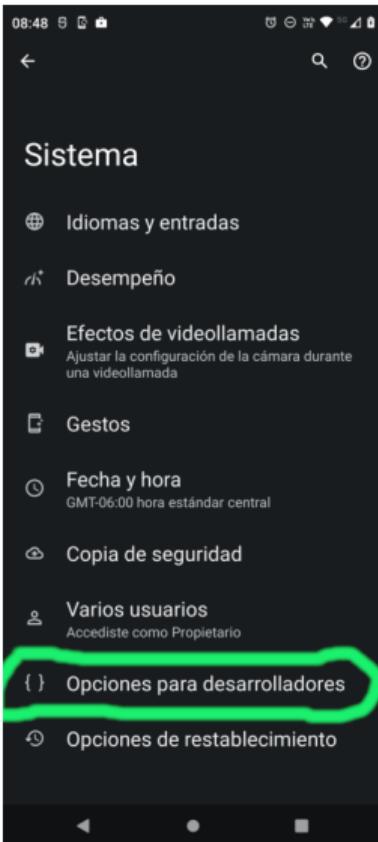
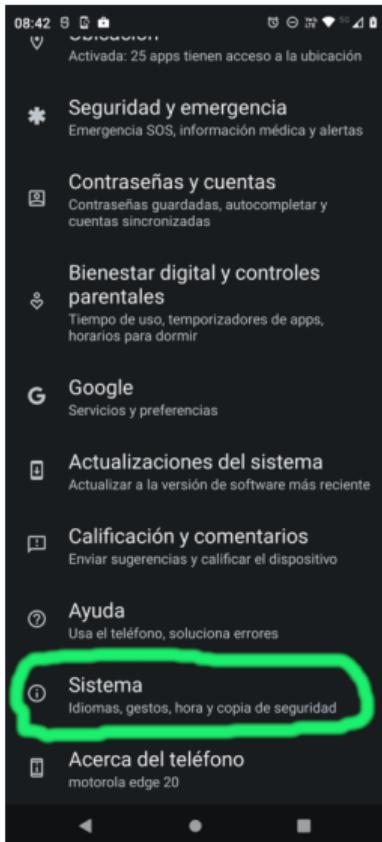
# Configurar telefono inteligente en modo de desarrollador (0)

- Es necesario buscar en las opciones de configuración, en el apartado acerca del telefono ubicar el número de compilación
- Dar 5 taps (toques) sobre el número compilación. Deberá aparecer un mensaje que diga que estás a X taps de ser un desarrollador
- Lo anterior habilita un nuevo menú en el apartado sistema dentro de opciones de configuración con título “opciones para desarrolladores”
- Debe habilitarse tanto “opciones para desarrolladores” como la opción “depuración USB”

# Configurar teléfono inteligente en modo de desarrollador (1)

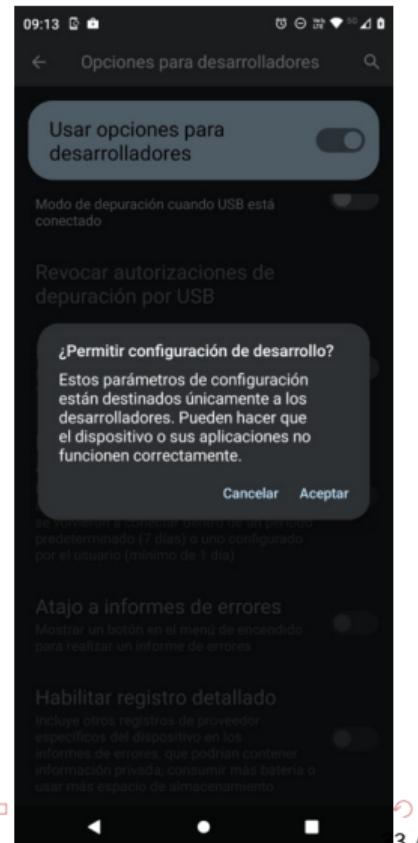


# Configurar teléfono inteligente en modo de desarrollador (2)



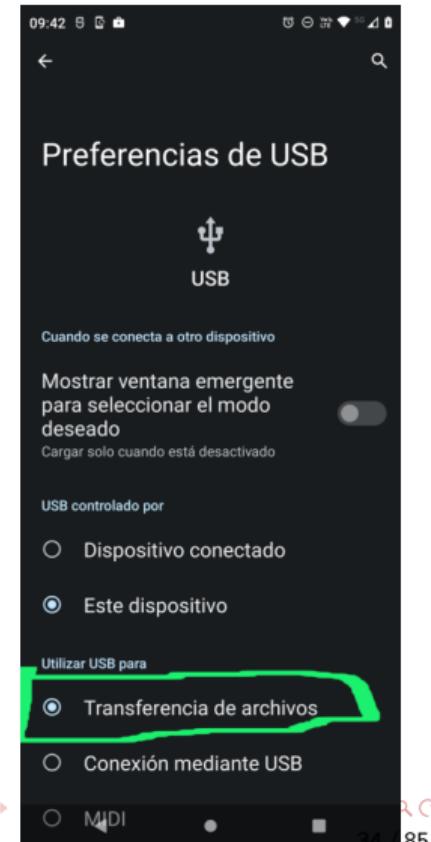
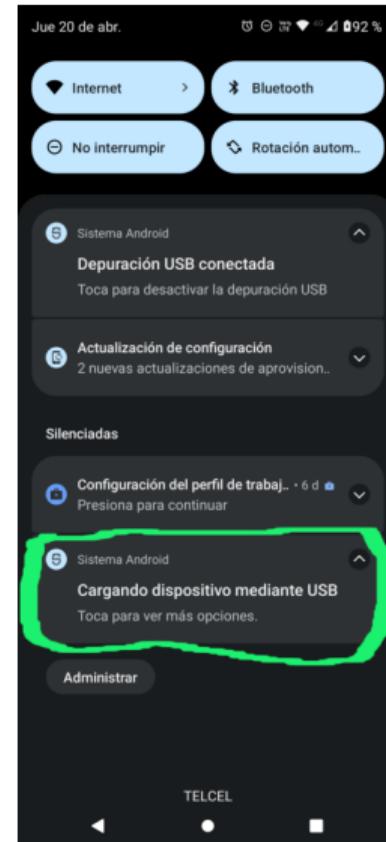
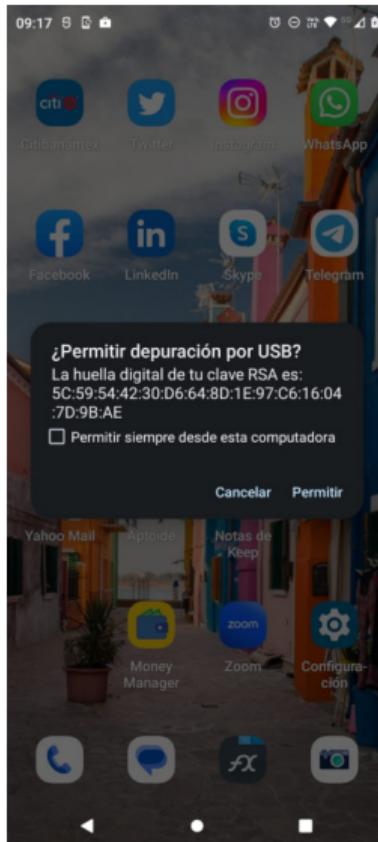
# Configurar teléfono inteligente en modo de desarrollador (3)

- Al habilitar la depuración, aparece un mensaje de advertencia
- Una vez que hayas terminado tus pruebas, se recomienda deshabilitar este modo de depuración
- Para poder instalar una aplicación desarrollada con Android Studio, se debe conectar el teléfono inteligente a la computadora usando un cable USB



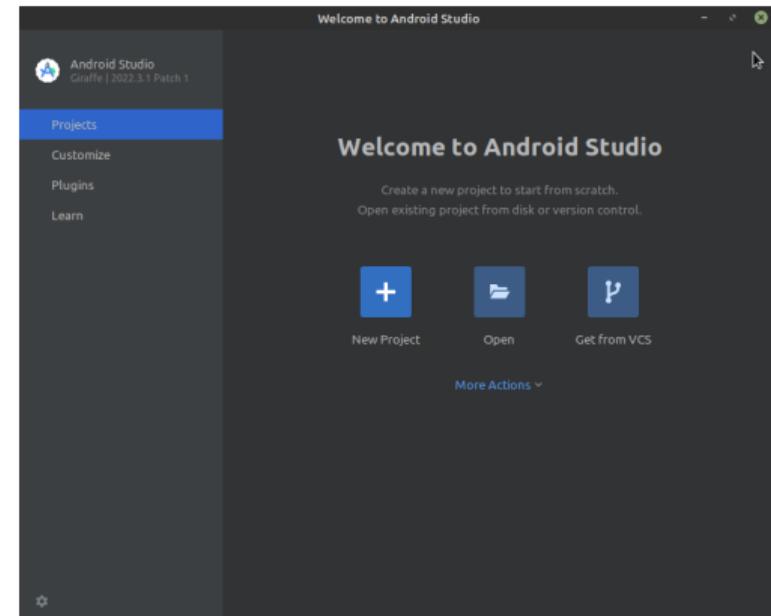
# Conectar smartphone a la computadora (Cable USB)

