

TRABAJO PROFESIONAL FIUBAAR





Esteban
Guillardoy

Padrón: 81597



Ezequiel
Aguilar

Padrón: 73935

1

INTRODUCCIÓN

Que es FIUBAAR?



Sistema de Realidad Aumentada

- Plataformas móviles Android
- Modelos 3D obtenidos de forma dinámica
- Reconocimiento y seguimiento de manos
- Interacción mediante gestos frente a la cámara



MOTIVACIÓN

PERSONAL

- Plataformas móviles
- Sistemas gráficos
- Procesamiento de imágenes
- Diferente a nuestro ámbito de trabajo

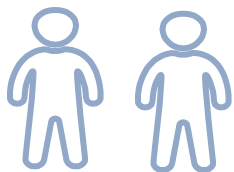
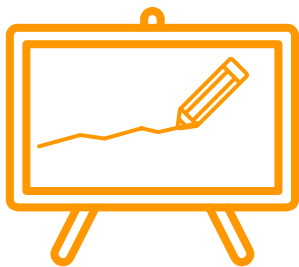
DIFERENCIACIÓN

- Contar con modelos 3D no embebidos en la aplicación
- Interactuar sin tocar el dispositivo
- Acercar la experiencia de Realidad Aumentada a Realidad Virtual

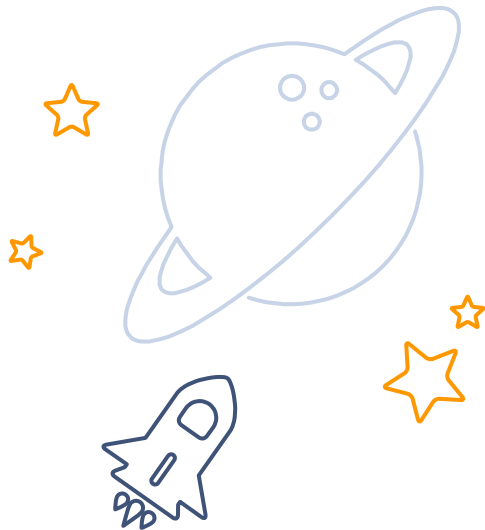


USOS Y APLICACIONES

DIDÁCTICA



ENTRETENIMIENTO



MARKETING





COMPONENTES

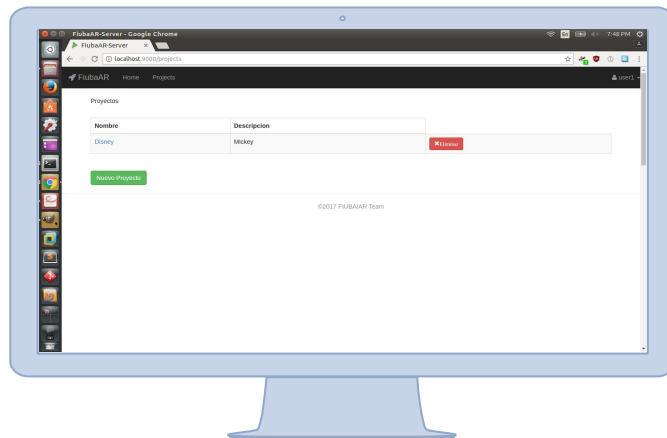
CLIENTE

Aplicación Android



SERVIDOR

Aplicación Web Java



2

DESAFÍOS

