

Desarrollo de Markless en un generador de aplicaciones de Realidad Aumentada (ARAPP Builder)

Sergio Antonio Cornu Tinoco.^a, López Martínez Ignacio.^b,
Reyes Hernández Luis Angel.^c, Olivares Zepahua Beatriz A.^d,
Muñoz Contreras Hilarion.^e

^a Instituto Tecnológico de Orizaba División de Estudios de Posgrado e Investigación, Oriente 9 colonia Emiliano Zapata código postal 94320, scornu.tinoco@gmail.com, Orizaba, Veracruz México

^b Instituto Tecnológico de Orizaba División de Estudios de Posgrado e Investigación, Oriente 9 colonia Emiliano Zapata código postal 94320, ilopezm@orizaba.tecnm.mx, Orizaba, Veracruz México

^c Investigación, Oriente 9 colonia Emiliano Zapata código postal 94320, l.a.reyes.h@gmail.com, Orizaba, Veracruz México

^d Instituto Tecnológico de Orizaba División de Estudios de Posgrado e Investigación, Oriente 9 colonia Emiliano Zapata código postal 94320, bolivares@ito-depi.edu.mx, Orizaba, Veracruz México

^e Instituto Tecnológico de Orizaba División de Estudios de Posgrado e Investigación, Oriente 9 colonia Emiliano Zapata código postal 94320, hmunozc189@msn.com, Orizaba, Veracruz México.

Resumen

Este trabajo muestra la ampliación de capacidades de ARAPP Builder (Sistema de Autor de Aplicaciones de Realidad Aumentada [1]) para integrar la tecnología markless, aplicando un proceso de reingeniería al módulo de captura, dándole nuevas opciones a los usuarios al momento de crear sus propias aplicaciones. La nueva tecnología integrada proveerá el uso de imágenes sin necesidad que sean marcadores específicos.

Como parte del desarrollo se analizó el uso de la API de realidad aumentada (RA) ARToolKitX y el uso de markless, así también se revisó a fondo el api que ofrece Google que es ARCore y las posibilidades que tiene para el desarrollo del sistema.

Se presenta la problemática de uso de la API ARToolKitX y su incompatibilidad con versiones anteriores de la misma.

Palabras clave— Realidad Aumentada, Sistema de Autor, Markless.

Abstract

This work shows the expansion of capabilities of ARAPP Builder (Augmented Reality Applications Author System [1]) to integrate markless technology, applying a reengineering process to the capture module, giving users new options when creating their own applications. The new integrated technology will provide the use of images without needing to be specific markers.

As part of the development, the use of the ARToolKitX Augmented Reality API (AR) and the use of markless were analyzed, as well as the api offered by Google that is ARCore and the possibilities it has for the development of the system.

The problem of using the ARToolKitX API and its incompatibility with previous versions of it is presented.

Keywords— Augmented Reality, Author System, Markless.

1. INTRODUCCIÓN

El presente trabajo propone ampliar las capacidades de ARAPP Builder [1] para crear aplicaciones de RA, permitiendo así la creación de aplicaciones que hagan uso de nuevas tecnologías y técnicas para la RA, como son Markless, NFC, GPS; centrándose por el momento en la primera. En años recientes las aplicaciones de RA han presentado un auge en su desarrollo y en el uso por los usuarios de dispositivos móviles así [2] así como también en ciertas áreas de la industria. Estas aplicaciones han sido desarrolladas por grandes empresas y grupos de desarrolladores dejando a un lado al usuario promedio para la creación de estas.

Entre las mejoras propuestas para ampliación de las capacidades de ARAPP Builder se presenta markless (sin marcadores, por significado en inglés) que es la extracción de características naturales de una imagen [3] ya que dicha tecnología ha presentado un gran uso en la RA ampliando así la aceptación y uso entre los usuarios.

La principal ventaja del uso de la RA con markless es que puede utilizarse con objeto físicos e imágenes del mundo real, sin la necesidad de que estos tengan algún identificador (marcador, imágenes diseñadas para la aplicación que desencadenan una acción) permitiendo así sobreponer los elementos virtuales sobre él, entre las desventajas que presenta el uso de markless es el elevado uso del procesador necesario para hacerla funcionar.

