Manual Técnico

Geolocalizador Móvil de AR para el Reconocimiento Del campus Universitario.

Presentado por:

Nell Yesid Olaya Calderon

Universidad de Cundinamarca

Facultad de Ingeniería

Programa Tecnología en Desarrollo de Software

Soacha (Cundinamarca)

Mayo 2019

Geolocalizador Móvil de AR para el Reconocimiento Del campus Universitario.

Presentado por:

Nell Yesid Olaya Calderon

Director:

Ing. José del Carmen Ortega

Trabajo Para Obtener El Título de Tecnólogo en Desarrollo de Software

Universidad de Cundinamarca

Facultad de Ingeniería

Tecnología en Desarrollo de Software

Soacha (Cundinamarca)

Mayo 2019

Tabla de contenido

Tabla de Ilustraciones

Introducción

El manual técnico tiene como objetivo describir el diseño del prototipo para la aplicación Unigeo en ambientes móviles. permitirá a cualquier persona que tenga ciertas bases en uso de aplicaciones para dispositivos smartphone realizar la instalación correcta creada para apoyar a la comunidad de la universidad de Cundinamarca sede extensión Soacha a lograr reconocer el campus.

Cabe aclarar que este manual contiene las especificaciones mínimas de hardware y de software para la correcta instalación de la aplicación, en su totalidad la aplicación esta para producción en sistemas operativos Android y modelos de algunos dispositivos móviles.

Herramientas tecnológicas.

Los ambientes móviles ocupan un lugar importante a un mercado que crece ampliamente y los potenciales usuarios cada vez se internan a las nuevas tecnologías que constantemente presentan innovación, esto influye a involucrarse en nuevos paradigmas de programación, herramientas multimedia, modelado de datos, diseño de interfaz, conectividad de las bases de datos, herramientas de seguridad y uso del hardware.

La aplicación es para el sistema operativo Android lo cual en el momento esta limitada a algunos dispositivos y funciona en aquellos que son considerados de una mejor gama debido al recurso que se pueden aprovechar sin tener inconvenientes en la ejecución de la aplicación.

Utilizar el sistema operativo de Android permite que sea mas accesible para los usuarios debido a que la aplicación corre en una plataforma de software libre.

Especificaciones Técnicas

Dispositivos móviles: en la siguiente tabla se especifican los dispositivos que soportan la aplicación.

SAMSUNG	HUAWEI	GOOGLE
Samsung Galaxy A5	Huawei Mate 10	Google Pixel
Samsung Galaxy J5 Pro	Huawei Mate 10 Pro	Google Pixel C
Samsung Galaxy S6	Huawei P10	Google Pixel XL
Samsung Galaxy S6 Edge	Huawei P10 Lite	Google Pixel 2
Samsung Galaxy S6 Edge+	Huawei P20 Lite	Google Pixel 2 XL
Samsung Galaxy A7	Huawei P20	Google Pixel 3
Samsung Galaxy J7 Pro	Huawei P20 Pro	Google Nexus 6P
Samsung Galaxy S7	LG	MOTOROLA
Samsung Galaxy S7 Edge – Exynos	LG V30+	Motorola Moto G4
Samsung Galaxy S7 Edge – Snapdragon	LG V35 ThinQ	ASUS
Samsung Galaxy A8+	LG V40 ThinQ	ASUS ZenFone AR
Samsung Galaxy S8	LG G6	OTRAS
Samsung Galaxy S8+	LG G7/G7+	OPPO R11, R11
Samsung Galaxy S9	SONY	OPPO R11, R11t
Samsung Galaxy S9+	Sony Xperia Z5	ONEPLUS
Samsung Galaxy Note 5 – Exynos	Sony Xperia XZ	ONEPLUS 3 (A3000)
Samsung Galaxy Note 8	XIAOMI	ONEPLUS 5 (A5000)
Samsung Galaxy Note 9	Xiaomi Redmi 3S	
Galaxy Tab S3 9.7	Xiaomi Redmi Note 3 – Snapdragon	
Galaxy Tab Active2		

Ilustración 1: Dispositivos compatibles

Entorno de desarrollo (IDE): UNITY 3D Versión Unity 2018.1.0f2 (64-bit), Mono DEVELOP.

Lenguaje de programación: C# o C SHARP.

kit de desarrollo de software (SDK): Maps SDK para unity, Vuforia SDK Android, SDK Tools.

Java Development Kit (JDK): Java SE Development Kit 8u211 Windows x64.

Herramientas Utilizadas Para el Desarrollo.