- Al conectar el dispositivo por primera vez, aparece un mensaje de confirmación en el dispositivo
- Configurar el teléfono para que se conecte en modo de transferencia de archivos (por defecto, esta cargando)



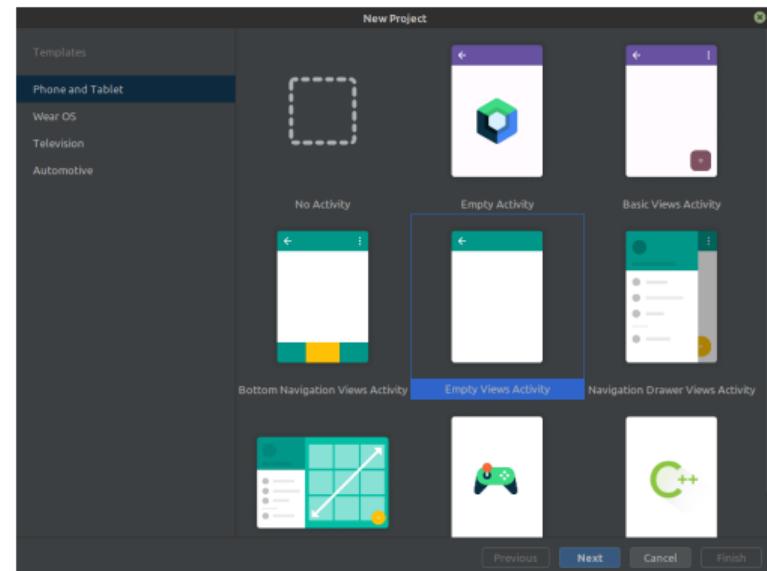
# Crear un primer proyecto (1)

- Buscar el icono de la aplicación y dar doble click
- En caso de que requiera algunas actualizaciones, ser paciente, puede tardar.
- Aparecerá una ventana como la mostrada, seleccionar la opción “New project”



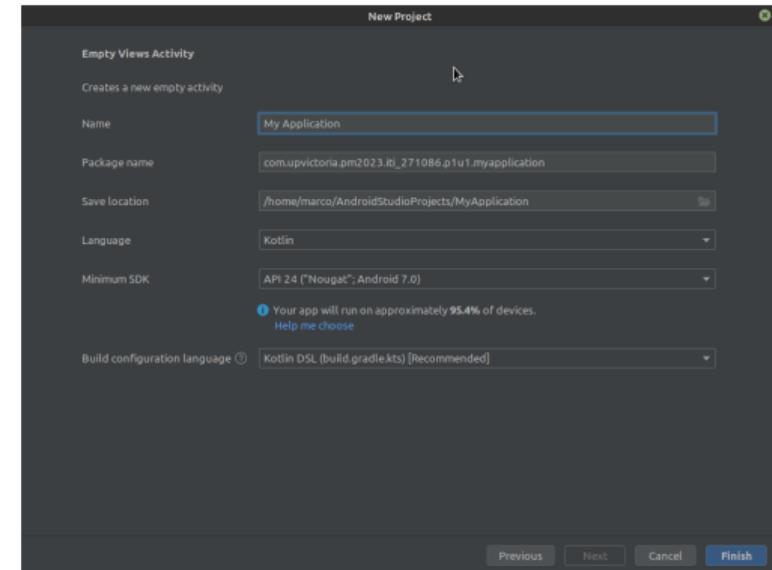
# Crear un primer proyecto (2)

- Existen varios tipos de dispositivos para los que pueden crearse proyectos. Por defecto nos ubica en el primer grupo (Phone and Tablet)
- Seleccionar tipo de proyecto “Empty Views Activity” y dack click en Next



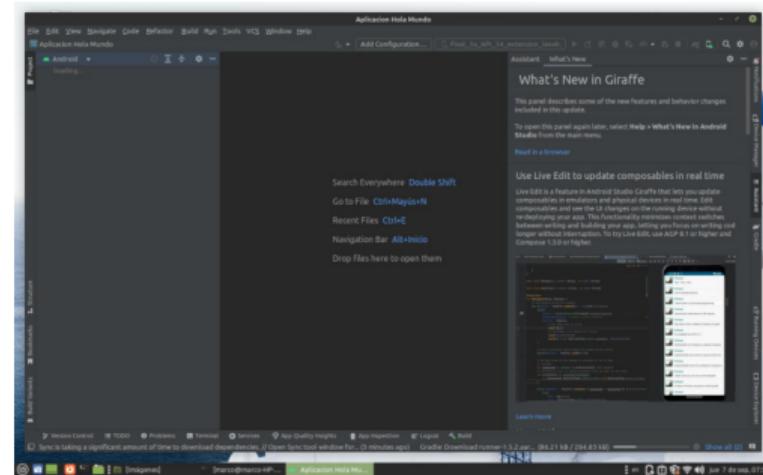
# Crear un primer proyecto (3)

- Dejar los valores por defecto, solo se sugiere cambiar el nombre del proyecto
- **Debe estar seleccionado como lenguaje Kotlin.** De estar seleccionado uno diferente, cambiar
- Presionar el botón Finish



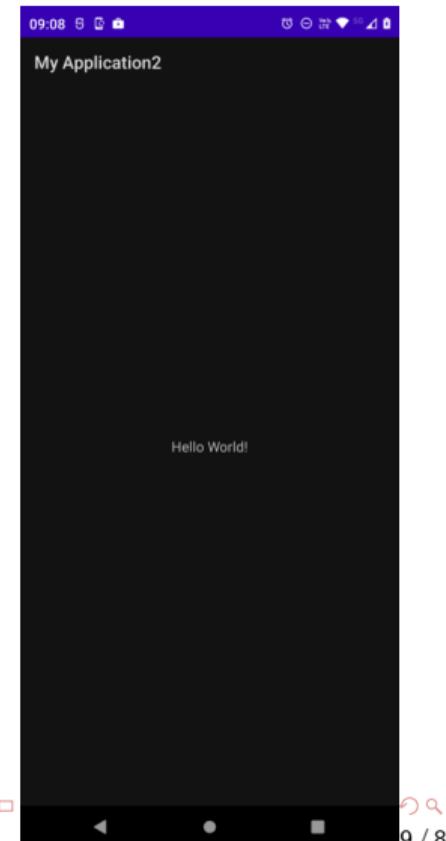
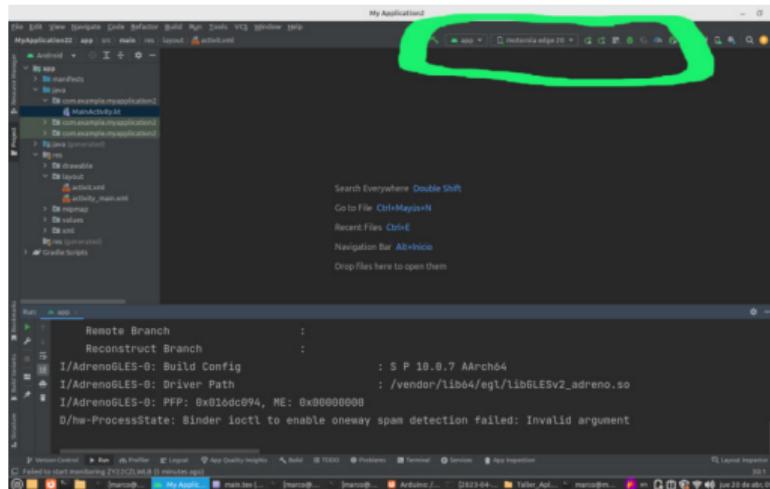
# Crear un primer proyecto (4)

- La primera vez que se crea un proyecto, puede requerir la descarga de archivos necesarios. Le recomiendo ser paciente
- Ya una vez que el proyecto fue creado, aparece la ventana mostrada



# Ejecutar proyecto en telefono inteligente

- Una vez conectado el telefono, debe aparecer el modelo en la parte superior (lado derecho) de tu proyecto de Android Studio
- Para instalar la aplicacion en el telefono, deber dar click en una flecha verde justo a un lado de nombre del telefono (sean pacientes, puede tardar un poco la primera vez)



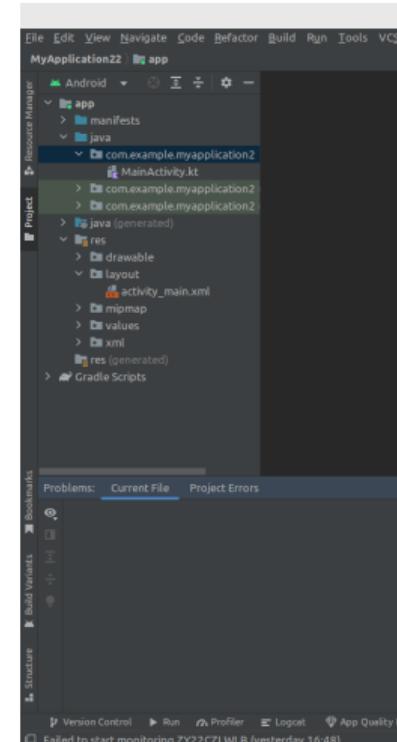
# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Carpetas del proyecto

## Carpetas y archivos importantes

- *Manifest* - Por el momento es de interés
- *Java* - Código fuente - Dentro hay un archivo *MainActivity.kt*
- *Res* - Recursos (imágenes) y definición de interfaces. Dentro hay una carpeta llamada *layout*, con un archivo *activity\_main.xml*



