



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño y Desarrollo de aplicaciones para
visualización de modelos 3D creados con
fotogrametría en Realidad Aumentada como
apoyo a la docencia en Geología**

MATERIAL DIDÁCTICO

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Diego Iñaki Garciarebollo Rojas

ASESOR DE MATERIAL DIDÁCTICO

Ing. Luis Sergio Valencia Castro



Ciudad Universitaria, Cd. Mx., 2021

INDICE

Contenido

Agradecimientos	2
Introducción	3
El estado del arte de la industria RA	4
Preproducción	7
Desarrollo de la aplicación	12
Modelado 3D	13
Sistema de Reconocimiento de imágenes	15
Sistema de descarga de contenido	18
Muestra del modelo	21
Sistema de Actualización	22
Pruebas de Implementación	29
Escena AR	29
Menú Colección	30
Menú de descargas	31
Conclusiones	33
Referencias	35

Tabla de ilustraciones

Ilustración 1	Captura del video demostrativo de la app Dinosaur 4D+	4
Ilustración 2	Captura de pantalla de la aplicación Element 4D	5
Ilustración 3	Diagrama de flujo instanciación modelo 3D.....	9
Ilustración 4	Diagrama agregar imagenes a la biblioteca de imágenes	10
Ilustración 5	Presentación ARGEO	12
Ilustración 6	Fotogrametría de una casa	13
Ilustración 7	Ejemplo Image Library	15
Ilustración 8	Demo disparador afloramiento.....	16
Ilustración 9	Componente AR Tracked Image Manager	16
Ilustración 10	Descarga de contenido	17
Ilustración 11	Ventana AssetBundle Browser	18
Ilustración 12	Ventana AssetBundle Browser, sección Build	19
Ilustración 13	Archivos AssetBundle Generados	19
Ilustración 14	Descarga en menú Colección y Descargas	20
Ilustración 15	Descargando el AssetBundle.....	21
Ilustración 16	Botón para abrir la ventana de actualización.....	22
Ilustración 17	Menú de Actualización al seleccionar un nuevo modelo 3D.....	23
Ilustración 18	Menú de Actualización al seleccionar un prefab de Roca	25
Ilustración 19	Opciones al usuario para descargar las imágenes de reconocimiento RA	27
Ilustración 20	Diagrama de Flujo sistema de actualización en la app.....	28
Ilustración 21	Modelo instanciado y detalles del modelo	29
Ilustración 22	Menú de Colección, sección seleccionada, vista del modelo 3D	30
Ilustración 23	Menú de descargas con un elemento descargado	31

Agradecimientos

Quiero agradecer a mi familia por apoyarme durante todos los años de la carrera, sin ustedes no habría podido pasar todo el tiempo que invertí durante la carrera.

A la DGAPA de la UNAM por facilitar los recursos presupuestales para contar con el material que se incluye en esta tesina a través del proyecto PAPIME PE 102120 “DISEÑO DE MANUALES DE CAMPO Y DE LABORATORIO INCORPORANDO FOTOGRAMETRÍA, MODELADO 3D Y REALIDAD AUMENTADA EN LA WEB PARA LA ENSEÑANZA DE LA GEOLOGÍA GENERAL EN LAS INGENIERÍAS: GEOLÓGICA, GEOFÍSICA, MINAS Y METALURGIA, PETROLERA, CIVIL, GEOMÁTICA Y AMBIENTAL”, cuya responsable es la Dra. Mayumy Amparo Cabrera Ramírez, y por la beca que se me fue otorgada durante el desarrollo y publicación del mismo.

Introducción

El objetivo principal de este proyecto fue desarrollar una aplicación que hiciera uso de la tecnología de Realidad Aumentada en dispositivos móviles (Android y iOS) para poder mostrar modelos 3D de rocas, minerales y de los lugares donde estos se pueden obtener (afloramientos), con el propósito de ayudar al proceso de enseñanza de las materias del departamento de Geología de la Facultad de Ingeniería.

Para poder generar parte de los insumos 3D se utilizaron los minerales y rocas que se encuentran dentro del acervo de la facultad, a los cuales se les realizó un proceso de fotogrametría para obtener un modelo lo más fiel y realista posible.

En cuanto a los modelos de los afloramientos se hicieron tomas fotográficas con drones para poder generar el modelo, dándonos una vista superior con la cual se puede ver la zona en general.

La forma en que se utilizan estos insumos actualmente es que los alumnos lograrán tener en sus manos las rocas y minerales durante sus clases de laboratorio, en donde su profesor les daría los datos importantes y sus características. Los afloramientos eran visitados en prácticas de campo, en donde podían explorar la zona bajo la supervisión de los instructores que los acompañaran.

Con el paso del tiempo se ha generado un problema debido a que la matrícula de las carreras ha ido aumentando continuamente, mientras el material seguía siendo el mismo, lo cual llevaba a que los alumnos tuvieran un tiempo más limitado para poder observar los minerales y rocas, además de que las salidas a campo eran cada vez más difíciles en su organización, debido a que se necesitaban llevar grupos más grandes a lugares que podrían resultar peligrosos si no se tiene todo el cuidado.

Con esto en mente, se desarrolló este proyecto para poder subsanar estas problemáticas, al permitir a los estudiantes ver los materiales disponibles en cualquier momento, apoyándonos de la tecnología de Realidad Aumentada la cual nos permite tener una mayor interactividad entre el usuario y el material mostrado; esto hará que los alumnos se vean más interesados en utilizar la aplicación, mejorando la experiencia de aprendizaje.

La Realidad Aumentada se utilizó para poder detectar imágenes dentro de nuestro mundo real las cuales sirven como disparadores para desplegar algún modelo en específico. Para nuestra aplicación se utilizó un nuevo manual de laboratorio donde se tiene una fotografía de la roca, mineral o afloramiento, cuando se detecta esta imagen se desplegará el modelo correspondiente sobre el manual, y el alumno podrá leer y ver sus características al poder hacer más grande el modelo y rotarlo a gusto para ver todos sus detalles.

El estado del arte de la industria RA

Dentro del público general es conocido el uso de Realidad Aumentada en juegos como fue el caso de Pokémon GO, o con el uso de filtros para la toma de fotografías dentro de aplicaciones como Snapchat, que utiliza la realidad aumentada y el reconocimiento facial para que los usuarios coloquen efectos especiales antes de tomar la foto y que se acomode al usuario en tiempo real.

El uso de la realidad aumentada en la educación permite a los profesores el poder mostrar a los alumnos ejemplos virtuales de conceptos u objetos que normalmente no podrían ver de forma directa, con lo que se busca que los alumnos puedan aprender mejor y memoricen la información de forma más sencilla.

Dentro de los ejemplos que son funcionalmente parecidos al producto que se desarrolló en este proyecto, encontramos la aplicación llamada *Dinosaur 4D+*.

En esta aplicación se escanean tarjetas con la aplicación, y al ser detectadas se despliega un modelo 3D del dinosaurio, los usuarios pueden ver e interactuar con el modelo, con la posibilidad de modificar su escala y lanzar animaciones del dinosaurio o ver información más detallada del mismo.



Ilustración 1 Captura del video demostrativo de la app Dinosaur 4D+

Otro ejemplo conocido es *Element 4D* una aplicación que utiliza la realidad aumentada para poder mostrar elementos de la tabla periódica y como estos interactúan entre sí, en esta aplicación se ocupan unos cubos que se pueden imprimir y armar para poder detectar que elemento de la tabla periódica se está

observando, dentro de esto se buscaba poder mostrar información relevante del elemento, como su peso atómico.



Ilustración 2 Captura de pantalla de la aplicación Element 4D

La tecnología de realidad aumentada no solo se ha utilizado dentro de las industrias del entretenimiento y educación, también se ha utilizado en las empresas para poder realizar entrenamientos profesionales. Buscando tener experiencias más interactivas que los entrenamientos básicos normales (videos, textos de información) e intentar reducir gastos en industrias donde se necesita aprender a operar maquinaria muy costosa, reduciendo así riesgos que podrían existir durante la capacitación de los nuevos empleados. Dentro de estas industrias se busca implementar en una mayor medida sistemas de Realidad Mixta, que toma la base de la realidad aumentada y mejorarla para tener una mejor inmersión al tener elementos reales que interactúen con la realidad virtual.

Las tecnologías más comunes para el desarrollo de aplicaciones de realidad virtual dentro de dispositivos móviles son ARCore, ARKit y Vuforia. ARCore es una tecnología desarrollada por Google, principalmente para dispositivos Android, pero también tiene compatibilidad con dispositivos iOS, ARKit es desarrollada por Apple exclusivamente para sus dispositivos. Finalmente, Vuforia es desarrollada por Qualcomm teniendo sus propios algoritmos para el reconocimiento RA permite colocarse en proyectos para ambas plataformas y logra traer las experiencias para dispositivos que no soporten ARCore o ARKit.

Dentro de todos estos ejemplos vemos que se realizaron aplicaciones que se necesitan instalar dentro del dispositivo para poder funcionar correctamente, sin embargo, esta no es la única forma de poder disfrutar de experiencias RA,

actualmente se ha desarrollado una nueva API que nos permite poder lanzar experiencias de AR desde los navegadores web de distintos dispositivos llamados WebXR, esta nueva API aun utiliza como base ARCore para poder desplegar la experiencia, por lo que sigue estando atada a los dispositivos que soporten esta tecnología para poder desplegar la experiencia. Además, que no todas las funcionalidades que existen dentro de ARCore funcionan dentro de WebXR debido a que es una tecnología relativamente nueva y que tiene que funcionar mediante navegadores web.

A pesar de ciertas limitaciones dentro de la tecnología WebXR puede ser una buena alternativa para usos sencillos de la Realidad Aumentada, como mostrar algún modelo en nuestra realidad, teniendo en cuenta que para este tipo de muestras se deberá de contar con conexión a internet debido a que se tiene que cargar la página web y todos los modelos e imágenes que la compongan.

Preproducción

Antes de iniciar el proceso de desarrollo se llevaron a cabo múltiples juntas con la Dra. Mayumy Amparo Cabrera Ramírez, quien es la responsable del proyecto PAPIME de donde nace el desarrollo, en las cuales se platicó lo que se necesitaba de la aplicación en cuanto a funcionamiento y uso, con base a estas juntas se obtuvieron los siguientes requerimientos.