El desarrollo de la nueva característica de ARAPP Builder se realizó con la finalidad de poder utilizar markless dentro de las aplicaciones generadas, ampliando así las opciones que el usuario tiene al momento de estar realizando sus aplicaciones, debido a que hasta el momento solo se ha podido utilizar marcadores tradicionales en 2D en alto contraste o bicolor, es decir, patrones o marcos que se podían identificar dentro de la imagen, pero no la imagen completa determinada para realizar el aumento.

El interés de esta investigación es verificar la capacidad de la API (Application Programming Interface en inglés) utilizada como motor de RA para aceptar la nueva tecnología propuesta.

2. Trabajos Relacionados

López, D et al. [1] describe la tecnología de RA y sobre cómo esta difiere de la realidad virtual, se aborda la problemática principal que da inicio a la creación de ARAPP Builder, dentro del artículo se describen las diferentes tecnologías utilizadas para la creación de dicho sistema y las características que el sistema tiene para la creación de las

aplicaciones para dispositivo móvil. El trabajo concluye con la presentación de una beta funcional del sistema creando aplicaciones de RA para ejecutar en diferentes dispositivos móviles.

En [4] E. Duque-bedoya describe los tipos de RA que existen que se dividen en dos exponentes los basados en imágenes siendo el medio más utilizado para la realización de la RA; el segundo exponente de la RA que menciona es la que hace uso de la localización del usuario para mostrar en pantalla información referente al lugar donde se encuentra, dentro del mismo trabajo se aborda el uso de la RA como una herramienta educativa permitiendo así que los docentes hagan uso de esta tecnología en el aula de clases.

A. Shatte et al. abordan en [5] como las operaciones de manejo de una biblioteca son tediosas y propensas a los errores debido a la limitante que tienen las personas para manejar varios objetos al mismo tiempo, para dar solución a esta problemática se propone el uso de una aplicación de RA, de esta aplicación se realizó un prototipo que se implementó en una biblioteca donde se observaron resultados satisfactorios en comparación a los métodos de manejo tradicionales de la biblioteca.

A. Rese at al. hace una recopilación en [6] sobre cómo las aplicaciones de RA son aceptadas por los usuarios en general y sobre los diferentes estudios que se realizan para evaluar la usabilidad de estas, en este trabajo se hace énfasis principalmente en la técnica TAM (Technology Acceptance Model) la cual se utiliza para medir la aceptación de las aplicaciones de RA entre los usuarios.

J. Paulo Lima et al, en [7] menciona las problemáticas que se presentan al utilizar aplicaciones de RA en el sector automovilístico debido a que en los automóviles no es posible o conveniente poner un marcador a la vista, para dar solución a esta problemática se hace uso de la tecnología markless; ya con esta tecnología implementada se realizó una aplicación que se probó en el Ikswagen/ISMAR Tracking Challenge 2014.