# Agregar código a la interfaz

## Apariencia

### Archivo activity\_main.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="Hello World!"
13        app:layout_constraintBottom_toBottomOf="parent"
14        app:layout_constraintEnd_toEndOf="parent"
15        app:layout_constraintStart_toStartOf="parent"
16        app:layout_constraintTop_toTopOf="parent" />
17 </androidx.constraintlayout.widget.ConstraintLayout>
```

## Comportamiento

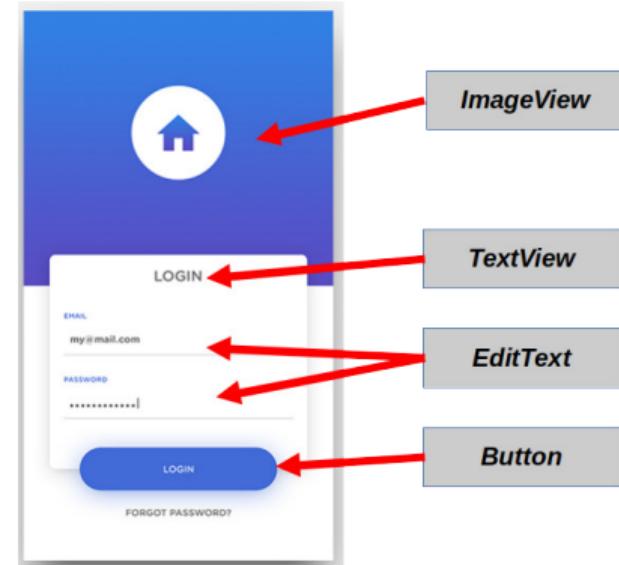
### Archivo MainActivity.kt

```

1 package com.example.primeraplicacionandroid
2 import androidx.appcompat.app.AppCompatActivity
3 import android.os.Bundle
4 class MainActivity : AppCompatActivity() {
5     override fun onCreate(savedInstanceState: Bundle?) {
6         super.onCreate(savedInstanceState)
7         setContentView(R.layout.activity_main)
8     }
9 }
```

# Interfaz de Usuario

- Define los controles de interfaz que la aplicación muestra
- Estos controles pueden estar agrupados en contenedores
- El archivo **activity\_main.xml** es la definición del layout (la organización de dichos componentes)



# Primer modificación

## Archivo Original **activity\_main.xml**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintEnd_toEndOf="parent"
15         app:layout_constraintStart_toStartOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17 </androidx.constraintlayout.widget.ConstraintLayout>
```