Requerimientos funcionales:

- La aplicación debe de reconocer varias imágenes y por cada imagen que reconozca deberá de mostrar el modelo correspondiente.
- Los modelos se deberán de poder rotar y cambiar su escala.
- Mostrar ventanas con texto correspondiente al modelo.
- Los modelos se deben de poder descargar de un servidor remoto, para aligerar la aplicación base.
- Sistema para eliminar los modelos previamente descargados.

Requerimientos no Funcionales:

- Los modelos deben ser ligeros para el funcionamiento en realidad aumentada, y facilitar su descarga.
- El sistema contiene una base de datos que maneja la relación de los modelos con las imágenes que se reconocen.
- Sistema de actualización para agregar más elementos al sistema de reconocimientos de imágenes.

Con base en los requerimientos se decidió separar las funciones principales de la aplicación para poder tener líneas de trabajo en paralelo siendo estas:

- Implementación reconocimiento de imágenes
- Sistema de descarga del contenido a la aplicación base
- Flujo de los menús y transiciones entre escenas
- Generación de una base de datos para el control del contenido descargado
- Diseño y generación de la interfaz de usuario
- Implementación del visualizador de contenido sin Realidad Aumentada
- Sistema de actualización

Al tener las tareas separadas nos permitió el aumentar la productividad, dado que, cada miembro estaría trabajando en un área donde tuviera mayor experiencia. Con lo que los roles quedaron designados de la siguiente manera:

- Diego Iñaki Garciarebollo Rojas, administrador del proyecto y desarrollador de software

- Ricardo Hernández Gómez, administrador de bases de datos y desarrollador de software
- Miguel Ángel Cruz Zavala, diseñador gráfico
- Alberto Alonzo Lona López, desarrollador de software

Al ser un equipo pequeño y al ser todos estudiantes se decidió tener una metodología de trabajo tipo Scrum, en donde cada semana se realizaban juntas dentro de las cuales se discutían los avances que se tenían, los problemas que surgían dentro del desarrollo y los objetivos de la semana para la aplicación. Al haber paralelizado el desarrollo de la aplicación en su mayor parte rara vez surgían tiempos de inproductividad para algún miembro.

Al momento de escoger las tecnologías con la que se desarrollaría la aplicación se tomó en cuenta los conocimientos y habilidades de cada miembro, como también los alcances de la aplicación, principalmente que se quería desarrollar versiones para dispositivos Android e iOS. Con el propósito de facilitar el desarrollo se decidió utilizar un motor gráfico, esto nos permitió realizar ambas versiones de la aplicación al mismo tiempo.

Dentro de los motores gráficos disponibles se decidió utilizar Unity, debido a que contábamos con experiencia en su uso y que cubría perfectamente nuestras necesidades, al tener la posibilidad de desarrollar la aplicación tanto para el sistema operativo Android e iOS al mismo tiempo, dado que ya cuenta con un Framework que encapsula a ambos sistemas de realidad aumentada (ARCore y ARKit respectivamente), por lo que se pudo simplificar el proceso de programación al ser multiplataforma. Sin embargo, al no contar con un dispositivo Mac no se lograron realizar las pruebas de funcionamiento para los dispositivos iOS ya que se necesitaba este dispositivo para realizar la generación e instalación de la aplicación en un dispositivo iOS.

Otra opción que se tomó en cuenta para la implementación del sistema de reconocimiento de imágenes en realidad aumentada fue Vuforia, desarrollada por Qualcomm, el cual se puede añadir a los proyectos de Unity. Esta opción nos proporcionaba una base de datos que se puede modificar desde una pagina web para poder incluir de forma sencilla nuevas imágenes a ser reconocidas por el sistema, sin embargo, el uso de este es restringido a una licencia de paga que tiene un costo de \$42 USD (cuarenta y dos dólares americanos) mensuales para poder publicar y distribuir la aplicación.

Se decidió utilizar el sistema de ARFoundation, el cual no tiene ningún costo y que es un sistema que se va implementando a los teléfonos recientes de forma nativa. Además de que el proceso de añadir nuevas imágenes y modelos 3D al sistema de Vuforia implica tener que volver a generar una nueva versión de la aplicación completa, dado que se debe de asignar de forma manual el modelo a la imagen que se reconoce para que funcione con múltiples objetos, mientras que el sistema de

actualización que se realizó para ARFoundation no se necesita lanzar actualizaciones continuamente en las *apps stores* [7].

Debido a estas diferencias fundamentales fue que se decidió utilizar el sistema nativo de Unity de ARFoundation para poder realizar el sistema de reconocimiento de imágenes en realidad aumentada.

Al tener delimitado el proyecto generamos varios diagramas de flujo, intentando encapsular lo más posible los distintos comportamientos que se tendrían dentro de la aplicación.

Dentro de la escena RA (donde se tienen todos los comportamientos de Realidad Aumentada) tenemos la parte de actualización de la biblioteca de imágenes a reconocer y el proceso de instanciación de los modelos 3D al reconocer la imagen.

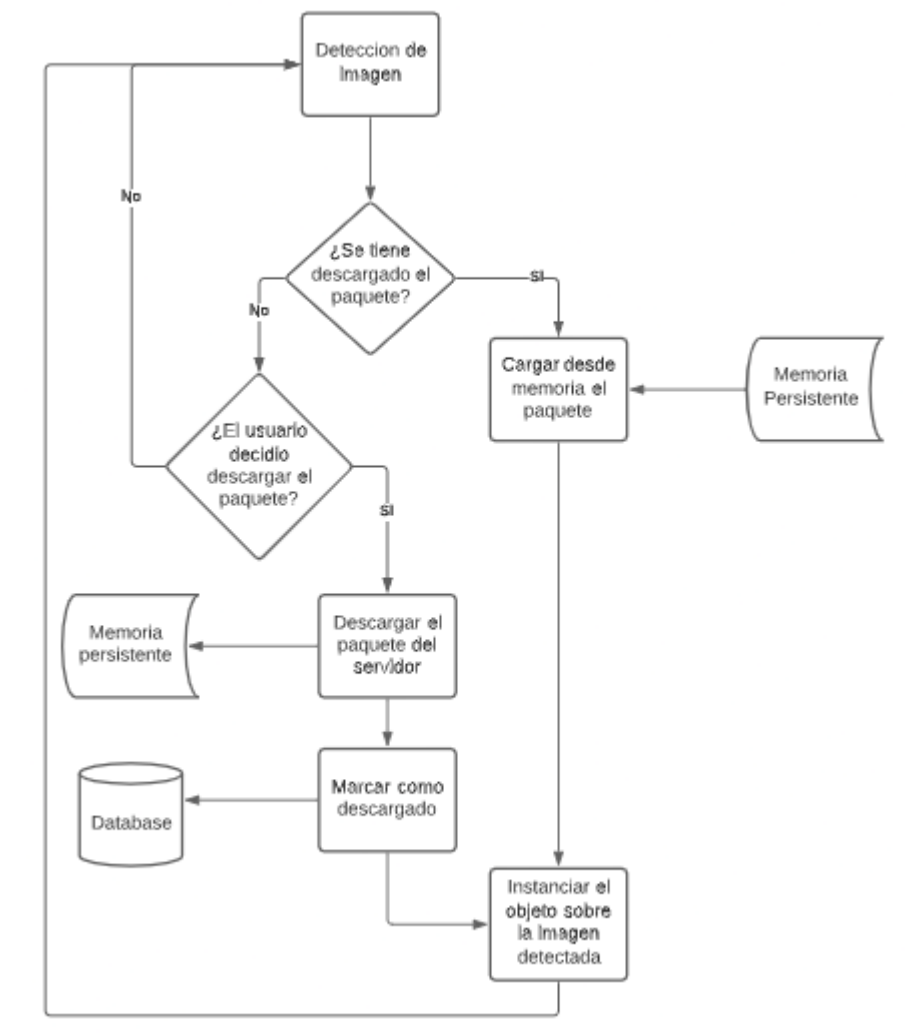


Ilustración 3 Diagrama de flujo instanciación modelo 3D

De forma preliminar siempre consideramos que el usuario debería de tener la posibilidad de escoger cuando descargar los modelos, evitando las descargas automáticas para no consumir datos móviles sin autorización, además de que esto permitirá al usuario controlar la cantidad de memoria que ocupara la aplicación al tener un menú separado para descargar o eliminar los paquetes que contienen los modelos, en caso de que ya se hubieran descargado en un momento anterior se cargara de forma automática el modelo 3D.

El comportamiento de reconocimiento de imágenes es un proceso cíclico que debe de ejecutarse siempre que esté viva la escena de Realidad Aumentada, sin embargo, los comportamientos de la ilustración superior solo se ejecutan cuando se detecta una imagen dentro de la vista de la cámara del dispositivo.