Investigación e Implementación



Diseño y desarrollo para plataforma Android



Planificación de tareas



Estimación de tiempos



Disponibilidad del equipo



Estado del arte de la Realidad Aumentada



Procesamiento de imágenes



Motores graficos 3D



Marcadores dinámicos



Detección de manos y dedos



Reconocimiento de gestos



Performance



Librerías existentes

3

IMPLEMENTACIÓN

Componentes y Arquitectura



TECNOLOGÍAS



FIUBAAR



ANDROID



Qualcomm
vuforia



OpenCV



Rajawali

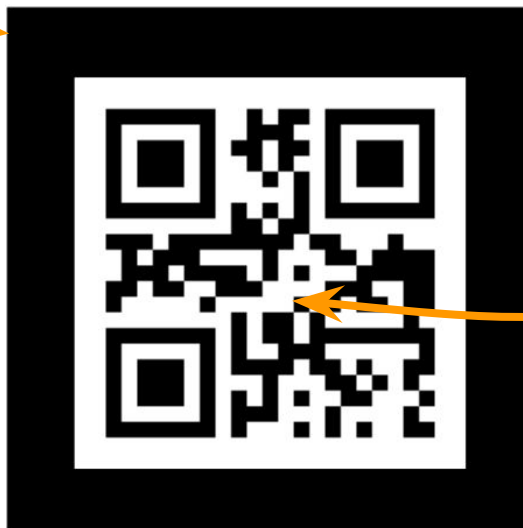


Apache Commons™



CONTENEDOR

Patrón fijo fácilmente reconocible que permite realizar seguimiento

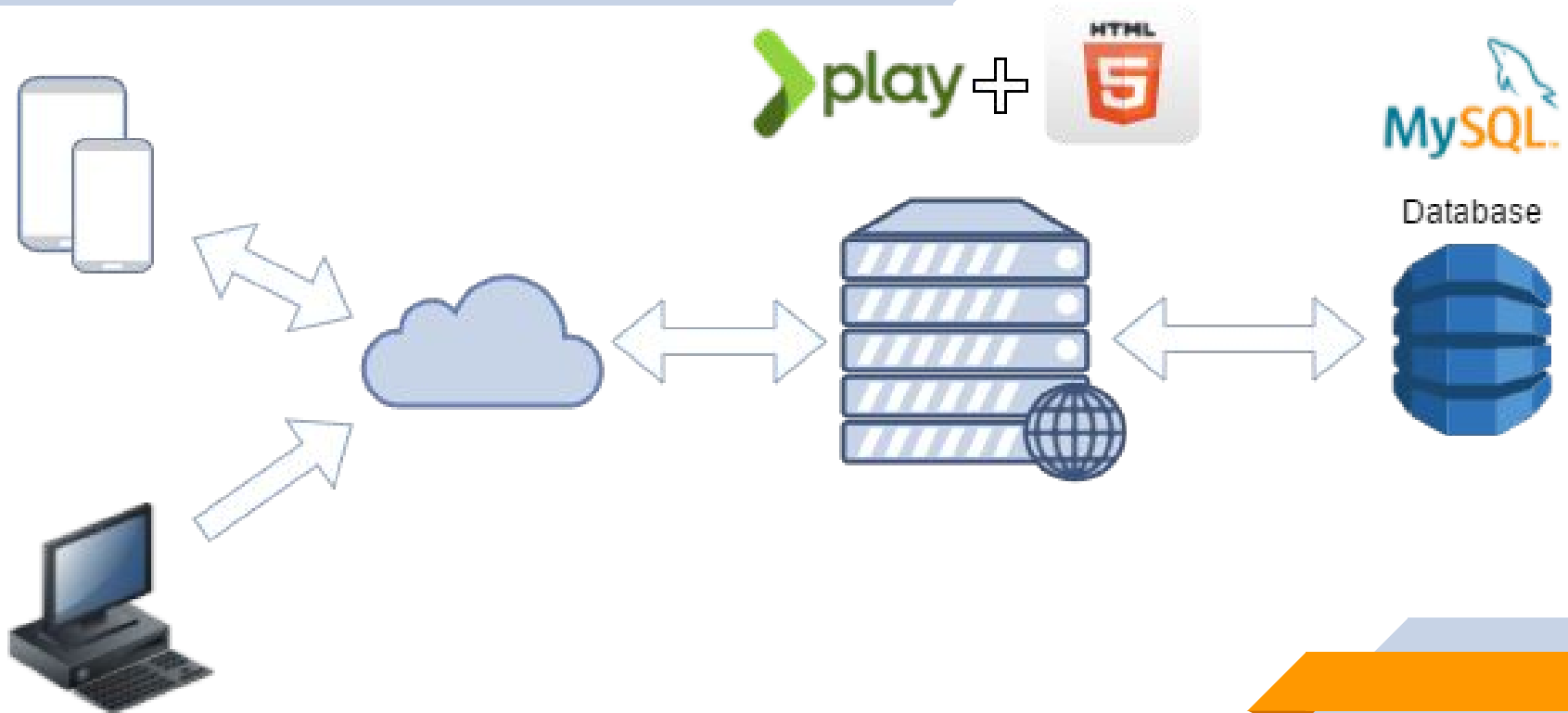


NÚCLEO

Código QR que contiene información necesaria para la obtención dinámica del modelo 3D asociado



ARQUITECTURA SERVIDOR





EVOLUCIÓN ARQUITECTURA CLIENTE

1

- OpenCV
- Aruco (C/C++)
- ZBar (C/C++)
- Detección y seguimiento de marcadores
- Lectura de códigos QR
- Problemas de memoria