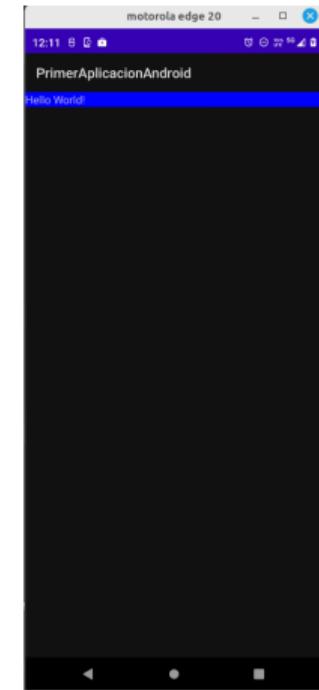
## Archivo Modificado **activity\_main.xml**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     tools:context=".MainActivity">
10    <TextView
11        android:background="#00ffff"
12        android:layout_width="match_content"
13        android:layout_height="wrap_content"
14        android:text="Hello World!" />
15 </LinearLayout>
```

# Primer modificación

- ① Reemplazar `androidx.constraintlayout.widget.ConstraintLayout` por `LinearLayout`
  - Tipo de contenedor donde los componentes se agregan secuencialmente
- ② Agregar propiedad `android:orientation="vertical"` al `LinearLayout` principal
  - Los componentes serán ordenados verticalmente
- ③ Cambiar la propiedad `android:layout_width="wrap_content"` por `android:layout_width="match_parent"`
  - El ancho de `TextView` abarcar toda la pantalla
- ④ Agregar la propiedad `android:background="#00ffff"` al `textview`
  - - La cadena `#00ffff` define el color azul. Se pueden probar con otros buscando en Internet un generador de colores hexadecimal y cambiando por otros de tu preferencia

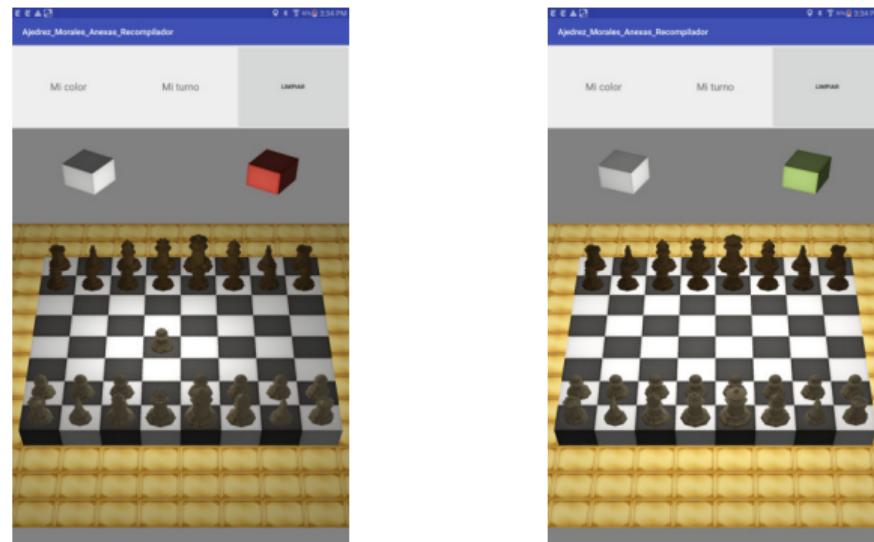


# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Multiplayer Chess

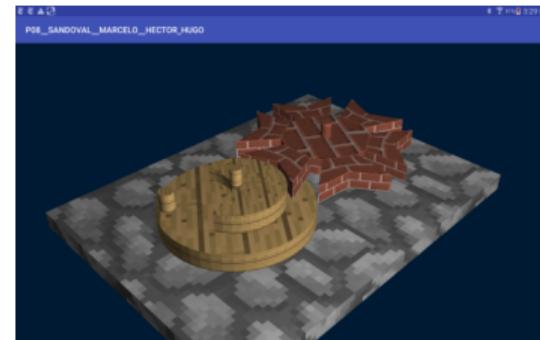
- Cada pieza fue modelada en Blender y exportada a la aplicación de Android
- Aplicación multidispositivo, que permite llevar una partida de ajedrez.
- El control del juego queda del lado del servidor.



\* Carlos Eduardo Garcia-Garcia, Jesus Antonio Luna-Alvarez, Hector Hugo Sandoval-Marcelo, and Jose Eduardo Torres-Montalvo. *Ajedrez Multijugador*. Informe técnico proyecto de asignatura “Graficación por Computadora Avanzada”. Universidad Politécnica de Victoria, 2017.

# Simulación del Mecanismo de la Cruz de Malta

- Se modelaron los componentes individuales 3D (cilindros, tetraedros, etc)
- Se incorporó texturas, iluminación y sombras
- Es posible ver el modelo desde múltiples perspectivas



\* Héctor Hugo Sandoval-Marcelo. *Movimiento de la Cruz de Malta con Iluminación y Texturas con Java y OpenGL ES 2.0 en Android*. Informe técnico proyecto de asignatura “Graficación por Computadora Avanzada”. Universidad Politécnica de Victoria, 2017

# Aplicación Tetris 3D (1)

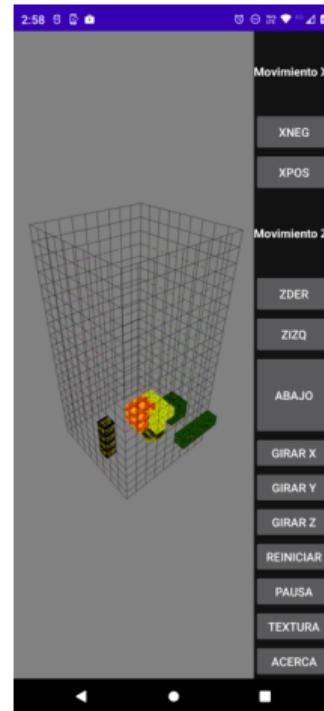
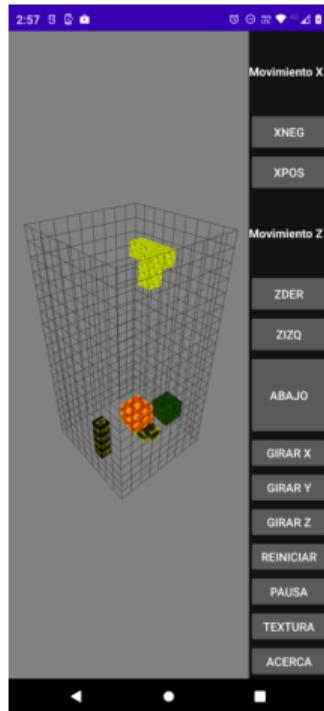
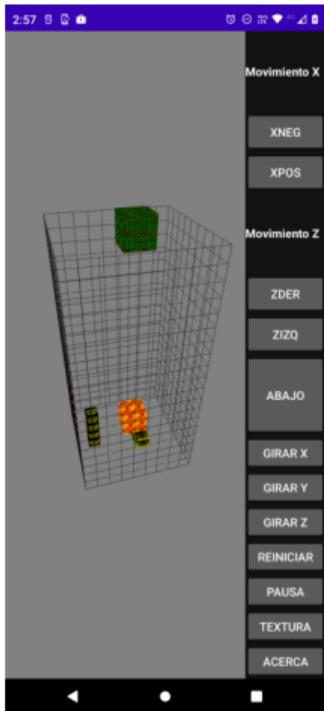
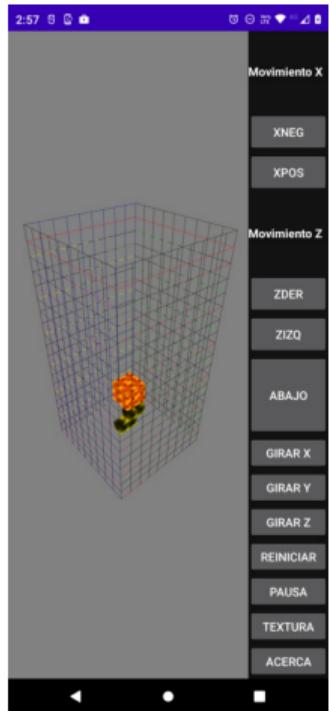
## Motivación

Este proyecto esta inspirado en un tetris 3D para PC http:

//users.csc.calpoly.edu/~zwood/teaching/csc471/finalproj24/gzipkin/

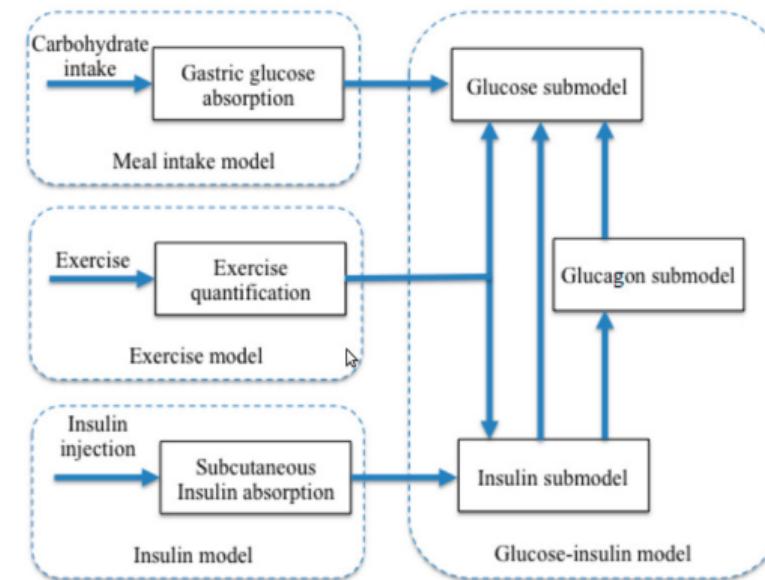
- Varias piezas caen sobre una rejilla y el objetivo es borrar las lineas.
- Las piezas se mueven en un entorno tridimensional (ejes X, Y y Z)
- Las fronteras de la red estan definidas mediante una malla alámbrica
- La cámara tambien se puede mover en base a eventos de toque sobre la pantalla del dispositivo

# Aplicación Tetris 3D (2)



# “Medical Assistant Mobile Application for Diabetes Control by Simulating a Compartmental Model” (1)

- Es necesario herramientas para educar a pacientes con Diabetes para manejar su enfermedad.
- Se propone una aplicación móvil que permita generar una simulación del comportamiento de la glucosa en un organismo con Diabetes a partir de modelos:
  - Dosificación de insulina
  - Ejercicio
  - Ingesta de alimentos

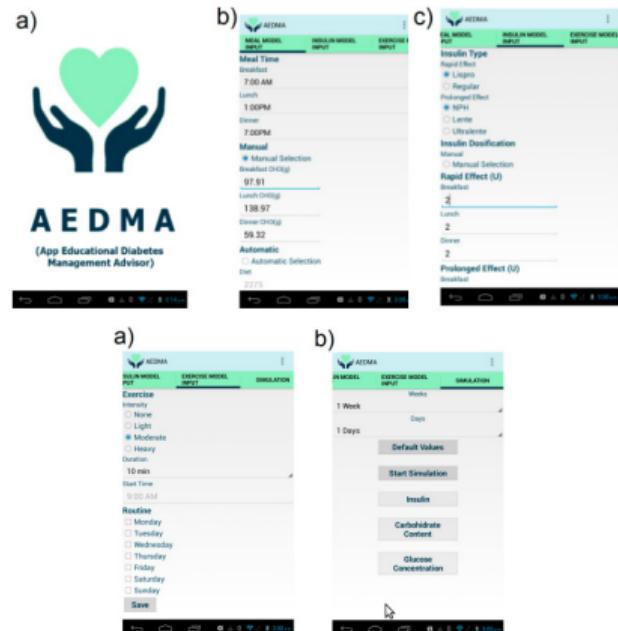


\* Martín Hernández-Ordoñez, Marco Aurelio Nuño-Maganda, Carlos Adrián Calles-Arriaga, Abelardo Rodríguez-León, Guillermo Efren Ovando-Chacon, Rolando Salazar-Hernández, Omar Montaño-Rivas, and José Margarito Canseco-Cortinas. “Medical Assistant Mobile Application for Diabetes Control by Simulating a Compartmental Model”. In: *Applied Sciences* 10.19 (Oct. 2020). <https://doi.org/10.3390/app10196846>. ISSN: 2076-3417. DOI: 10.3390/app10196846. URL: <https://www.mdpi.com/2076-3417/10/19/6846>

# “Medical Assistant Mobile Application for Diabetes Control by Simulating a Compartmental Model” (2)

Pantallas de la aplicación:

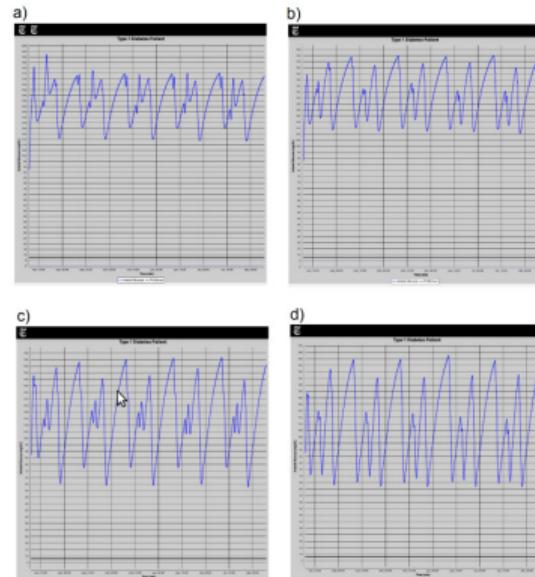
- ① Pantalla inicial
- ② Configuración de ingestión de alimentos
- ③ Configuración de dosificación de insulina
- ④ Configuración de configuración de rutina de ejercicio
- ⑤ Inicio de simulación



# “Medical Assistant Mobile Application for Diabetes Control by Simulating a Compartmental Model” (3)

La aplicación genera una gráfica de comportamiento de la glucosa a lo largo del tiempo de simulación. Se muestran las gráficas para cuatro casos:

- Pacientes controlados en cuanto a los niveles de glucosa (a y b)
- Pacientes fuera de control debido a la falta de ejercicio y al exceso de carbohidratos (c y d)



# Detección de movimientos en un tablero de ajedrez

Aplicación de escritorio:

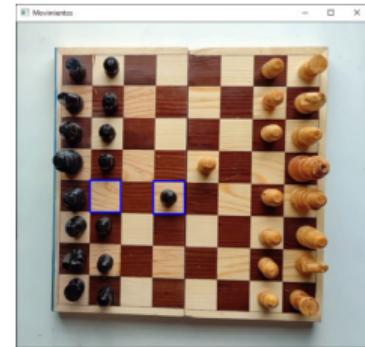
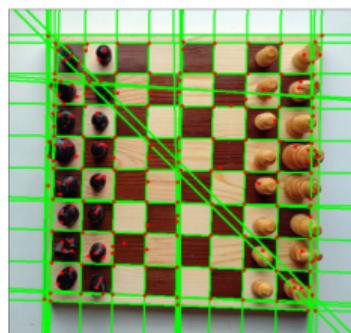
- Entorno semicontrolado con una cámara y una Laptop con OpenCV
- Detecta las esquinas del tablero de ajedrez
- Transformada de Hough
- Detectar si hay casilla o no dentro de la región de interés



# Detección de movimientos en un tablero de ajedrez (2)

Aplicación de escritorio:

- Una vez detectadas las regiones, se emplea un cronómetro para determinar la diferencia entre dos instantáneas consecutivas
- Problemas actuales:  
Iluminación, sombras,  
occlusiones



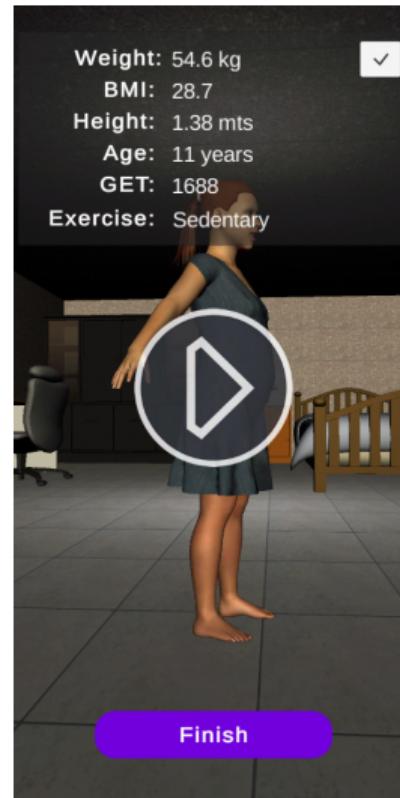
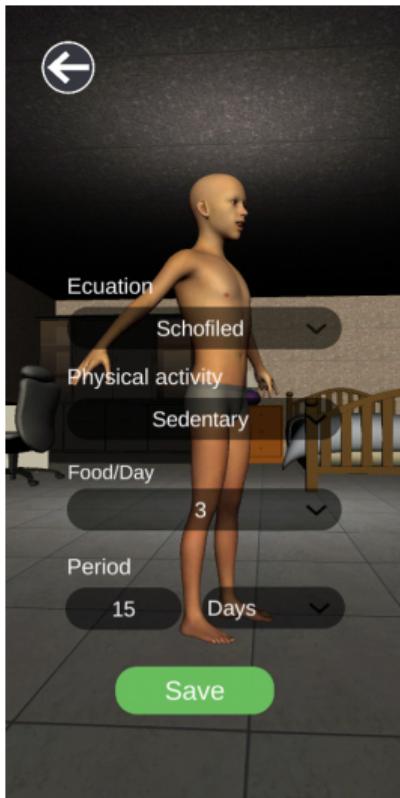
# Mobile Application for Overweight Education (1)

- Se requieren herramientas que apoyen en la concientización de problemas médicos graves (Obesidad)
- Se integra una aplicación móvil que muestre un avatar tridimensional cuyo peso, estatura y edad sea configurable con por usuario
- Es posible configurar el numero de comidas y el tipo de ejercicio de la simulación
- Al finalizar la simulación, la aplicación mostrará el antes o despues basado en los IMCs y los parametros del avatar.
- Herramientas utilizadas
  - Unity
  - MakeHuman

# Mobile Application for Overweight Education (2)

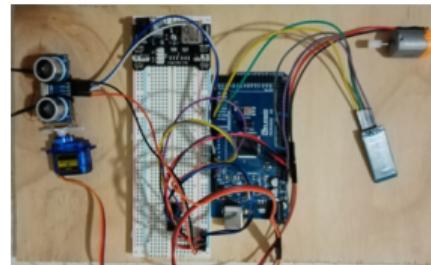


# Mobile Application for Overweight Education (3)



# Integración de un escáner 3D con sensores ultrasónicos y una aplicación móvil para visualización de modelos tridimensionales\* (1)

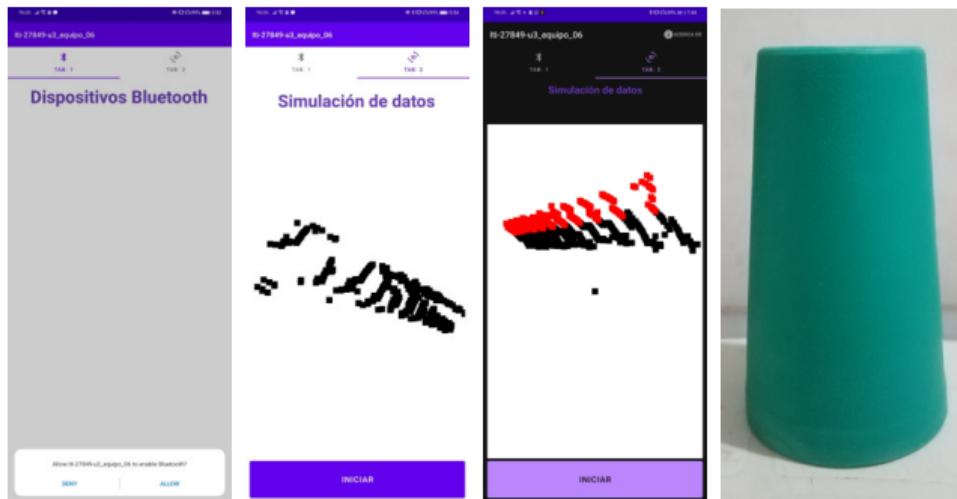
- Muchas de las tareas que tradicionalmente se hacían en una PC ahora han migrado hacia el teléfono inteligente
- La digitalización de objetos 3D ha tenido un auge en tiempos recientes
- El escáner se obtiene datos de un sensor ultrasónico montado en un motor (escaneo vertical)
- Una plataforma giratoria rota el objeto para obtener los datos desde todos los ángulos



\* Axel Issai Alemán-Delgado, Isaac De Jesús Becerra-Martínez, Karen Lizeth García-Rivera, Daniela Estefanía Huerta-Vázquez, Didre Dayan Mota-González, Carlos Alejandro Reyes-Puga, and Marco Aurelio Nuño-Maganda. *Integración de un escáner 3D con sensores ultrasónicos y una aplicación móvil para visualización de modelos tridimensionales*. Informe técnico proyecto de asignatura "Programación Móvil". Universidad Politécnica de Victoria, 2022.

# Integración de un escáner 3D con sensores ultrasónicos y una aplicación móvil para visualización de modelos tridimensionales (2)

- La aplicación visualiza el modelo adquirido y lo almacena en un formato estándar
- Se propone una App para un escáner 3D “casero” mediante comunicación Bluetooth



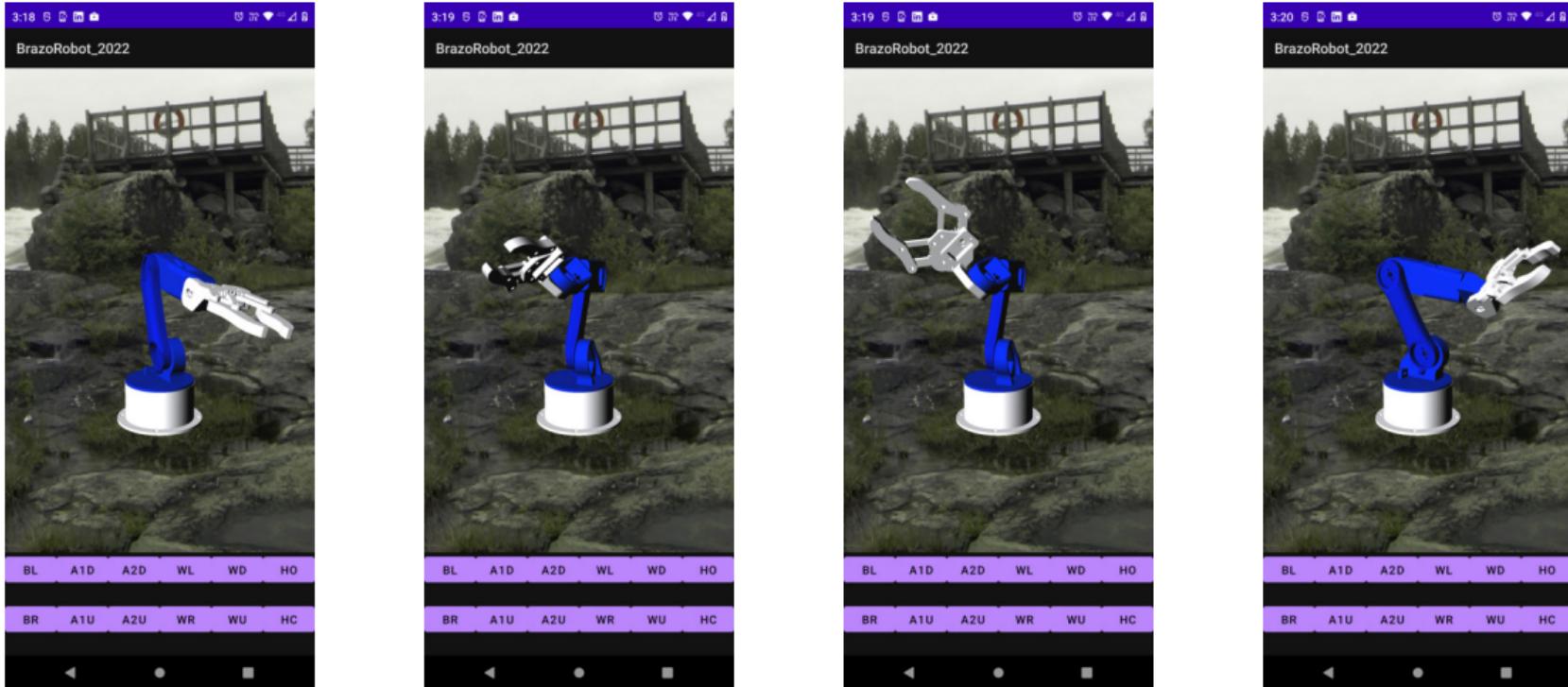
# Brazo Robótico Versión 1

- Se retoma un diseño previamente realizado para WebGL.
- Los componentes del robot son movidos mediante motores, y puede ser visto desde diferentes perspectivas.



\* José Carlos Morin-Garcia, Froylán Melquiades Wbario-Martinez, and Mariela Georgina Reyes-Fonseca. *Implementación de un Modelo 3D de un Brazo Robótico en una Aplicación Android*. Informe técnico proyecto de asignatura “Graficación por Computadora Avanzada”. Universidad Politécnica de Victoria, 2019

# Brazo Robótico Versión 2



\* Dino Tonatiuh Gutierrez-Martinez, Marly Yuleth Cantú-Charles, and Jesus Eduardo Uriegas-Ibarra an Marco Aurelio Nuño-Maganda. *Simulación de brazo robótico en dispositivos móviles utilizando OpenGL ES*. Informe técnico proyecto de asignatura “Programación Móvil”. Universidad Politécnica de Victoria, 2022

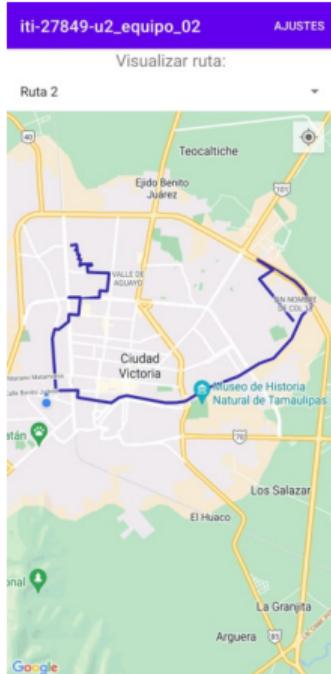
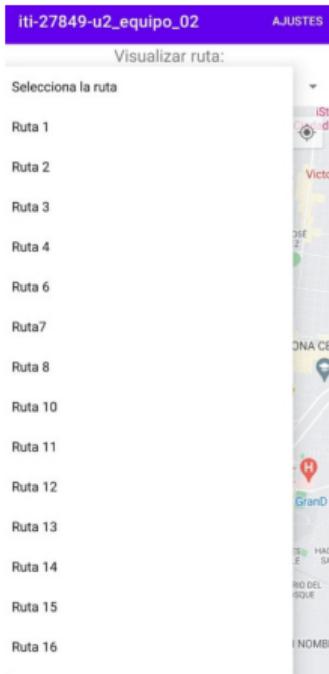
# *Aplicación móvil para consultar rutas de autobuses urbanos en Ciudad Victoria, Tamaulipas\* (1)*

- Una aplicación eficaz para uso de transporte público debe mostrar información acerca de las opciones de movilidad el usuario tiene en una localidad específica.
- En algunas localidades, esta información no está disponible en las aplicaciones de mapas, lo que complica la movilidad del usuario.
- Se implementó una App específicamente para Ciudad Victoria, Tamps

---