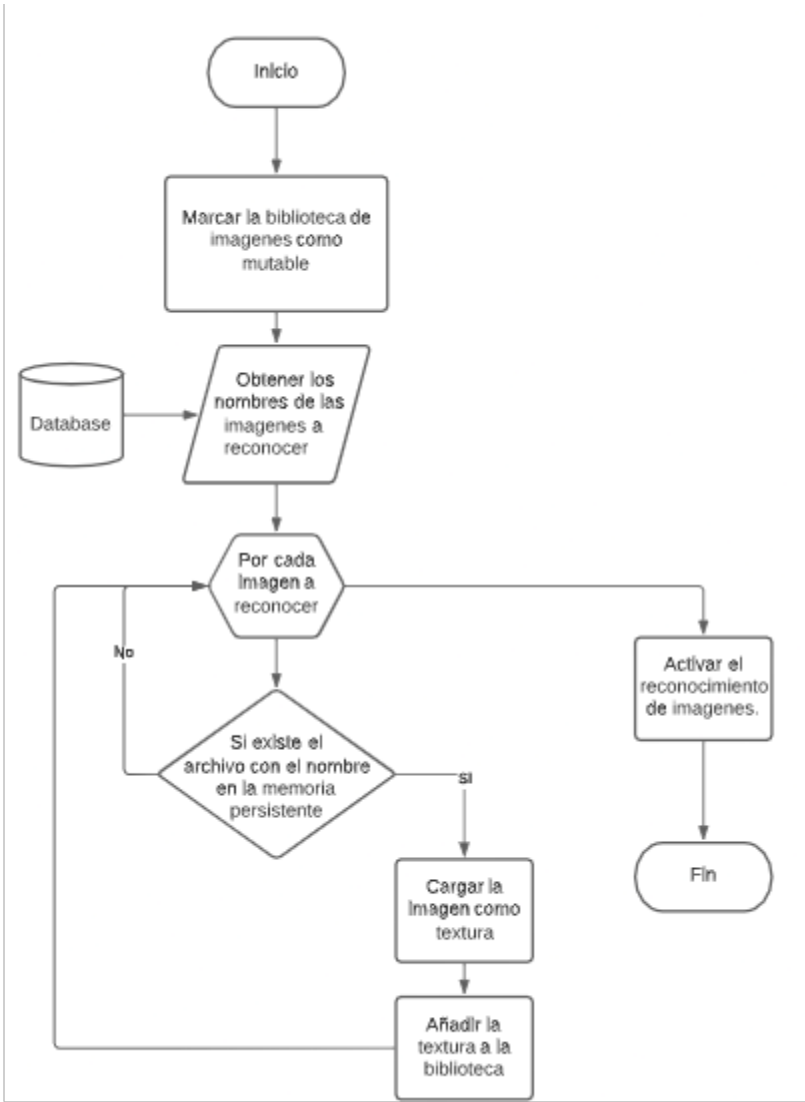


Ilustración 4 Diagrama agregar imagenes a la biblioteca de imágenes

Este proceso solo se ejecutará una sola vez cada que se abre la escena de Realidad Aumentada al inicio de esta escena al ser un comportamiento de configuración. Al desarrollarlo nos dimos cuenta de que solo se debe de ejecutar una sola vez mientras que la aplicación esté abierta, debido a que el sistema de Realidad Aumentada es persistente durante la sesión de la app, si se cierra se deberá de volver a ejecutar para agregar las imágenes.

Desarrollo de la aplicación

ARGEO (Aplicación para dispositivos móviles)



Ilustración 5 Presentación ARGEO

En las siguientes secciones se hablará sobre cada parte del desarrollo de la aplicación en la que estuve involucrado de forma directa al realizar alguna actividad, empezando con el manejo de los modelos 3D como insumos para la aplicación, el sistema de reconocimiento de imágenes, el sistema de descarga de contenido y el sistema de actualización.

De forma preliminar se realizó un bosquejo de la interfaz de usuario de la aplicación, en donde se muestra cómo se vería el menú principal y la pantalla cuando se reconoce una imagen con el modelo 3D desplegado. [Ilustración1]

Modelado 3D

La técnica utilizada para obtener los modelos 3D fue la fotogrametría, la cual consiste en obtener fotos del objeto en distintos ángulos, estas fotos son ingresadas a un software especializado el cual detecta su posición y genera una malla detallada del modelo, y es texturizado utilizando fragmentos de las fotos.

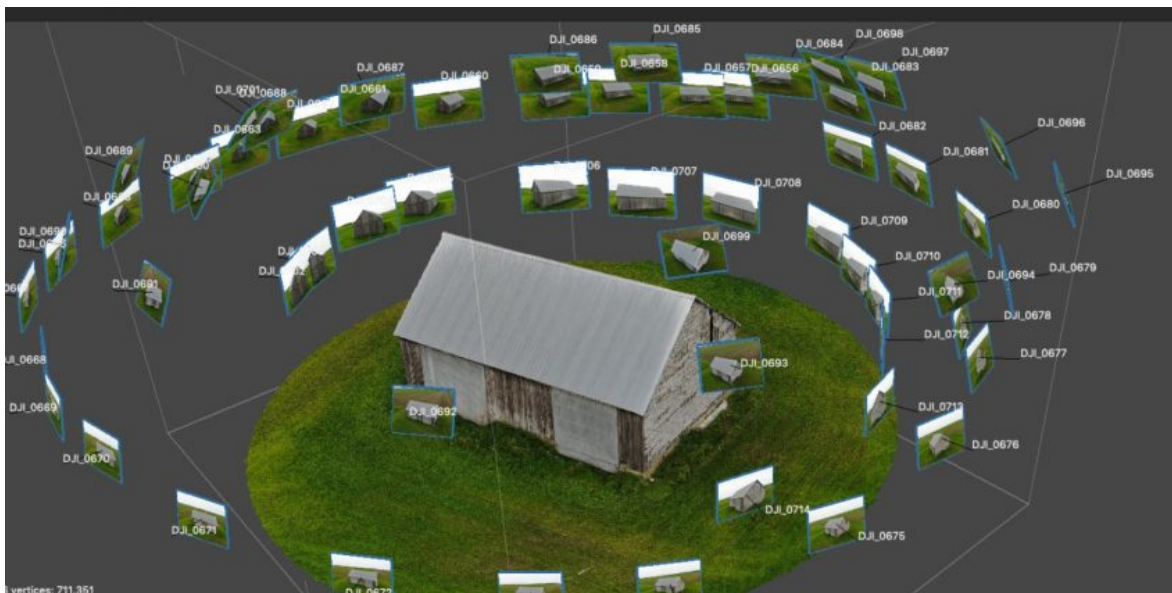


Ilustración 6 Fotogrametría de una casa

Se realizaron la toma de fotografías de las rocas y minerales que se encuentran dentro del acervo de la Facultad de Ingeniera. Para la toma de fotos en distintos ángulos se colocó el objeto sobre un pedestal de madera el cual tiene un brazo rotatorio donde se coloca la cámara, generando un resultado similar al que se ve en la Ilustración 2.

Una vez generadas todas las fotografías del objeto que se quería digitalizar, se generó el modelo 3D al ingresarlas al software de Meshroom, con lo cual se obtuvo un modelo de alta calidad. Sin embargo, también se generó geometría que no pertenecía al objeto en sí, dado que también capturaba al mismo pedestal, por lo que, una vez obtenido el modelo 3D este se tuvo que pasar a un software de modelado (Blender) en donde se retiró todas las partes que no correspondieran al mineral o roca que se quería modelar.

Dado a que el modelo generado es de gran calidad, para el uso dentro de la aplicación de realidad aumentada se tuvo que disminuir su calidad para que este se pudiera desplegar en cualquier dispositivo, además de que buscábamos disminuir su tamaño debido a que todos los modelos se van a poder descargar al teléfono y que su tamaño no sea tan restrictivo para usuarios que consuman la aplicación con

datos móviles. Por lo que, dentro del software de modelado de Blender, se hizo una disminución de la cantidad de polígonos que se tenía en el modelo para que, en promedio, se tuvieran un total de cincuenta mil polígonos por modelo, en algunos casos fue necesario que se tuvieran un poco más de polígonos para conservar ciertos detalles y que no se deformara mucho el modelo original.

A cada modelo se le añadió una pequeña base en donde se muestra el nombre del modelo junto a los logos de la Facultad de Ingeniería y el de la UNAM, con el propósito que sean fácilmente identificables.

Dentro de esta parte del desarrollo, únicamente participé de forma activa en la optimización de los modelos obtenidos para su uso en la aplicación de realidad aumentada.

Sistema de Reconocimiento de imágenes

El Framework ARFoundation de Unity cuenta con una implementación sencilla para el reconocimiento de imágenes. Para poder utilizarlo se necesita generar un archivo llamado ImageLibrary, el cual contiene las imágenes a reconocer, su nombre con que se identificara, sus dimensiones en el mundo real (dato opcional para Android, pero obligatorio para iOS) y tiene la opción de mantener la textura en la memoria RAM en tiempo de ejecución [3]. Este archivo es compilado al momento de generar el paquete de instalación de la aplicación, en el caso de Android se genera un archivo con la extensión imgdb (image database) que contiene toda la información proporcionada, este archivo se encuentra dentro del paquete de instalación. Se puede modificar la biblioteca de imágenes mientras la aplicación esta funcionando, lo cual sirve para poder agregar nuevas imágenes sin la necesidad de tener que generar un nuevo archivo de instalación y subir la actualización a la tienda o sistema de distribución, el único problema es que estos cambios no son permanentes por lo que cada vez que se abre la aplicación se repetirá el proceso de agregar las imágenes (ver la explicación más detallada en la sección de Sistema de Actualización).

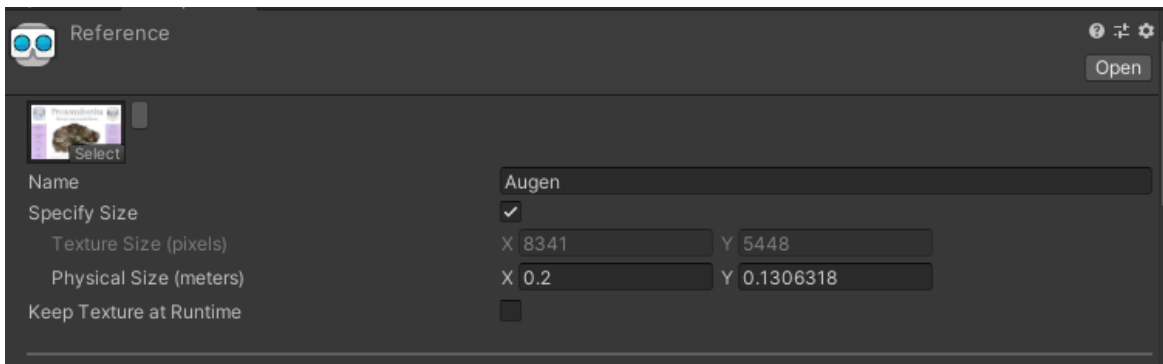


Ilustración 7 Ejemplo Image Library

Los datos que se tienen dentro de la Image library son:

- La imagen para reconocer: en este caso se utilizó la ficha técnica del mineral o roca que se fuera a desplegar, recortada para abarcar únicamente donde se encuentra el mineral o roca.
- Nombre: nombre con el que se identifica la imagen dentro del programa, en nuestro caso ocupamos el nombre del paquete que va ligado a la imagen del mineral, con el propósito de poder llamarlo por nombre directamente.
- El tamaño físico de la imagen a reconocer: en este punto se ocupó como valor predeterminado de 20 cm en el ancho que es el tamaño para una hoja carta.

Para los casos de los afloramientos se quiere utilizar una imagen representativa del lugar junto con un código QR para poder hacer más sencilla el reconocimiento.