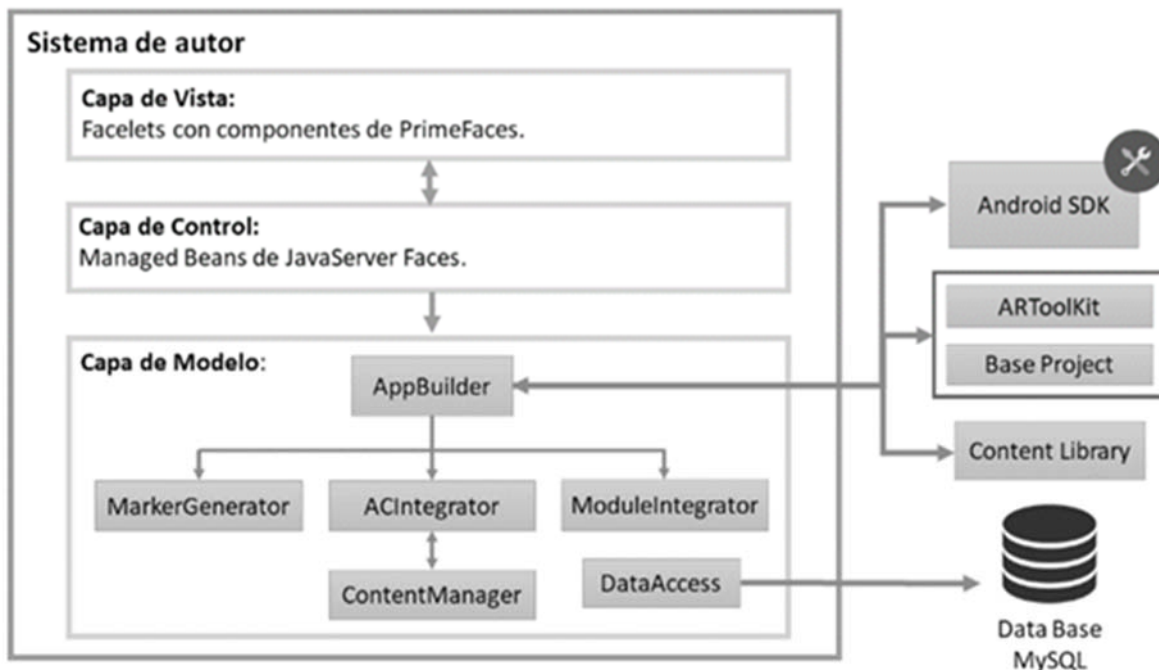
W. Fang et. al. describe en [8] como la RAU (realidad aumentada usable, wearable augmented reality, WAR en inglés) se utiliza para visualizar elementos virtuales en tiempo real en escenarios reales por medio de dispositivos móviles. Las RAU proveen una interfaz humano-computadora al permitir la percepción del mundo real con elementos externos, en este trabajo se propone el uso de una cámara monocular como sensor de markless y un sensor de inercia.

3. Desarrollo

La solución para la problemática presentada es aplicar un proceso de reingeniería a la aplicación ARAPP Builder, reescribiendo solo el módulo con la función de la detección de marcadores para que este haga uso de markless, permitiendo la utilización de imágenes como si fueran marcadores.

Para realizar dicha modificación al módulo se realizó una modificación a la arquitectura general de la aplicación original (figura 1).

Figura 1. Arquitectura original de ARAPP Builder.