\* Dino Tonatiuh Gutiérrez-Martínez, Orlando Samuel Martínez-Dorantes, Juan José Torres-Grimaldo, Adriana Palmero-Torres, Ana Karen Echartea-Juárez, Angela Carrizales-Pérez, Osiel Gómez-Flores, Gabriela Robles-Rodríguez, and Marco Aurelio Nuño-Maganda. *Aplicación móvil para consultar rutas de autobuses urbanos en Ciudad Victoria, Tamaulipas*. *Research in Computing Science, Accepted*. Universidad Politécnica de Victoria, 2022.

# Aplicación móvil para consultar rutas de autobuses urbanos en Ciudad Victoria, Tamaulipas (2)

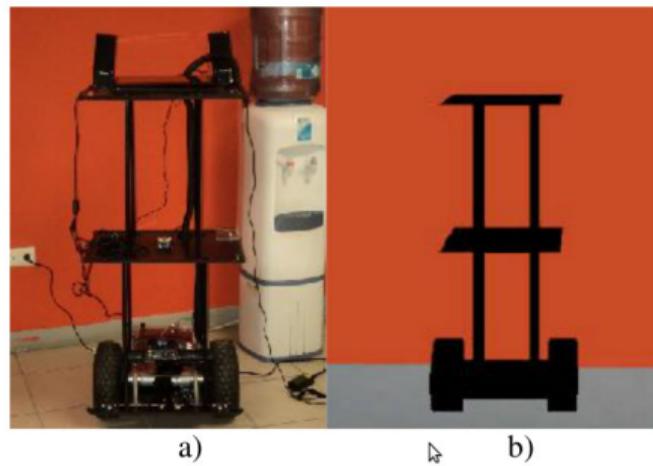


# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# “Implementación de un Ambiente de Simulación Virtual en 3D para el Robot de Servicio SerBot I” (1)

- Prototipo SerBot I
- Modelo 3D del Robot
- Modelo 3D del entorno
- Plataforma: PC de escritorio

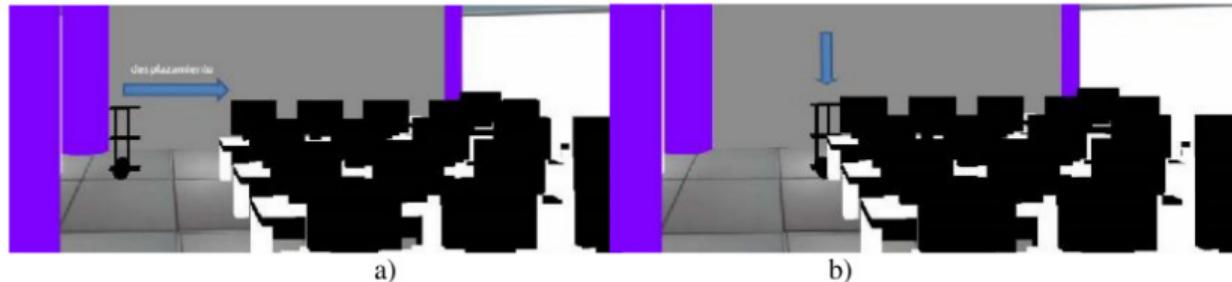
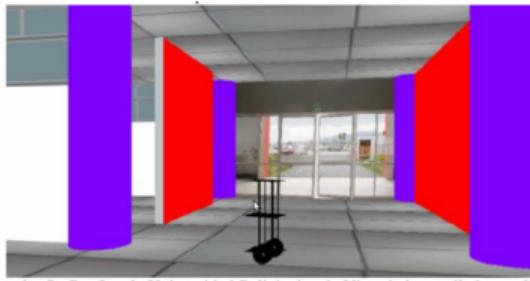


\* Gustavo Bujano-Guzmán, Erick Noel Pérez-Hernández, Marco Aurelio Nuño-Maganda, and Hector Hugo Aviles-Arriaga. “Implementación de un Ambiente de Simulación Virtual en 3D para el Robot de Servicio SerBot I”. In: *Congreso Internacional de Investigación 5.3* (Nov. 2013).

[ISSN:1945-5351](#), pp. 368–372

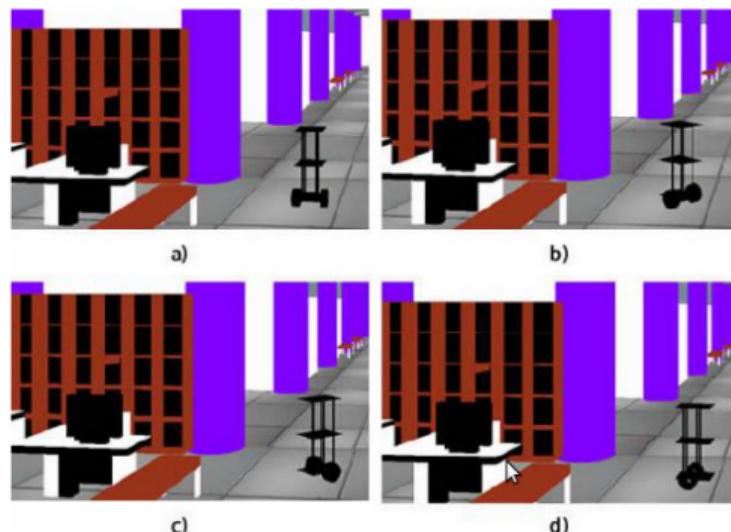
# “Implementación de un Ambiente de Simulación Virtual en 3D para el Robot de Servicio SerBot I” (2)

- A partir de los planos de los edificios y de las fotografías, se construyó un prototipo del entorno
- El plano permitía determinar en donde estaban ubicadas paredes y puertas



# “Implementación de un Ambiente de Simulación Virtual en 3D para el Robot de Servicio SerBot I” (3)

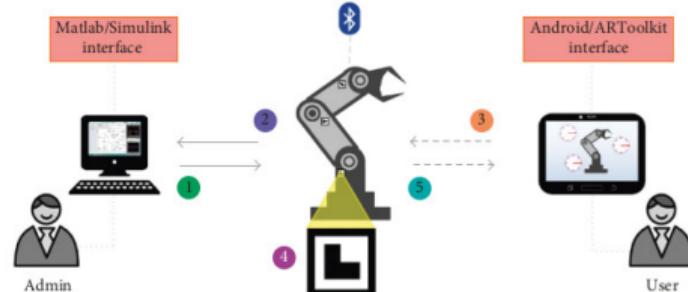
- Se implementaron controles para movimiento del Robot y del cambio de Vista
- El robot podía avanzar hacia adelante (moviendo las ruedas en la misma dirección) pero girar moviendo una rueda y dejando la otra quieta



# "An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality" (1)

## Componentes:

- Un sistema (Arduino) que genera los movimientos del brazo robot incluye un transmisor Bluetooth
- Una aplicación móvil que visualizar un transportador virtual encima de una articulación robótica con el ángulo en tiempo real

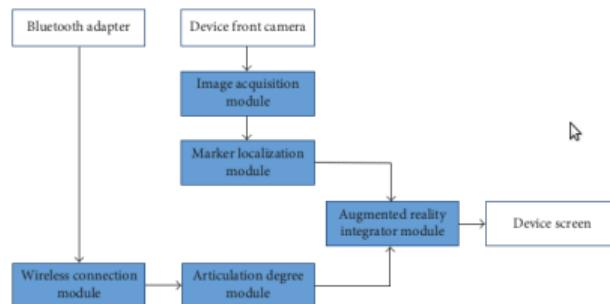


\* Martín Hernández-Ordoñez, Marco A. Nuño-Maganda, Carlos A. Calles-Arriaga, Omar Montaño-Rivas, and Karla E. Bautista Hernández. "An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality". In: *Mobile Information Systems* 2018 (Aug. 2018). <https://doi.org/10.1155/2018/6047034>, Article ID: 6047034, ISSN: 1875-905X

# "An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality" (2)

Funcionamiento de la aplicación:

- Se emplea un marcador ARUCO para determinar de que articulación se trata
- Mediante comandos Bluetooth se obtiene el ángulo



# Sistema de Realidad Aumentada

- Detección de Códigos QR
- Decodificación del texto codificado en el código QR
- Sobreposición del modelo 3D dependiendo de la posición del código QR



Chango

# Sistema de Realidad Aumentada (2)



Avion



Gato

# Desarrollos Tecnológicos

## Mascota de la UPV con Realidad Aumentada

- Detección de Códigos QR
- Modelado de la mascota en 3D mediante el Software Blender
- Sobreposición del modelo 3D dependiendo de la posición del código QR



\* Andrés García-González, Cristian Aléxis Lazo-García, and Damaris Mendoza-Vázquez. *Modelo 3D del Jaguar de la UPV sobre un código QR.* Informe técnico proyecto de asignatura “Graficación por Computadora Avanzada”. Universidad Politécnica de Victoria, 2020

# Desarrollos Tecnológicos

## Mascota de la UPV con Realidad Aumentada



# Recorrido UPV Virtual en teléfonos inteligentes

Demo incremental, que emplea OpenGL ES 2.0 (compatible con el 100% de los smartphones).

- Versión 1: Solo mundo virtual (no inmersivo). El usuario se movía con presionando teclas de la interfaz de usuario
- Versión 2: Mundo virtual inmersivo (integrado a unos lentes). El usuario movía la vista mediante los datos obtenidos por el sensor giroscópico del teléfono inteligente y avanzaba usando un manos libres alámbrico.
- Versión 3: Controlado por voz. El usuario se movía dentro del entorno mediante comandos de voz.
- Versión 3.5: Controlado mediante control de videojuegos (Bluetooth o USB)

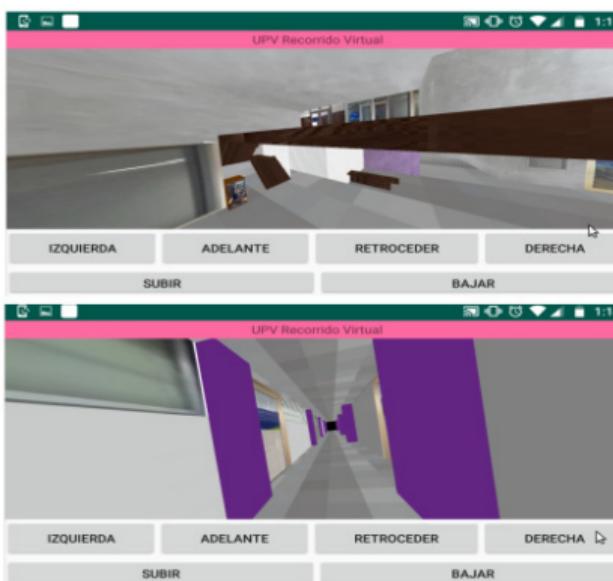
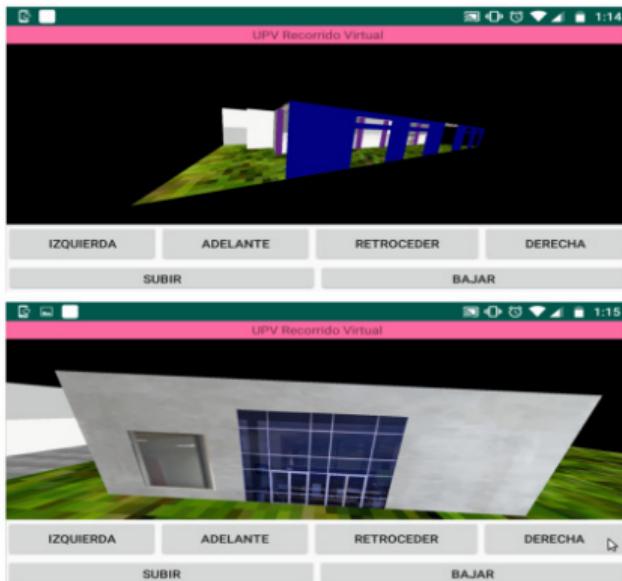
# Recorrido UPV Virtual 1.0

- La navegación es mediante botones (adelante, atrás, izquierda, derecha, subir, bajar)
- Una potencial mejora es mediante eventos de toque en pantalla



# Recorrido UPV Virtual 1.0 (2)

- Vistas de los diferentes edificios



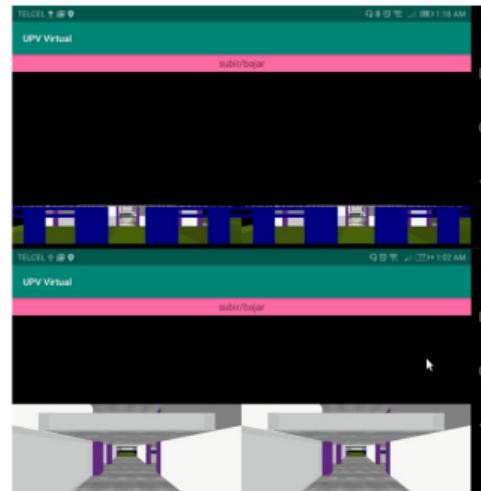