Ilustración 8 Demo disparador afloramiento

Una vez teniendo el esquema básico de la biblioteca de imágenes se hizo una escena básica con los componentes necesarios para poder desplegar la escena en RA y con el reconocimiento de imágenes. Esto al agregar a nuestra escena las pre hechos *AR Session* y *AR Session Origin*. La primera controla las entradas de la pantalla del dispositivo hacia la aplicación y además hace el frame rate coincida con el de la pantalla. El segundo objeto es quien tiene los componentes para reconocer el movimiento del celular y que estos se vean reflejados en un movimiento dentro del ambiente virtual. Además de que cuenta con la cámara virtual que hará que se rendericen los objetos del ambiente virtual y nos muestren lo que ve nuestra cámara real.

Se le agrego el componente de *AR Tracked Image Manager* que es un componente pre hecho que reconoce las imágenes de la biblioteca de imágenes que se le proporcione (puede tener varias la aplicación, pero en este desarrollo solo se hay una) e instancia un objeto en las coordenadas donde reconoció una imagen de la biblioteca. El objeto que se instancia se asigna directamente en el campo *Tracked Image Prefab*.

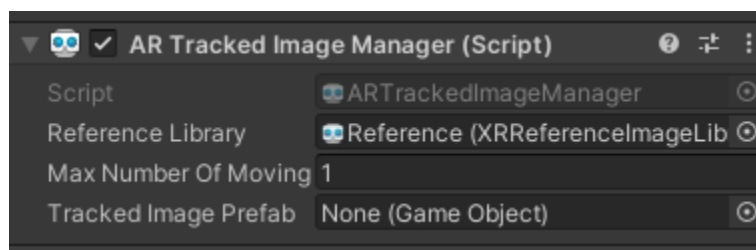


Ilustración 9 Componente AR Tracked Image Manager

Una de las principales restricciones que se tienen con el manejador pre hecho es que solo se puede mostrar un solo objeto con todas las imágenes de la biblioteca.

Sin embargo, nosotros queríamos poder mostrar un modelo distinto dependiendo de la imagen que se reconoce. Para satisfacer esta necesidad desarrollé un código que reconoce los eventos que levanta el componente *AR Tracked Image Manager*, siendo el evento *On Tracked Images Changed* el cual nos devuelve un argumento con tres listas distintas, *added*, *updated* y *removed* [4]. Para nuestro código realmente solo nos interesaron las primeras dos listas.

Cuando se agrega un nuevo elemento a la lista *added* significa que el programa reconoció por primera vez durante la sesión una de las imágenes de la biblioteca de imágenes. Al obtener la referencia a la imagen reconocida se obtiene el nombre que se le asignó a la imagen (como podemos ver en la Ilustración 3) y dado que el nombre que se le asignó es el mismo nombre que tiene el paquete que contiene el modelo 3D que se quiere mostrar, se instancia el modelo 3D dentro del mundo virtual, haciendo así que cada imagen pueda desplegar un objeto distinto.

Para poder instanciar el modelo dentro de nuestro mundo virtual primero se checa si ya se había descargado previamente el paquete de información correspondiente al modelo, en caso de que no se tuviera descargado se muestra la opción de descargarlo o no al usuario (Como se puede observar en la Ilustración 6).

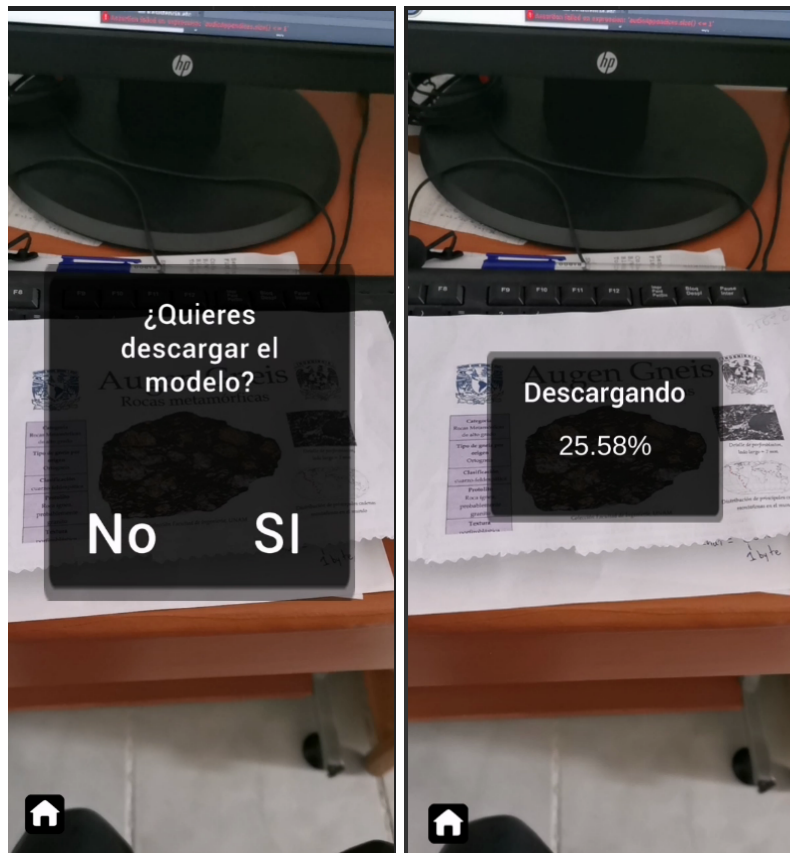


Ilustración 10 Descarga de contenido

Sistema de descarga de contenido

Para el sistema de descarga de contenido de un servidor remoto se utilizó el sistema de AssetBundles, un formato de archivo que puede generar Unity el cual contiene datos específicos por plataforma que contienen *Assets* que no sean códigos [5] (como modelos 3D, imágenes o archivos de audio). Al tener todos los modelos como AssetBundles nos ayuda a poder reducir el tamaño de la aplicación base, sin embargo, la aplicación no puede hacer mucho si no tiene al menos un paquete descargado o una conexión a internet.

Utilicé una herramienta descargable para el editor de Unity llamada *AssetBundle Browser* el cual nos permite ver cómo está compuestos nuestros paquetes y ver que archivos son los que se están cargando al llamarlo desde nuestro código principal (Ilustración 7).

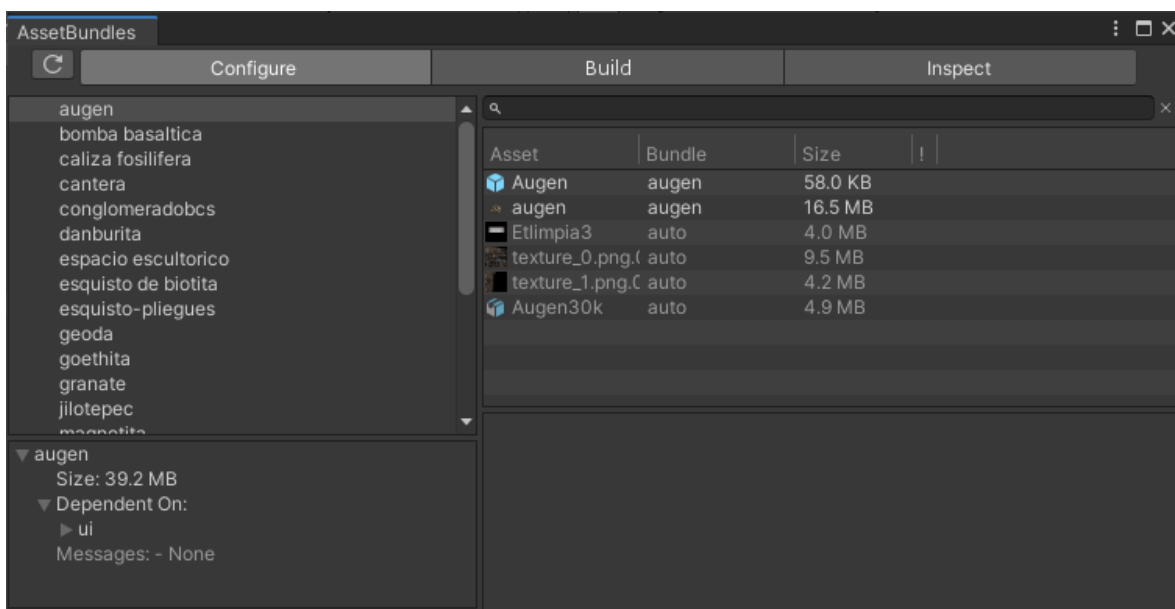


Ilustración 11 Ventana AssetBundle Browser

Como se puede observar nuestro paquete contiene al modelo principal, con las texturas, que es traído por el pre hecho que se muestra en la aplicación final. Se hizo este segundo objeto debido a que se muestra información en ventanas de interfaz de usuario. Además de que se incluye una pequeña imagen que corresponde a la miniatura que se podrá ver en el menú de colección.

El assetBundle se genera en la pestaña de *Build* donde se escoge hacia que plataforma se van a generar, la dirección en nuestro computador donde se van a guardar los archivos generados y el formato de compresión que se quiere usar (el valor por defecto es LZMA)