2

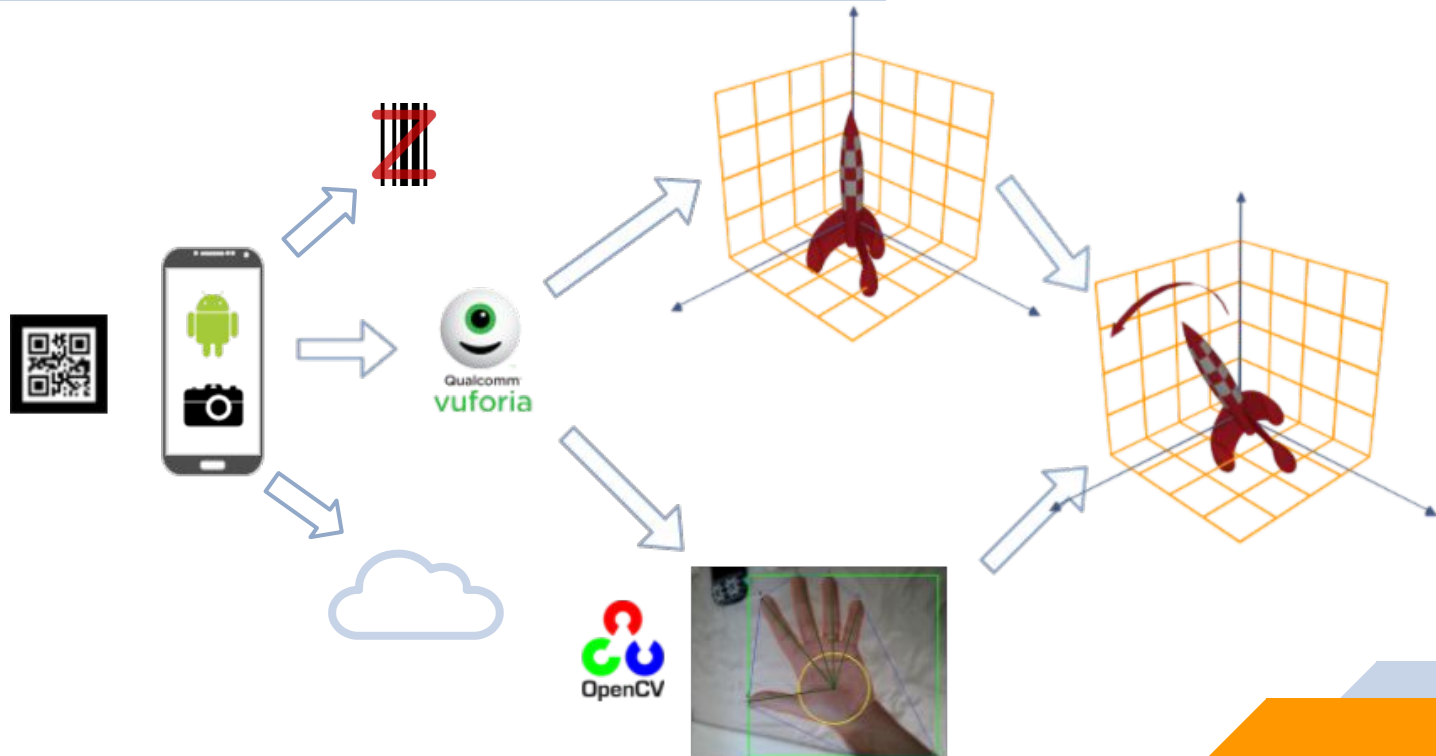
- Eliminación de Aruco
- Reemplazo con Vuforia SDK
- Solución a problemas de memoria
- Modificación de ZBar (Java)
- Motor gráfico Rajawali
- Posicionamiento de modelos 3D sobre marcadores