# Recorrido UPV Virtual 2.0

- Se extendió el demo 1.0 para generar una vista dual, requerida para su uso en conjunto con un armazón de VR.
- La vista cambia en base a lo obtenido por el giroscopio, y el movimiento se controla mediante el botón del manos libres.



# Recorrido UPV Virtual 3.0

- Se eliminó el uso del botón del manos libres para incluir comandos de voz
- La aplicación respondía a los comandos de voz, de tal forma que el usuario no debería mover nada



# Recorrido UPV Virtual 3.5

- Se incorporaron varios controles de consolas de videojuego para la navegación.



---

\* Mario Alberto Delarbre-Quintanilla, Gerardo Loperena-Bustillos, Alejandro Lumbreras-Olvera, Jorge Eduardo Monita-Rodríguez, Christian Alejandro Saldaña-Calderon, Juan Ramiro Soto-Gómez, Claudia Corina Vásquez-Pérez, Hector Yahude Vallejo-Cepeda, Andrea Medrano-Salvidar, Julio Alberto Zuniga-Gallegos, and Darianna Carreon-Gomez. *Incorporación de Controles de Consola de Video Juego al Recorrido Virtual de la UPV*. Informe técnico proyecto de asignatura “Cómputo en Dispositivos Móviles”. Universidad Politécnica de Victoria, 2023.

# *Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria\* (1)*

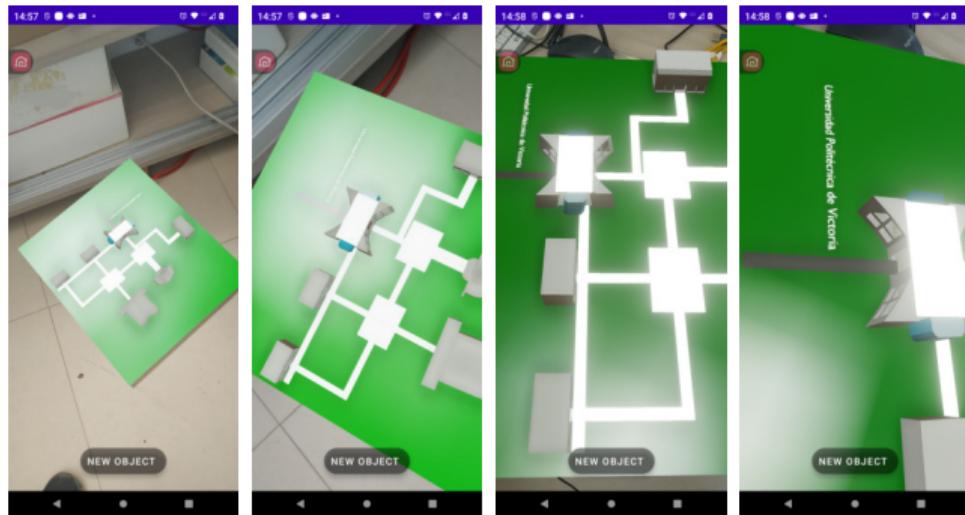
- Se propone una aplicación de RA que despliegue un MAPA de la UPV
- Se generó el modelo utilizando Blender y se exportó a un formato GLB
- La aplicación hace uso de la cámara para desplegar el modelo en una superficie con textura

---

\* Eymi Lizeth Andrade-Martinez, Edson Jair Hernández-Garza, Oscar Esteban Infante-González, José Luis Leal Requena, Pablo Nahum Ochoa-Morales, Jesús Guadalupe Rangel-Turrubiates, José Manuel Rodríguez-García, and Carlos Eduardo Sánchez-Charles. *Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria*. Informe técnico proyecto de asignatura “Cómputo en Dispositivos Móviles”. Universidad Politécnica de Victoria, 2022.

# Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria (2)

- Se propone una aplicación de RA que despliega un MAPA de la UPV



# Outline

- 1 Conceptos
- 2 Herramientas
- 3 Programación y Teléfonos Inteligentes
- 4 Herramientas Requeridas
- 5 Pasos para configurar Smartphone
- 6 Primer demo en Android Studio
- 7 Prototipos de Aplicaciones Graficas y de VC
- 8 Prototipos de Realidad Virtual y Aumentada
- 9 Conclusiones parte 1

# Conclusión parte 1

- Se presentó la relación entre Inteligencia Artificial, Visión por Computadora y los Gráficos por Computadora con la RA y la RV
- Se establecieron las herramientas Hardware y Software para Realizar proyectos Gráficos y de RA y RV para Teléfonos Inteligentes en Android Studio
- Se presentaron proyectos de Graficación y de RA y RV, principalmente desarrollados por estudiantes de la UPV (Ingeniería en Tecnologías de la Información y de Maestría en Ingeniería)
- Algunos trabajos establecen vínculos de colaboracion con otras universidades

# Artículos publicados I

- [2] Carlos Eduardo Garcia-Garcia, Jesus Antonio Luna-Alvarez, Hector Hugo Sandoval-Marcelo, and Jose Eduardo Torres-Montalvo. *Ajedrez Multijugador*. Informe técnico proyecto de asignatura "Graficación por Computadora Avanzada". Universidad Politécnica de Victoria, 2017.
- [3] Héctor Hugo Sandoval-Marcelo. *Movimiento de la Cruz de Malta con Iluminación y Texturas con Java y OpenGL ES 2.0 en Android*. Informe técnico proyecto de asignatura "Graficación por Computadora Avanzada". Universidad Politécnica de Victoria, 2017.
- [4] Víctor Manuel Gloria-Vázquez, Ana Karen Herrera Hernández, and Víctor Manuel Torres Moreno. *Tetris 3D en Android*. Proyecto Final de la Asignatura Graficación por Computadora Avanzada. Universidad Politécnica de Victoria, 2017.
- [5] Martín Hernández-Ordoñez, Marco Aurelio Nuño-Maganda, Carlos Adrián Calles-Arriaga, Abelardo Rodríguez-León, Guillermo Efren Ovando-Chacon, Rolando Salazar-Hernández, Omar Montaño-Rivas, and José Margarito Canseco-Cortinas. "Medical Assistant Mobile Application for Diabetes Control by Simulating a Compartmental Model". In: *Applied Sciences* 10.19 (Oct. 2020). <https://doi.org/10.3390/app10196846>. ISSN: 2076-3417. DOI: 10.3390/app10196846. URL: <https://www.mdpi.com/2076-3417/10/19/6846>.
- [6] Froylan Melquiades Wbario-Martinez. *Seguidor de movimientos de Ajedrez*. Informe técnico proyecto de asignatura "Visión por Computadora". Universidad Politécnica de Victoria, 2021.
- [7] Cristian Isidro Echartea-De-la-Rosa, Marco Aurelio Nuño-Maganda, and Yahir Hernández-Mier. *Mobile Application for Overweight Education*. Artículo en preparación. Universidad Politécnica de Victoria, 2021.