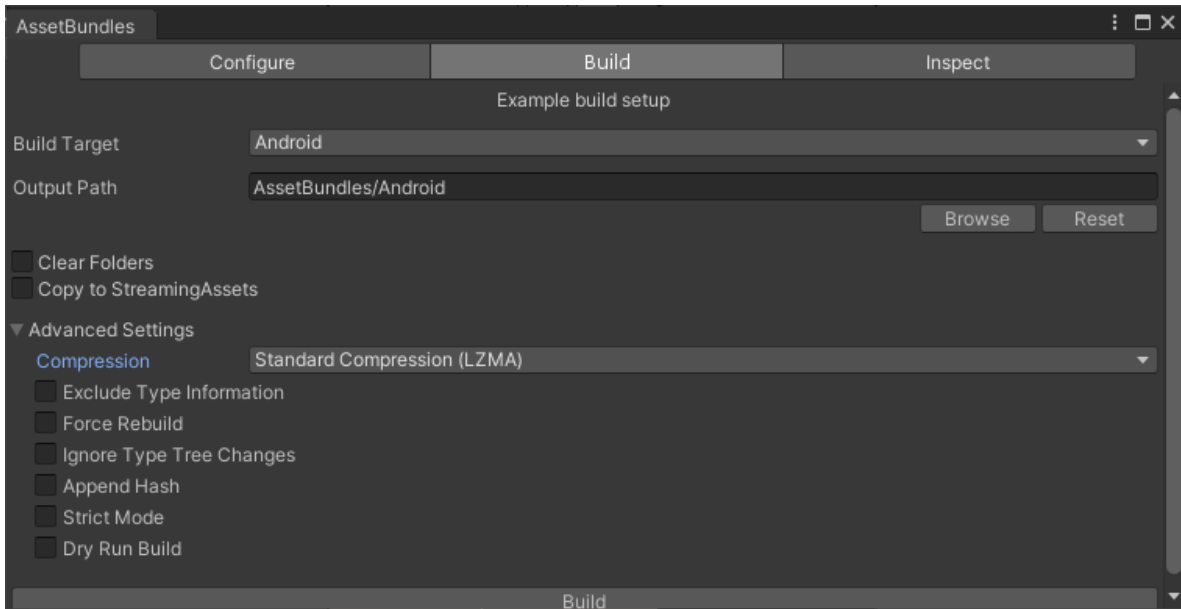


Ilustración 12 Ventana AssetBundle Browser, sección Build

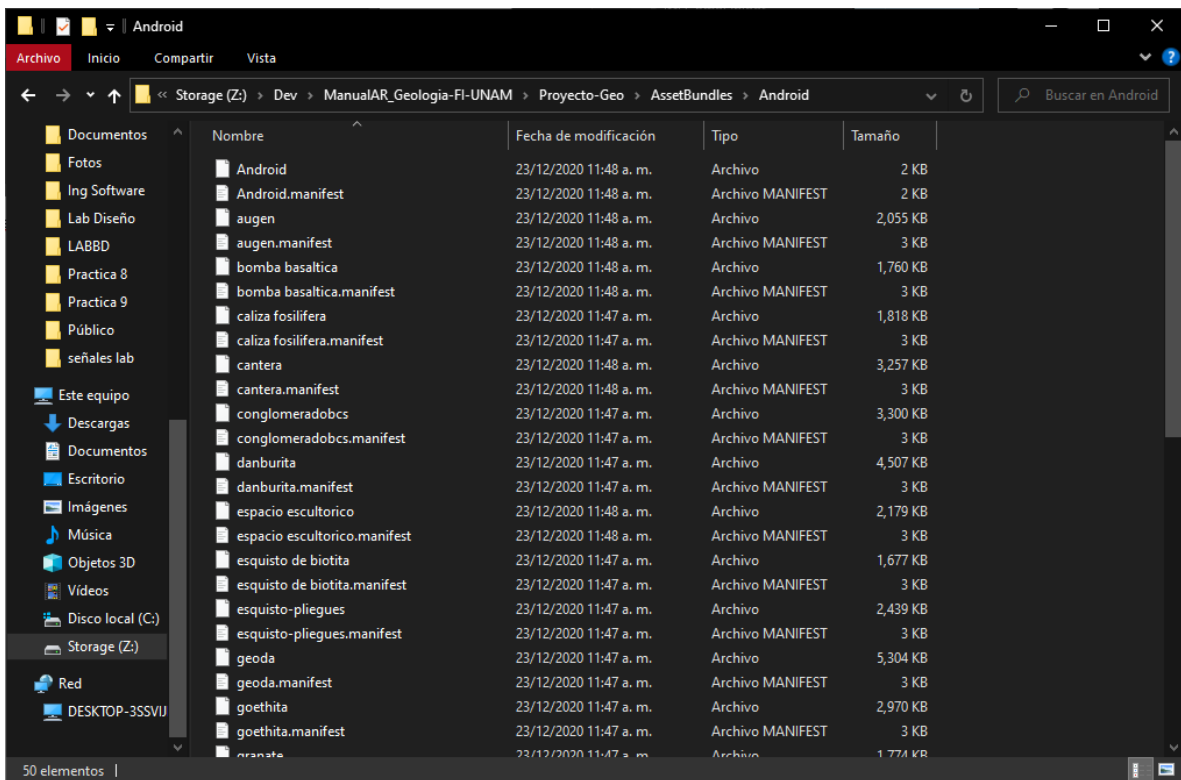


Ilustración 13 Archivos AssetBundle Generados

Se generan dos archivos por paquete de AssetBundle, el paquete en sí, que contiene todos los archivos que se incluyeron, y un manifiesto que es legible para humanos en general, en donde podemos ver todos los nombres de los archivos del paquete, si hay dependencias con otros paquetes. Los paquetes son los archivos

que no tienen extensión, estos archivos se pueden subir a un servidor remoto HTTP para poder ser descargados después con la aplicación.

La descarga puede ocurrir en tres momentos distintos dentro de la aplicación, primero dentro de la vista de Realidad Aumentada (Ilustración 6), en el menú de descargas o al momento de intentar acceder a un modelo dentro del menú de colección (Ilustración 10).



Ilustración 14 Descarga en menú Colección y Descargas

Para poder realizar la descarga del contenido dentro de la aplicación se utilizó la biblioteca de Unity que maneja las conexiones *UnityEngine.Networking* debido que contiene los métodos para realizar comunicaciones HTTP necesarias para poder descargar los paquetes del servidor.

Primero se descarga el archivo de manifest de la plataforma el cual contiene los datos como el nombre de todos los assetbundles, su hash de verificación, la versión y si tiene alguna dependencia con otro assetbundle. Se crea un espacio para el cache con la clase *CachedAssetBundle* que toma como parámetros el nombre del paquete y el hash, con este espacio se genera la solicitud del contenido usando el método de *UnityWebRequestAssetBundle.GetAssetBundle* que toma como parámetros la dirección web en donde se encuentra el AssetBundle, el espacio de cache que se generó y si se quiere hacer la comprobación CRC para encontrar

errores en la descarga. Con esto lo que se hace es una petición GET directamente al servidor pidiendo el paquete.

Con esto se hace la petición hacia el servidor remoto para iniciar la descarga, se muestra en la pantalla al usuario el porcentaje de descarga que lleva cada frame (Ilustración 11).



Ilustración 15 Descargando el AssetBundle

Muestra del modelo

Finalmente, en este proceso es la muestra del contenido, debido a que el AssetBundle ya se encuentra descargado a la memoria interna del modelo primero se recupera de la memoria interna. Una vez que se tiene cargado en la memoria RAM del dispositivo lo que se hace es cargar de forma asíncrona el objeto que contiene el modelo 3D del material a mostrar. Este se instancia en la posición donde se reconoce la imagen con la cámara de AR, o enfrente de nosotros para poderlo ver en la vista de colección sin la necesidad de la cámara AR [Ilustración 16].

Una vez que se cargó el objeto por completo se quita de memoria RAM el paquete del AssetBundle, con esto se queda el modelo en nuestro mundo virtual sin tener

cargadas cosas innecesarias en la memoria RAM como la miniatura para el menú de colección.

Dentro de cada frame de la aplicación nuestro modelo podrá actualizar su posición dentro del mundo virtual para poder seguir la imagen reconocida si esta se mueve dentro del mundo real.

Sistema de Actualización

Como parte del sistema de actualización en general se generó un nuevo menú contextual para el editor de Unity con el propósito de facilitar el proceso para generar nuevo contenido para la app.

Primero lo necesario es tener el modelo 3D ya optimizado dentro de nuestra carpeta de proyecto (Assets/Models/ ya sea en la carpeta de “Rocas y Minerales” o la de “Afloramientos” con el propósito de tener todo organizado) también si ya se cuenta con la imagen para ser reconocida dentro de la escena AR esa se deberá de colocar dentro de la dirección *Assets/Models/ImagenesRecog.*

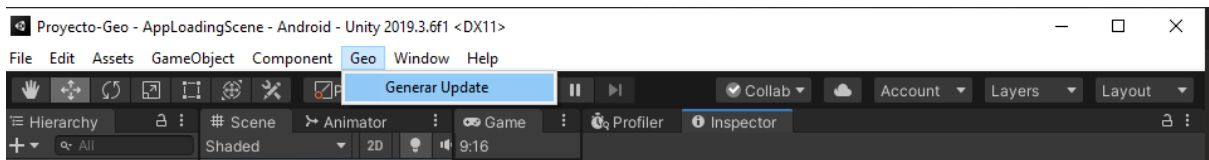


Ilustración 16 Botón para abrir la ventana de actualización

Con la ventana abierta veremos cuál va a ser la nueva versión que se va a generar, y tenemos la opción de escoger el modelo 3D que se quiera agregar o actualizar para nuestra aplicación.

Viendo el caso de agregar un modelo completamente nuevo la ventana de actualización desplegará todos los componentes que tendrá el nuevo modelo 3D [Ilustración 12] dentro de estos se encuentran:

- El tipo de modelo que es: si pertenece a las *Rocas*, *Minerales* o *Afloramientos*
- En caso de que sea roca se desplegara el menú para seleccionar su categoría, *Ígnea*, *Sedimentaria*, *Metamórfica* o *Estructura sedimentaria*. En caso de no pertenecer a las rocas este menú se ocultará de forma automática.
- Nombre oficial: que es el nombre que se desplegara al usuario al verlo dentro de la aplicación tanto en el menú de descargas como en el de colección.
- Nombre del prefab: nombre con el que se guardara el assetBundle, este no se puede repetir con ningún otro assetBundle.
- Descripción mínima: un pequeño texto que se despliega sobre el modelo que contiene la información más importante.

- Descripción detallada: un texto de mayor extensión que puede contener toda la información del modelo.
- Miniatura: imagen que se desplegara dentro del menú de colección una vez que haya sido descargado el assetBundle
- Ficha: imagen que se utilizara para el reconocimiento en RA dentro de la aplicación.

Para poder agregar un nuevo elemento dentro de la misma actualización lo que se hace es presionar el botón que se encuentra en la parte superior de la ventana de actualización (tiene el símbolo de “+”) con lo que se agregara un nuevo elemento a la ventana.