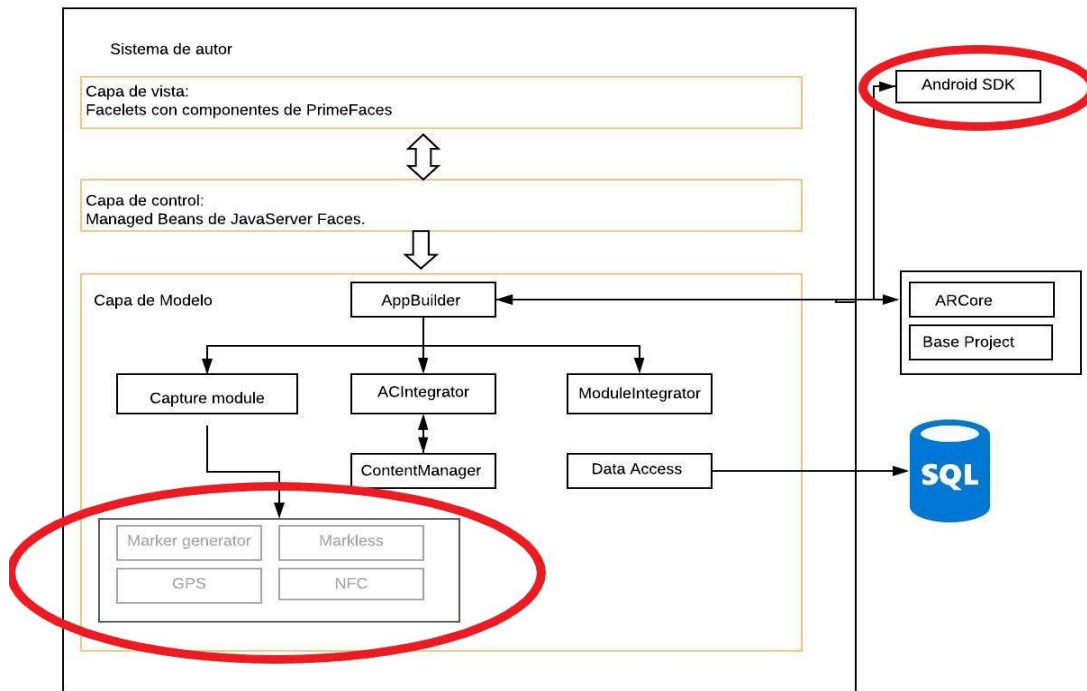


Fuente: Arquitectura y procesos para el desarrollo de un Sistema de Autor de aplicaciones móviles con Realidad Aumentada [1].

La modificación propuesta a la arquitectura se centró en el módulo de MarkerGenerator dividiéndolo en cuatro submódulos cada uno encargado de una tecnología diferente (figura 2).

tratarlas como marcadores todo esto mediante el código mostrado en la figura 3.

Figura 2. Arquitectura de ARAPP Builder modificada.



Fuente: Autoria propia.

ARAPP Builder está realizado con la API de RA ARToolKit 5.4, la cual empleaba funciones y características que no son reconocidas por ARToolKitX [9] (versión más actual de la API), esta incompatibilidad entre las versiones llevó a realizar pruebas para verificar que la característica de uso de markless fuera posible utilizando esta nueva versión.

Como parte del trabajo realizado para lograr la modificación de ARAPP Builder se realizaron pruebas con la API ARToolKitX en su versión para Android 1.0.5, con las pruebas realizadas se determinó que ARToolKitX no es la API adecuada para la modificación propuesta.

Para continuar con la modificación del sistema se realizaron pruebas con diferentes APIs de RA, determinando que la más adecuada para realizar la modificación es ARCore de Google [10], esta API ha demostrado ser la más adecuada para cumplir con el objetivo, las pruebas realizadas con ARCore han demostrado ser más que satisfactorias al momento de detectar imágenes y usarlas con markless.

Una vez realizada la modificación a la arquitectura de la aplicación se procede a trabajar en cada módulo de manera específica, en este caso solo se abordará el desarrollo dentro del módulo de markless, lo cual ha permitido crear una clase JAVA que se encarga del manejo de las imágenes permitiendo

Parte de las modificaciones realizadas a ARAPP Builder fue un cambio al modo en que las aplicaciones eran generadas, el proceso de generación de las aplicaciones se llevaba a cabo utilizando la consola de Windows invocandola, utilizando la instrucción “executeCoomand” de JAVA esta instrucción debido a la gran cantidad de parámetros que necesita es susceptible a fallos y también puede llegar a fallar si en el sistema donde se ejecuta falta algún componente del class path de Windows, el código sin modificar se muestra en la figura 4.

La modificación se realizó cambiando el modo de invocación de la consola de Windows a una manera menos susceptible a fallos, esto se logró utilizando la instrucción “ProcessBuilder” y solo hace uso del archivo por lotes Gradlew (gradlew.bat), que es una herramienta de compilación de código abierto [11] que se genera al momento de construir el código de las aplicaciones, dicho código se muestra en la figura 5.

Figura 3. Seudocódigo generado utilizando ARCore.

```

Algoritmo onUpdateFrame
//variables a utilizar en el metodo
    Frame frame
    Collection updateAugmentedImages
    Node node
    ArFragment arFragment

frame = obtenerFrame()

si frame = null
    regresar
si no
    updateAugmentedImages =
frame.obtenerImagenesAumentadas()

Para updateAugmentedImages tenga imagen
    Selecciona en caso de imagen.obtenerEstado()
    Caso Paused: ...
    Caso Stopped: ...
    Caso Tracking:
        Si augmentedImagesmap no contiene imagen
            En caso de imagen.obtenerIndex
            0:
                node = nuevo nodo(contexto) // se crea un nuevo
nodo
                node.setImage(imagen) // se le pasa al nodo la
imagen
arFragment.obtenerNodo().obtenerEscena().añadirNiño(nodo) // se
añade a la escena el nodo creado
1: ...
    Fin caso
    Fin para
    Fin si
Fin algoritmo
    
```

Fuente: Autoria propia.