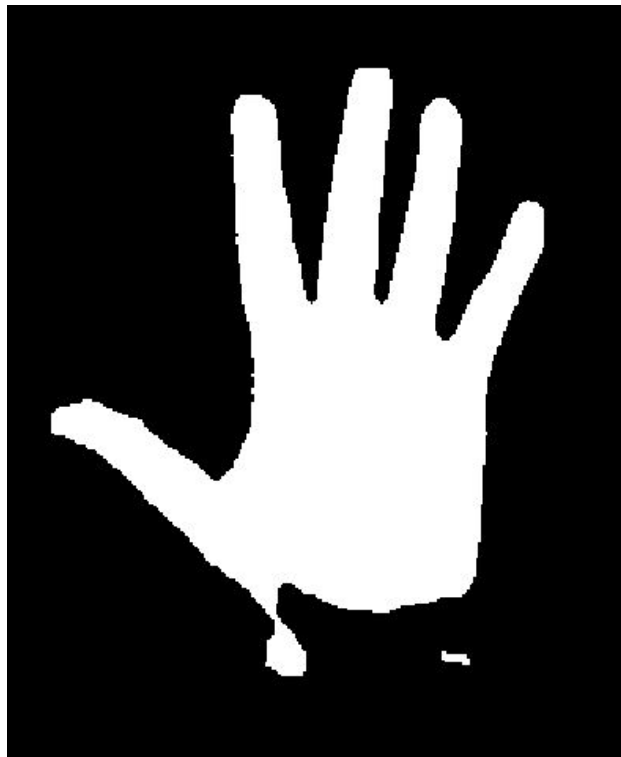
3

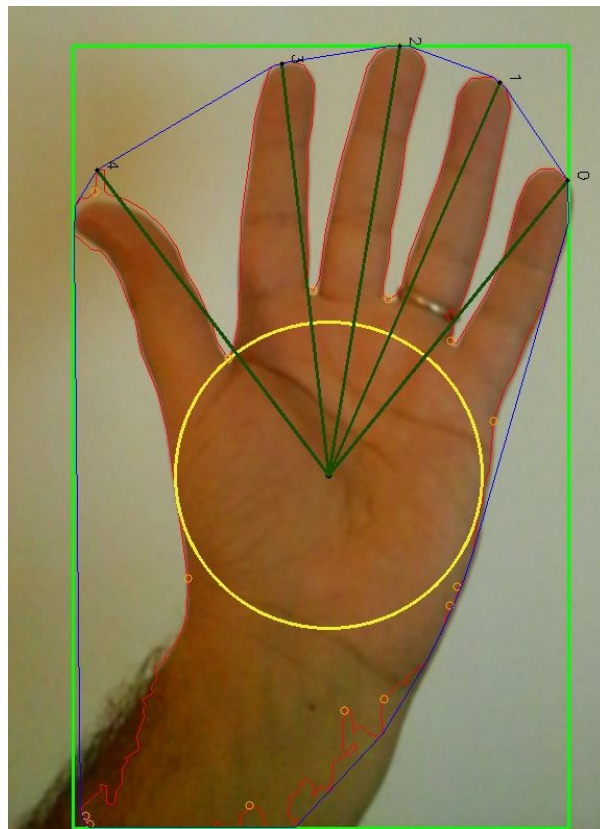
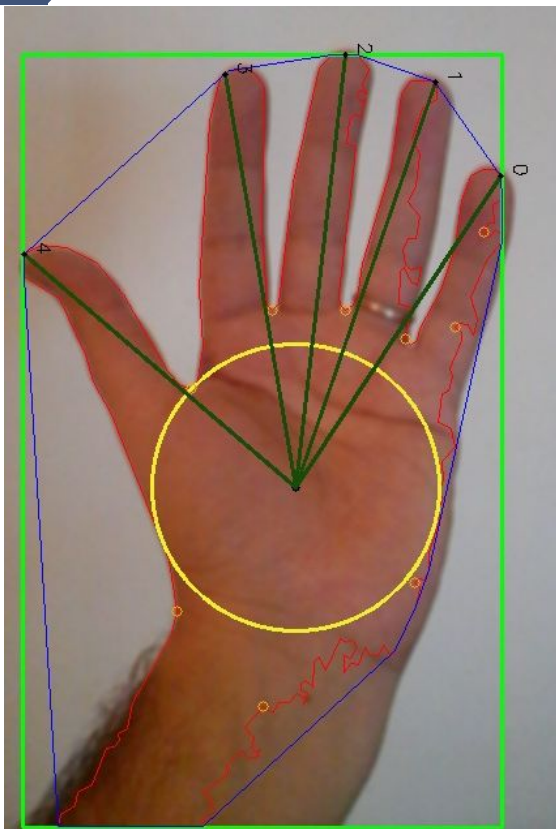
- Detección y seguimiento de manos (Java y C++ en simultáneo)
- Servicios de fondo
- Varias alternativas de programación concurrente implementadas
- Problemas de performance
- Reestructuración de paquetes y clases



ARQUITECTURA CLIENTE FINAL









DEMO

4

CONCLUSIONES

Limitaciones, mejoras y lecciones aprendidas



LIMITACIONES

- Unico objeto AR activo por vez
- Soporte de formatos en modelos 3D
- Performance en la detección y seguimiento de manos
- Gestos reconocidos e interpretados

A pesar de esto la implementación final representa un
prototipo funcional completo



MEJORAS Y ALTERNATIVAS

- Interfaz y experiencia de usuario
- Integración de webcam externa via USB OTG
- Utilización de hardware dedicado
- Posible uso de programación reactiva
- Reemplazo del algoritmo de detección de manos



LECCIONES APRENDIDAS

- Diseño y desarrollo en plataformas móviles
- Motores gráficos y modelado 3D
- Algoritmos de visión por computadora
- Estimación y manejo de proyectos



MUCHAS GRACIAS

¿Preguntas?