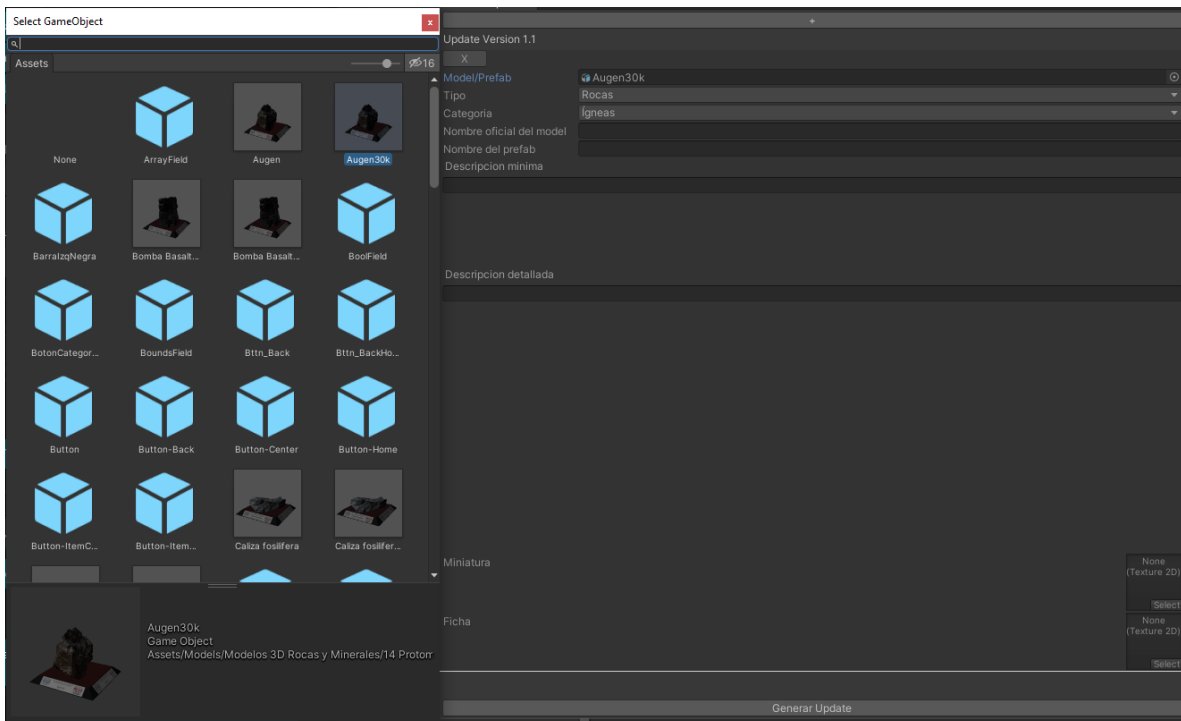


Ilustración 17 Menú de Actualización al seleccionar un nuevo modelo 3D

En caso de que se quiera realizar algún cambio dentro a un prefab preexistente se deberá de seleccionar dentro del campo de *Model/Prefab* [Ilustración 13].

De forma automática se mostrarán toda la información que ya se tenía del modelo. Y se podrá realizar todos los cambios a los campos que se tienen de forma libre.

Una vez que se tengan todos los cambios y agregados todos los modelos nuevos que se quieran para la actualización se deberá de dar click en *Generar Update* que se encuentran en la parte inferior de la ventana de actualización. Con esto se generarán los nuevos prefabs y se asignara su nombre de assetbundle como también se aplicarán todos los cambios a los prefabs que ya existan. Debido a que la aplicación cuenta con una base de datos interna que lleva el control del nombre, categoría, tipo, nombre del assetBundle y nombre del archivo de la imagen a

reconocer para el AR, la base de datos se verá modificada para reflejar todos los cambios que se hayan hechos en la actualización. A la par que se generara un archivo que se llama *updates.json* que contiene los cambios que sufrió la base de datos para que se puedan reflejar en la aplicación en dispositivos que ya la tengan instalada.

La imagen de miniatura (si se escogió una) es agregada al *assetBundle* para que pueda ser descargada junto con el modelo.

La imagen de la ficha, en caso de no estar dentro de la carpeta de *Assets/Models/ImagenesRecog* es copiada al directorio. Además de que es copiada a la dirección *AssetBundles/ImageRecog* sin extensión para que se pueda subir directamente todo el contenido de la carpeta *AssetBundles* al servidor web y pueda ser descargada sin problemas.

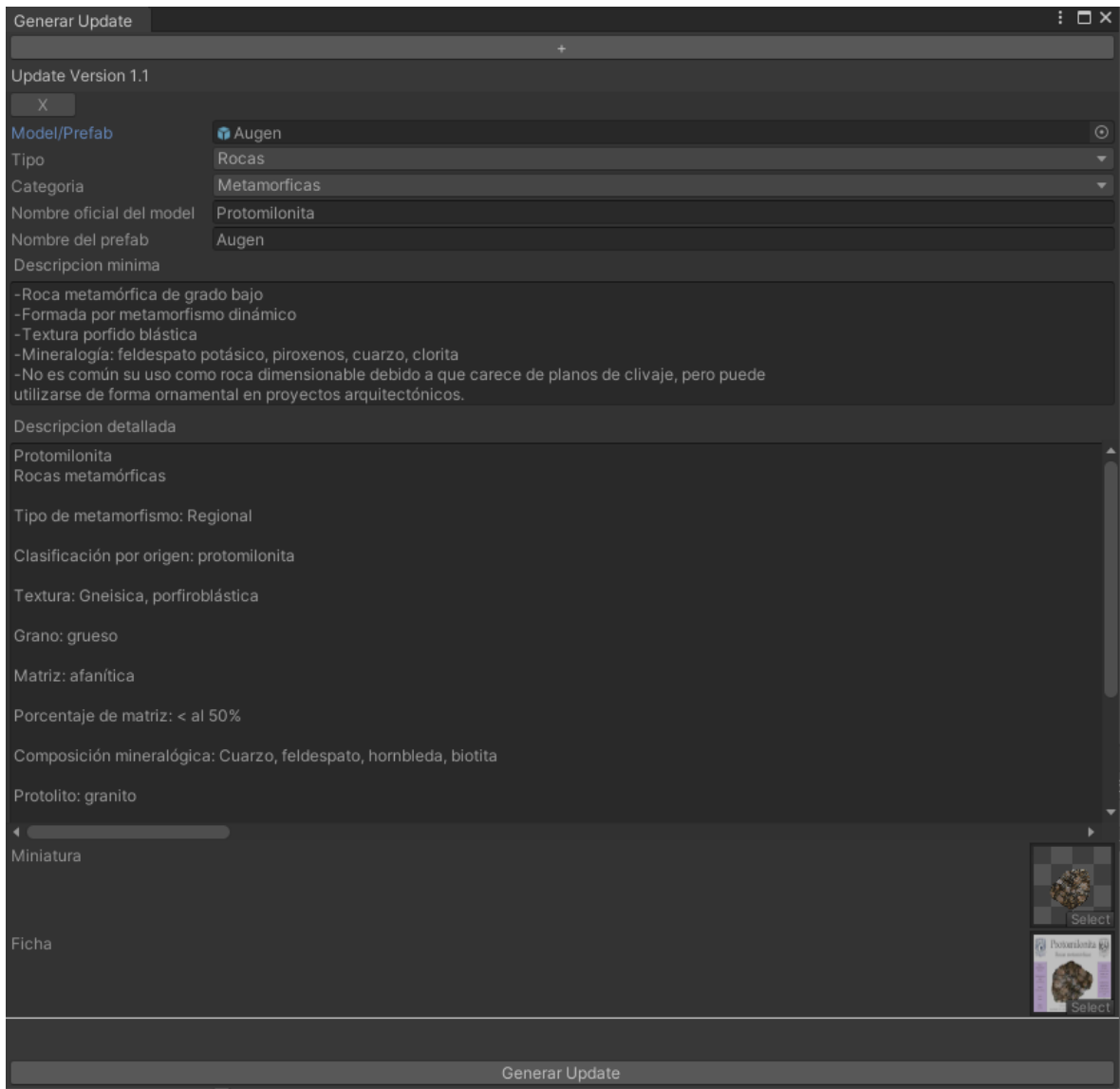


Ilustración 18 Menú de Actualización al seleccionar un prefab de Roca

El archivo *updates.json* generado en este paso contiene una lista de actualizaciones, cada actualización contiene la versión de la actualización y los cambios que se tienen que realizar dentro de la base de datos, se tienen únicamente dos opciones añadir un nuevo elemento o modificar un elemento existente. Al modificar un elemento existente se obtiene la llave primaria del elemento en la tabla el cual es un valor numérico de ID, con el cual se puede modificar el registro cambiando los valores como su nombre, categoría, tipo nombre del assetbundle, el nombre de la imagen a reconocer en la escena AR. En el caso de agregar un nuevo elemento solo se ingresarán los mismos datos como un nuevo renglón a la tabla. Todos estos cambios también se ven reflejados en la base de datos que se tiene en la computadora de desarrollo por lo que, si se lanza una nueva versión para actualizar en tiendas, está ya contendrá todos los cambios que se le hayan hecho.

Cada vez que se realiza una nueva actualización de manera automática se aumenta la versión mínima de la app, es decir si la app estaba en la versión 1.0 y se hace una actualización con esta implementación la versión registrada en el *update.json* es la 1.1 y esta es la versión que se queda asignada dentro de los *build settings* de Unity para que si se hace alguna construcción sea de la versión 1.1

Dentro de la aplicación instalada ya en un teléfono el proceso de actualizar se lleva a cabo al momento de abrirse la app en la primera pantalla antes de entrar al menú general.

Primero revisa si ya existe la base de datos en su memoria en la dirección de datos permanentes, debido a que el archivo de instalación (apk) es un archivo comprimido es necesario copiar la base de datos a memoria regular del teléfono para poder leer y modificar los datos que pueda contener. Una vez que la haya copiado genera en el teléfono un archivo llamado *appConfig.json* en el cual solo se guarda la versión actual de la app basados en la versión que se le dio al apk.

Una vez concluido el paso anterior busca dentro del servidor web si existe el archivo *updates.json* en caso de que exista lo descarga de forma automática y checa si la versión de las actualizaciones que contiene son mayores a la versión que se tiene registrada en el archivo *appConfig.json* en caso de que si sea mayor la versión del archivo descargado va checando de forma ordenada todas las modificaciones que se tienen que hacer a la base de datos y las va aplicando. Una vez que haya acabado de checar todas las actualizaciones que se tenían en el archivo se modifica el archivo *appConfig.json* para que guarde la versión más reciente que se haya tenido dentro del archivo de *updates.json*, para evitar que cargue actualizaciones que ya hayan sido efectuadas dentro de la base de datos.

Debido a que la base de datos contiene todos los nombres de las imágenes a reconocer para la escena de AR se obtiene una lista de todos estos nombres, con el nombre del archivo se checa dentro de la memoria interna del teléfono (en la dirección de instalación bajo la carpeta de *imgRec*) si se encuentra descargado el archivo con el mismo nombre, en caso de que no la tenga descargada se hace una petición HEAD de http al servidor para poder obtener el dato del tamaño del archivo, este tamaño se va sumando con el de todas las imágenes que no tenga descargadas para poder mostrárselo al usuario. Al terminar de buscar todas las imágenes dentro del servidor se le lanzara la opción al usuario sobre si las quiere descargar o no mostrándole el tamaño total de los archivos a descargar en MB (este tamaño no es exacto dado que se hizo un redondeo al siguiente valor entero).