Figura 4. Seudocódigo para generar apk sin modificar.

```

Algoritmo generación de APK

ejecutarComando(invocación consola de Windows + ruta del proyecto
+ invocación de JAVA + configuración de JAVA + configuración
Gradlew + Instrucción de generación de Gradlew )
    
```

4 Caso de Estudio

El caso de estudio seleccionado para este trabajo consistió en probar el recién modificado módulo de markless de la aplicación haciendo uso de un número definido de imágenes con características diversas, dichas imágenes fueron procesadas por la herramienta “arcoring.exe” que provee ARCore para determinar si eran óptimas para utilizar, la herramienta arroja un resultado por imagen que está en el rango de 0 a 100; el puntaje recomendado por imagen para ser

aceptado como un buen marcador es de 75 como se muestra en la figura 6.

Figura 5. Seudocódigo código para la generación de apk.

```

Algoritmo para generación de APK
//variables del método
    String rutaApp
    ProcessBuilder builder
    Procesos p

Builder = nuevo builder(instrucción consola Windows, bandera
consola Windows, rutaAPP) // se crea un nuevo builder
Builder.redirectErrorStream(verdadero) // se redireccionan los
errores producidos por builder para concatenarlos con la
salida estándar
P = builder.start() // se da inicio al proceso y se guarda en una
variable tipo Process
    
```

Fuente: Autoria propia.

Con los resultados del análisis de las imágenes se procedió a integrar estas dentro de una aplicación de prueba para determinar la funcionalidad del módulo de markless desarrollado. Una vez integradas a la aplicación de prueba se continuó con la prueba dando como resultado la muestra de elementos aumentados en la pantalla del dispositivo como se muestra en la figura 7.

4. Conclusiones

El desarrollo de las aplicaciones de RA ha tenido una gran aceptación en tiempos recientes gracias a la popularidad creciente de estas, gracias a ello y a lo analizado en los trabajos relacionados se ha podido concluir el estado actual de las aplicaciones de RA y sobre cómo están integradas en diferentes áreas de la industria y de la vida diaria de las personas.

Los resultados expuestos en la sección anterior han demostrado ser satisfactorios sobre la integración de markless dentro de una aplicación de RA, ya con este resultado se procede a la integración del módulo de markless a ARAPP Builder para así ampliar la funcionalidad al momento de generar aplicaciones de RA.

5. Trabajo a Futuro

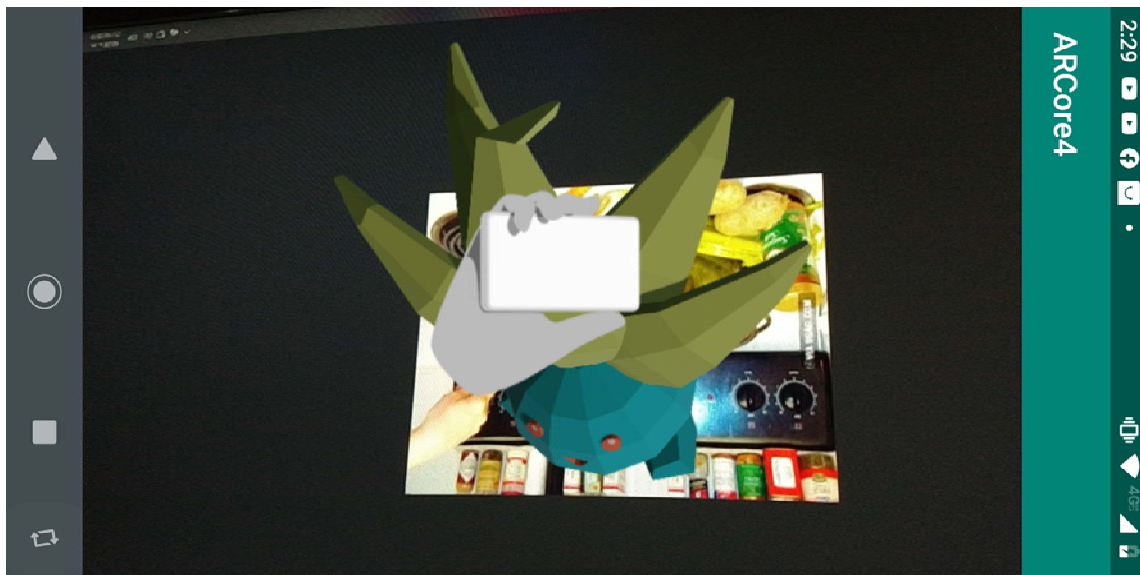
Como parte del trabajo a futuro se planea realizar el desarrollo de los módulos correspondientes a la tecnología de NFC y GPS, así como las pruebas necesarias para asegurar el correcto funcionamiento e integración de dichos módulos a la aplicación ARAPP Builder.