- [8] Axel Issai Alemán-Delgado, Isaac De Jesús Becerra-Martínez, Karen Lizeth García-Rivera, Daniela Estefanía Huerta-Vázquez, Didre Dayan Mota-González, Carlos Alejandro Reyes-Puga, and Marco Aurelio Nuño-Maganda. *Integración de un escáner 3D con sensores ultrasónicos y una aplicación móvil para visualización de modelos tridimensionales*. Informe técnico proyecto de asignatura "Programación Móvil". Universidad Politécnica de Victoria, 2022.
- [9] José Carlos Morin-Garcia, Froylán Melquiades Wbario-Martinez, and Mariela Georgina Reyes-Fonseca. *Implementación de un Modelo 3D de un Brazo Robótico en una Aplicación Android*. Informe técnico proyecto de asignatura "Graficación por Computadora Avanzada". Universidad Politécnica de Victoria, 2019.

# Artículos publicados II

- [10] Dino Tonatiuh Gutierrez-Martinez, Marly Yuleth Cantú-Charles, and Jesus Eduardo Uriegas-Ibarra an Marco Aurelio Nuño-Maganda. *Simulación de brazo robótico en dispositivos móviles utilizando OpenGL ES*. Informe técnico proyecto de asignatura "Programación Móvil". Universidad Politécnica de Victoria, 2022.
- [11] Dino Tonatiuh Gutiérrez-Martínez, Orlando Samuel Martínez-Dorantes, Juan José Torres-Grimaldo, Adriana Palmero-Torres, Ana Karen Echartea-Juárez, Angela Carrizales-Pérez, Osiel Gómez-Flores, Gabriela Robles-Rodríguez, and Marco Aurelio Nuño-Maganda. *Aplicación móvil para consultar rutas de autobuses urbanos en Ciudad Victoria, Tamaulipas*. *Research in Computing Science, Accepted*. Universidad Politécnica de Victoria, 2022.
- [12] Gustavo Bujano-Guzmán, Erick Noel Pérez-Hernández, **Marco Aurelio Nuño-Maganda**, and Hector Hugo Aviles-Arriaga. "Implementación de un Ambiente de Simulación Virtual en 3D para el Robot de Servicio SerBot I". In: *Congreso Internacional de Investigación 5.3* (Nov. 2013). [ISSN:1945-5351](#), pp. 368–372.
- [13] Martín Hernández-Ordoñez, Marco A. Nuño-Maganda, Carlos A. Calles-Arriaga, Omar Montaño-Rivas, and Karla E. Bautista Hernández. "An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality". In: *Mobile Information Systems 2018* (Aug. 2018). <https://doi.org/10.1155/2018/6047034>, Article ID: [6047034](#), [ISSN: 1875-905X](#).
- [14] Jesús Alfredo Cárdenas-Castillo, Eluis Carlo Ramos-Lucio Ana Karen Molina-Pastrana and, Linda Margarita Rodríguez-Terán, and César Jovany Vázquez-Luna. *Visión por Computadora para Realidad Aumentada*. Proyecto Final de la Asignatura Cómputo en Dispositivos Móviles. Universidad Politécnica de Victoria, 2019.
- [15] Andrés García-González, Cristian Aléxis Lazo-García, and Damaris Mendoza-Vázquez. *Modelo 3D del Jaguar de la UPV sobre un código QR*. Informe técnico proyecto de asignatura "Graficación por Computadora Avanzada". Universidad Politécnica de Victoria, 2020.
- [16] Maria Fernanda Baez-Zapata, Jesús Alfredo Cárdenas-Castillo, José Armando Olvera-Osuna, and Yu-Hsiang Wang. *Recorrido UPV en Android*. Proyecto Final de la Asignatura Cómputo en Dispositivos Móviles. Universidad Politécnica de Victoria, 2019.
- [17] Carlos Alberto Alarcon-Longoria, Leonardo Daniel Alonso-Cepeda, and Luis Angel Torres-Grimaldo. *Aplicación UPV-Virtual*. Proyecto Final de la Asignatura Cómputo en Dispositivos Móviles. Universidad Politécnica de Victoria, 2019.

# Artículos publicados III

- [18] José Treviño-Olvera. *Recorrido UPV Virtual en Android con Controles de Voz*. Proyecto Final de la Asignatura Cómputo en Dispositivos Móviles. Universidad Politécnica de Victoria, 2020.
- [19] Mario Alberto Delarbrea-Quintanilla, Gerardo Loperena-Bustillos, Alejandro Lumbreras-Olvera, Jorge Eduardo Monita-Rodríguez, Christian Alejandro Saldaña-Calderon, Juan Ramiro Soto-Gómez, Claudia Corina Vásquez-Pérez, Hector Yahude Vallejo-Cepeda, Andrea Medrano-Salvidar, Julio Alberto Zuniga-Gallegos, and Darianna Carreon-Gomez. *Incorporación de Controles de Consola de Video Juego al Recorrido Virtual de la UPV*. Informe técnico proyecto de asignatura "Cómputo en Dispositivos Móviles". Universidad Politécnica de Victoria, 2021.
- [20] Eymí Lizeth Andrade-Martinez, Edson Jair Hernández-Garza, Oscar Esteban Infante-González, José Luis Leal Requena, Pablo Nahúm Ochoa-Morales, Jesús Guadalupe Rangel-Turrubiates, José Manuel Rodríguez-García, and Carlos Eduardo Sánchez-Charles. *Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria*. Informe técnico proyecto de asignatura "Cómputo en Dispositivos Móviles". Universidad Politécnica de Victoria, 2022.