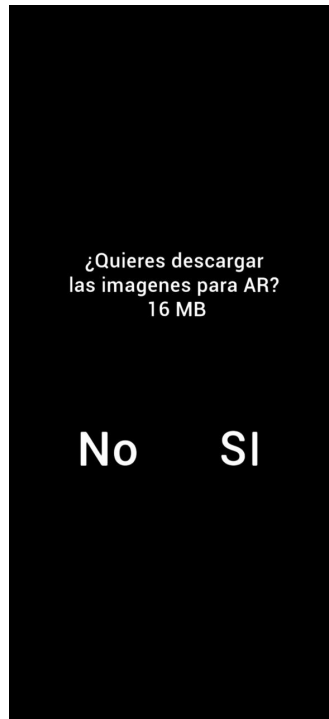


Ilustración 19 Opciones al usuario para descargar las imágenes de reconocimiento RA

Con todas las actualizaciones descargadas y las imágenes guardadas se procede al menú principal de forma normal.

Al iniciar la escena AR se obtiene la *imageLibrary* la cual es la encargada de tener las imágenes a reconocer por el subsistema de realidad aumentada, por lo que se manifiesta que la esta biblioteca va a ser mutable, debido a que los sistemas de AR son estáticos durante toda la sesión (es decir se genera la instancia de la biblioteca por instancia de la app) se checa si ya se habían añadido las imágenes descargadas (si por ejemplo hubiera entrado y salido de la escena de AR sin cerrar por completo al app) en caso de que no se obtienen los nombres de las imágenes a reconocer de la base de datos, por cada uno de estos nombres se carga a memoria la imagen como una *Texture2D* una vez cargada se genera un nuevo *Job* de unity para asignar la textura a la *imageLibrary* asignándole el nombre del assetbundle (recuperado de la base de datos) y el tamaño físico de 20 cm (el aproximado del ancho de una hoja tamaño carta).

En la siguiente página se muestra un diagrama de flujo con el proceso de actualización que realiza la aplicación al iniciar.

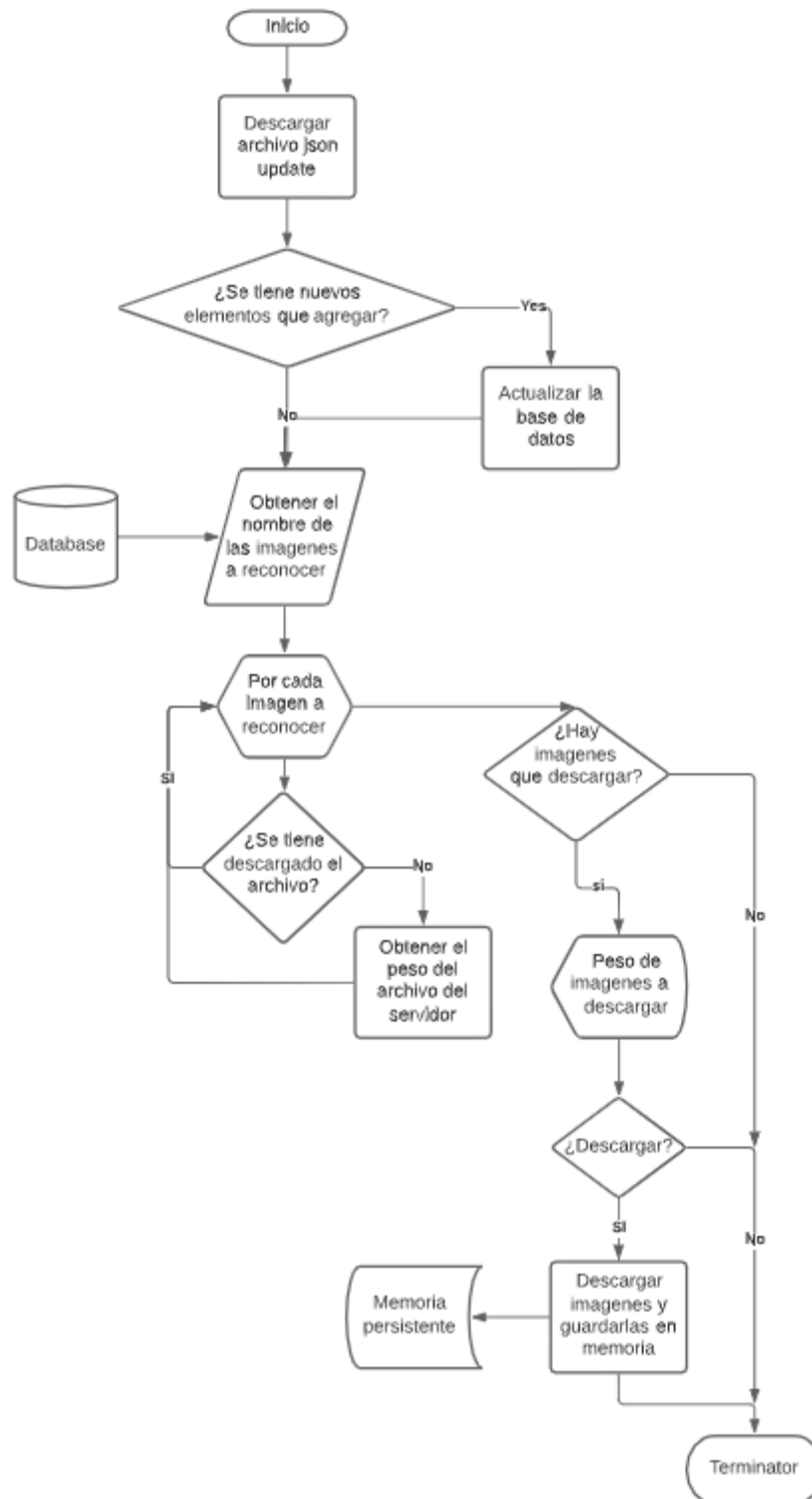


Ilustración 20 Diagrama de Flujo sistema de actualización en la app

Pruebas de Implementación

Escena AR

Dentro de esta sección se verán los resultados obtenidos con la aplicación desarrollada.

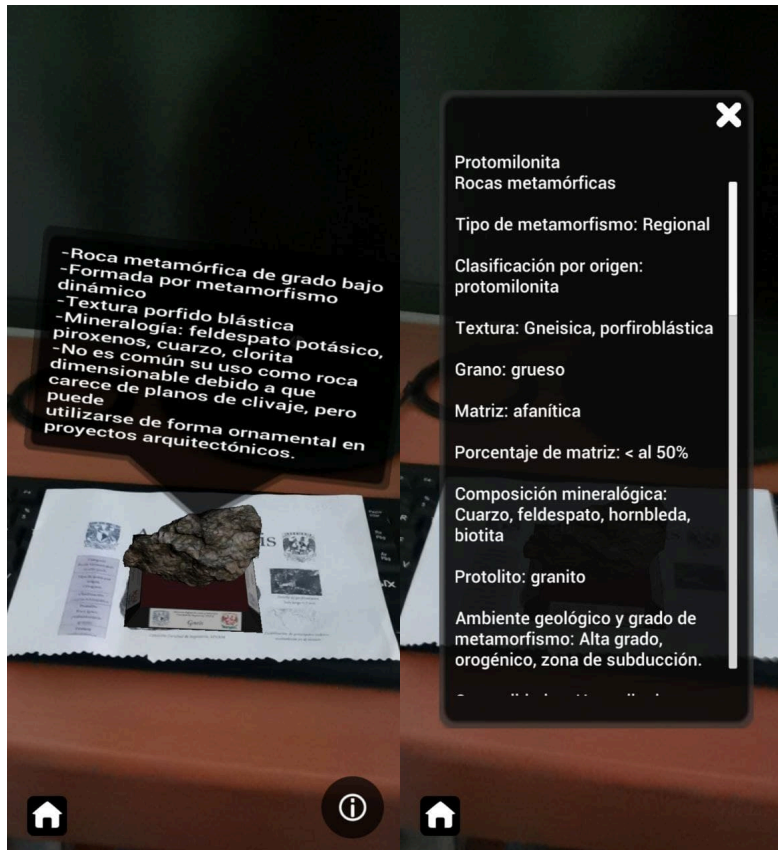


Ilustración 21 Modelo instanciado y detalles del modelo

Como podemos notar en la Ilustración 15 nuestra cámara reconoció la parte del manual de prácticas que contiene la imagen de la roca y que instancia en ese punto el modelo 3D que contiene también los elementos de UI que se le asignaron, que son los datos importantes en una ventana flotante, y una ventana más larga que contiene la información detallada del modelo.

Dentro de esta vista el modelo se puede mover la mover la imagen que se reconoce en el mundo real, se puede rotar el modelo y escalarlo para hacerlo más grande.