Figura 6. Resultado de arcoring.exe al analizar imágenes.

```
Simbolo del sistema
C:\Users\scorn\Pictures>arcoring eval-img --input_image_path=pug.jpg
100
C:\Users\scorn\Pictures>arcoring eval-img --input_image_path=test1.png
0
C:\Users\scorn\Pictures>arcoring eval-img --input_image_path=test2.jpg
85
C:\Users\scorn\Pictures>arcoring eval-img --input_image_path=test3.jpg
0
```

Fuente: Autoria propia.

Figura 7 resultado de la integración de markless.



Fuente: Autoria propia.

6. Agradecimientos

Los autores quisieran agradecer al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo económico para la realización de este.

4. REFERENCIAS

- [1] D. López, I. López, L. A. Reyes, C. Romero, and B. A. Olivares, "Arquitectura y procesos para el desarrollo de un Sistema de Autor de aplicaciones móviles con Realidad Aumentada," 2018.
- [2] A. Pyae and L. E. Potter, "A player engagement model for an augmented reality game: A case of Pokémon Go," *Proc. 28th Aust. Comput. Interact. Conf. OzCHI 2016*, pp. 11–15, 2016.
- [3] N. Gupta and M. K. Rohil, "An experimental study of markerless image registration methods on varying quality of images for augmented reality applications," in *Thematic Workshops 2017 - Proceedings of the Thematic Workshops of ACM Multimedia 2017, co-located with MM 2017*, 2017, pp. 502–510.
- [4] E. Duque-bedoya, "Usando realidad aumentada para motivar las competencias informacionales : experiencias en clase .," *XV Encuentro Int. Virtual Educ.*, pp. 1–11, 2014.
- [5] A. Shatte, J. Holdsworth, and I. Lee, "Mobile augmented reality based context-aware library management system," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2174–2185, 2014.
- [6] A. Rese, D. Baier, A. Geyer-Schulz, and S.

- Schreiber, "How augmented reality apps are accepted by consumers: A comparative analysis using scales and opinions," *Technol. Forecast. Soc. Change*, vol. 124, pp. 306–319, 2017.
- [7] J. Paulo Lima *et al.*, "Markerless tracking system for augmented reality in the automotive industry," *Expert Syst. Appl.*, vol. 82, pp. 100–114, 2017.
- [8] W. Fang, L. Zheng, and X. Wu, "Multi-sensor based real-time 6-DoF pose tracking for wearable augmented reality," *Comput. Ind.*, vol. 92–93, pp. 91–103, 2017.
- [9] artoolkitX, "Introduction to artoolkitX." [En línea]. Disponible:
<https://github.com/artoolkitx/artoolkitx/wiki/Introduction-to-artoolkitX>.
- [10] Google, "Choose Your Development Environment | ARCore | Google Developers," *ARcore Documentation*, 2017. [En línea]. Disponible:
<https://developers.google.com/ar/develop/>.
- [11] Google, "Cómo compilar tu app desde la línea de comandos." [En línea]. Available:
<https://developer.android.com/studio/build/building-cmdline?hl=es-419>.