Menú Colección

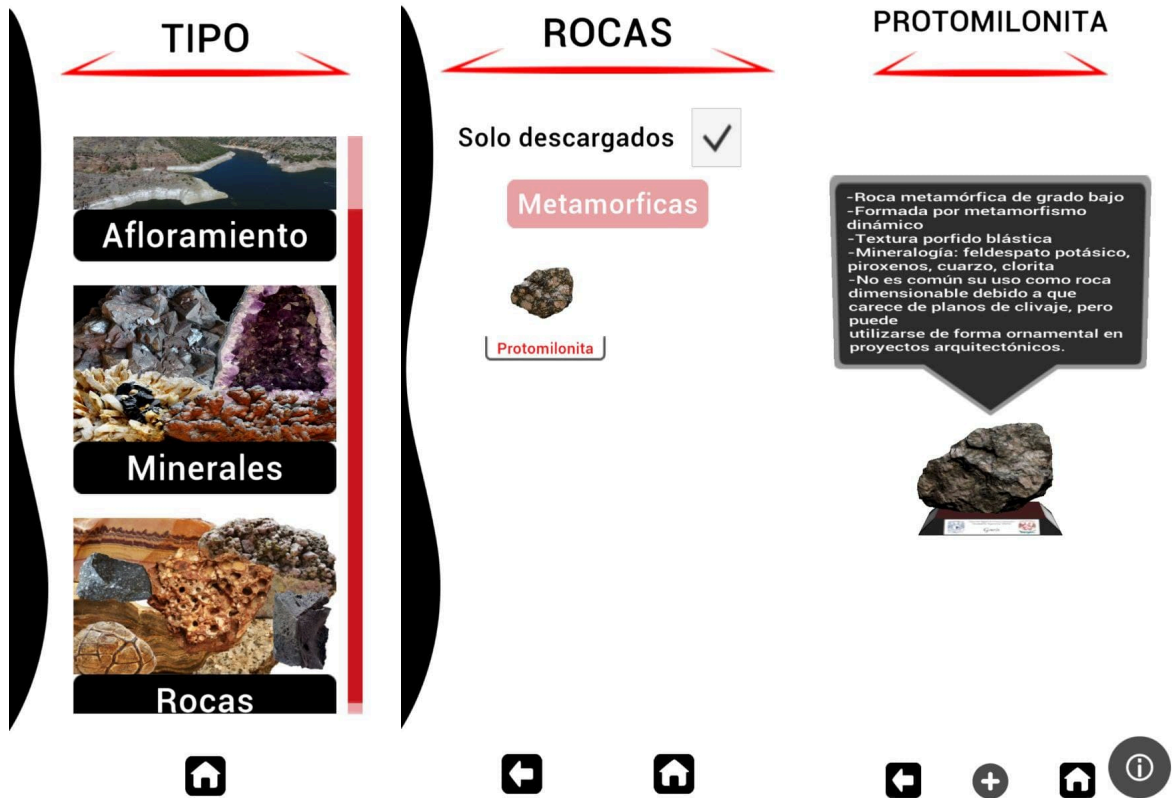


Ilustración 22 Menú de Colección, sección seleccionada, vista del modelo 3D

Dentro de este menú podemos ver las categorías principales de los modelos, los afloramientos, los minerales y las rocas, dentro del submenú de rocas tenemos la opción de filtrar los modelos por solo los descargados, dentro de este submenú se ven las miniaturas incluidas en los AssetBundles. Al hacerle click al modelo (previamente descargado o después de pasar la descarga) se pasa a una escena que contendrá al modelo en una posición fija donde podemos rotarlo y escalarlo.

Menú de descargas



Ilustración 23 Menú de descargas con un elemento descargado

En esta pantalla podemos ver todos los modelos que se pueden descargar desde el servidor en forma de lista ordenada alfabéticamente, también podemos ver el tamaño que estos tienen para tomar la decisión informada sobre si descargarlo o no. En caso de que existiera una actualización de los AssetBundles en este menú es donde se puede actualizar los AssetBundles correspondientes. También podemos eliminar el assetBundle de la memoria del teléfono para poder liberar espacio, en caso de que no se tenga conexión a internet se mostrarán todos los paquetes igual pero el botón tendrá como título *No Action* en caso de que no se tenga descargado el paquete por lo que no hará nada. Debajo del nombre saldrá el tamaño (redondeado hacia arriba) del paquete a descargar en caso de que se tenga conexión a internet

Pruebas con usuarios

Se realizó una prueba con usuarios buscando tener retroalimentación para mejorar la experiencia del usuario con el uso de la aplicación; la prueba se realizó sobre la aplicación en Android, debido a la facilidad de instalación para realizar la prueba con los estudiantes. Dentro de esta prueba notamos que el 57% de los usuarios lograron entrar a la sección RA, teniendo en cuenta que esta sección se encuentra restringida por la compatibilidad de los dispositivos con ARCore no sentimos que

sea un problema de momento, en especial debido a que con cada nueva generación de teléfonos se aumenta sus capacidades en todas las distintas gamas; esperando a que pronto la mayor parte de los dispositivos puedan disfrutar de esta sección sin problemas.

Se tuvieron un par de recomendaciones principales dentro del uso de la aplicación, estas fueron el aumentar la retroalimentación de la app y dar la posibilidad de poder mover el modelo en cualquier dirección dentro del menú del visualizador del menú de colección. Para poder solucionar el primer punto se busca que los botones dentro de la aplicación tuvieran distintas transiciones cuando se presionaran para poder darle la sensación de pulsar, mientras que para la segunda si se tomó la recomendación entera y se modificó el comportamiento original para que el movimiento fuera en cualquier dirección y se pudieran apreciar todos los detalles del modelo.

Como punto extra se busca actualizar la aplicación para agregar una ventana informativa dentro de la sección RA para explicar de forma rápida cómo funciona, explicando al usuario que debe de buscar con la cámara el manual de prácticas (que contiene todas las imágenes que puede reconocer el sistema) para que la aplicación pueda desplegar el modelo 3D del mineral, roca o afloramiento que se esté viendo, y haciéndole notar que puede rotar el modelo y modificar su escala; todo esto debido a que durante las pruebas notamos que a los nuevos usuarios no notaban que podían mover ni escalar los modelos hasta que los tocaban por accidente, al hacer esta modificación podríamos mejorar la experiencia del usuario y dejaríamos que los estudiantes tengan una mejor adaptación al flujo de la app.

Conclusiones

Dentro del desarrollo de la aplicación se tuvieron que aplicar múltiples conocimientos adquiridos gracias al plan de estudios de la carrera de Ingeniería en Computación, dentro de estos podemos notar los conocimientos de administración de proyectos y equipos de desarrollo, la habilidad de investigación y aplicación de nuevas tecnologías, además de la gran capacidad de desarrollo de software con la programación basada en objetos y el uso de bases de datos.

Observando la lista de requerimientos de la aplicación puedo decir que los cumplimos satisfactoriamente, debido a que todas las distintas funcionalidades se lograron integrar de forma natural, además de que durante las pruebas realizadas con un pequeño grupo de control se vio que durante el uso no existió ningún inconveniente con los usuarios que rompiera la experiencia, sin embargo, si existieron mejoras que se tuvieron que realizar, lo cual es bastante natural en el ciclo de vida de las aplicaciones.

Dentro del requisito funcional que más se tuvo que corregir fue la parte de la interacción del usuario con los modelos que se muestran, debido a que se tuvo que hacer notar más que el contenido se podía rotar y escalar, esto se hizo de distintas maneras, primero se introdujo pequeñas animaciones haciendo que los modelos empiecen muy pequeños y que al mostrarse vayan creciendo y rotando hasta llegar a un tamaño predefinido, además de que en el menú de colección se agregó una animación con una mano para indicar al usuario que puede tocar el modelo para rotarlo o escalarlo. Finalmente se tiene el plan de añadir una tarjeta explicativa de la sección RA para dejar bastante claro cómo funciona y dejar explícito que es un sistema de reconocimiento de imágenes.

El requisito más problemático fue la implementación de los modelos en RA dado a que nos vimos limitados a los dispositivos soportados por las tecnologías de ARCore y ARKit, sin poder encontrar una alternativa gratuita que pudiera ampliar la lista de dispositivos que soportaran esta funcionalidad, sin embargo, con la prueba que se realizó vimos que la mayor parte de los usuarios si lograron experimentar esta sección y debido a que esta aplicación se busca que se pueda ir renovando y ampliando con el paso del tiempo la lista de dispositivos se podrá ir ampliando y será más accesible con teléfonos de distintas gamas con el pronóstico que la cantidad de usuarios que no pueden experimentar esta parte de la aplicación disminuya cada año.

Con el ciclo de desarrollo completado, puedo decir que con este tipo de proyecto se espera que los alumnos puedan tener una nueva herramienta que les permita estudiar en cualquier lugar y, con base en esto, puedan tener una mejor experiencia académica y que vean nuevas formas en que se puede implementar las nuevas tecnologías en actividades ya existentes.

Finalmente solo falta reconocer que a pesar de que el proyecto se finalizó y se cumplieron todas las expectativas aún tiene lugares donde se puede mejorar o ampliar la aplicación, por ejemplo, se puede mejorar el sistema de actualización y entrega de contenido al pasarlo a un servicio REST que le pueda proporcionar los datos de la aplicación lo cual nos podría dar control más sencillo para la actualización de los datos que se pueden mostrar en la app o para subir de forma más sencilla las nuevas imágenes a reconocer sin tener un acceso directo al servidor.

Referencias

- [1] Bitfab Blog, Photogrammetry, 3D Scanning with A Photo Camera, <https://bitfab.io/blog/photogrammetry/> [En funcionamiento al momento de la consulta; visitado en Marzo – 2021]
- [2] Unity, About ARFoundation, <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/>, [En funcionamiento al momento de la consulta; visitado en Marzo – 2021]
- [3] Unity, XR image tracking subsystem, <https://docs.unity3d.com/Packages/com.unity.xr.ar subsystems@4.1/manual/image-tracking.html> [en funcionamiento al momento de la consulta; visitado en Marzo – 2021]
- [4] Unity, AR tracked imagen manager; Responding to detected images, <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/tracked-image-manager.html> [en funcionamiento al momento de la consulta; visitado en Marzo – 2021]
- [5] Unity, AssetBundles, <https://docs.unity3d.com/2019.3/Documentation/Manual/AssetBundlesIntro.html> [en funcionamiento al momento de la consulta; visitado en Marzo – 2021]
- [6] Vuforia Pricing, <https://www.ptc.com/en/products/vuforia/vuforia-engine/pricing> [en funcionamiento al momento de la consulta; visitado en Junio – 2021]
- [7] Create your first AR application with Unity and Vuforia, <https://www.itgenerator.com/augmented-reality-app-development/> [en funcionamiento al momento de la consulta, visitado en Junio – 2021]
- [8] Dinosaur 4D+ en la Google Play Store
<https://play.google.com/store/apps/details?id=com.OctagonStudio.ARDinoVR&hl=en&gl=US> [En funcionamiento al momento de la consulta, Octubre – 2021]
- [9] Augmented Reality in Education: A Staggering insight into the future
<https://elearningindustry.com/augmented-reality-in-education-staggering-insight-into-future> [En funcionamiento al momento de la consulta, Octubre – 2021]
- [10] Pagina de descarga Element 4D (ya no se puede descargar)
<https://www.common sense.org/education/app/elements-4d-by-dagri>
[Pagina en funcionamiento al momento de consulta, el producto ya no se puede descargar, Octubre 2021]

[11] Documentación WebXR <https://developers.google.com/ar/develop/webxr> [En funcionamiento al momento de la consulta, Octubre 2021]