Vuforia

Vuforia es un SDK que permite construir aplicaciones basadas en la Realidad Aumentada, en donde se entrelazan elementos del mundo real con elementos virtuales por medio de la cámara del dispositivo que se utilice.

C# o C Sharp

Actualmente uno de los lenguajes de programación más populares. El objetivo de Microsoft, que tiene todos los derechos sobre la plataforma de desarrollo .NET Framework en la que está incluido el lenguaje C#, es permitir a los programadores abordar el desarrollo de aplicaciones complejas con facilidad y rapidez.

Unity

Es un motor de videojuegos multiplataforma que permiten una edición e iteración rápida en el ciclo de desarrollo de proyectos. Este se apoya tanto en el desarrollo de la tecnología 2D como el de la 3D con prestaciones y funcionalidades para las necesidades del desarrollador logrando crear una experiencia de juego sumamente realista y de alto rendimiento.

Instalación de aplicaciones

Para implementar de manera correcta Unity 3d primero se deben descargar algunos programas que consigo traen kits de desarrollo de software que son de suma importancia.

Para el sistema operativo Windows la instalación de programas es sencilla no requiere de comandos como otros sistemas. únicamente de permisos de administrador por lo cual siempre que se ejecute un archivo se deberán contar con estos permisos.

Instalación de Android

Para descargar Android studio primero se debe ingresar a la página oficial y descargar el programa.

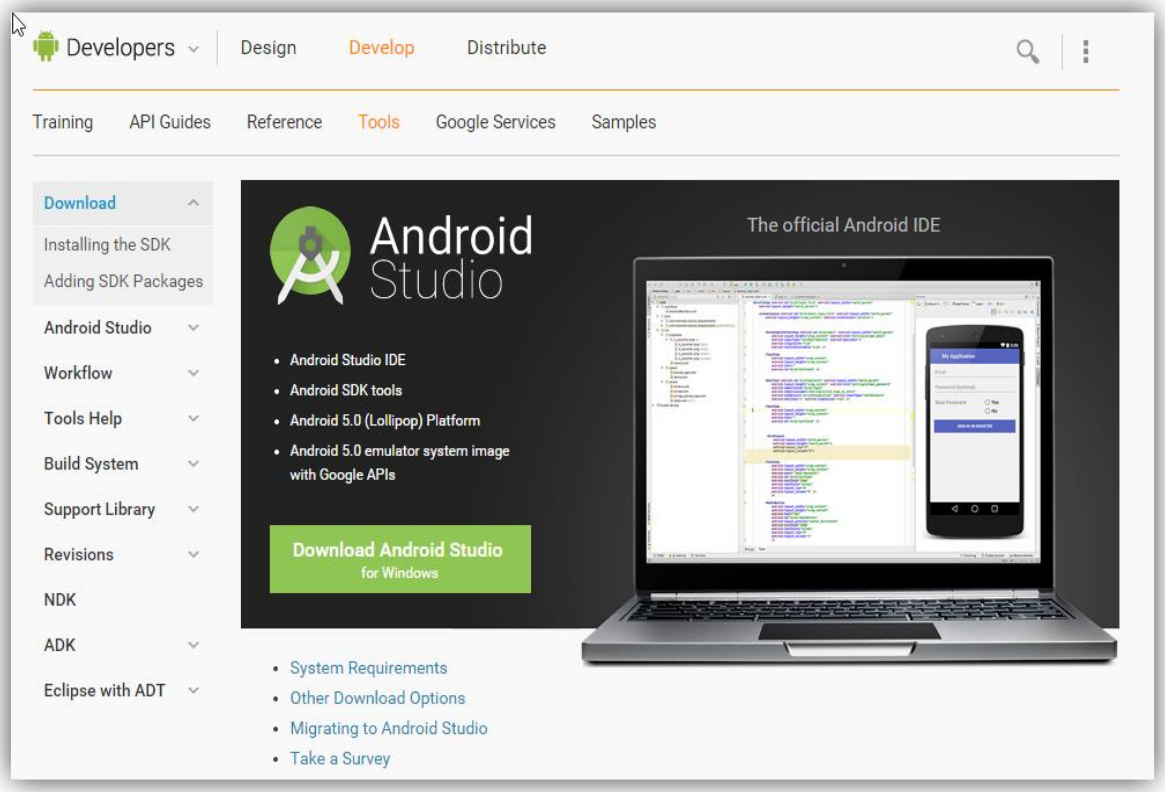

Ilustración 2: página oficial de Android Studio

Ilustración 3: Asistente de instalación Android Studio

Una vez descargado el archivo se debe ejecutar el asistente de instalación pide continuar para realizar la instalación de Android studio.

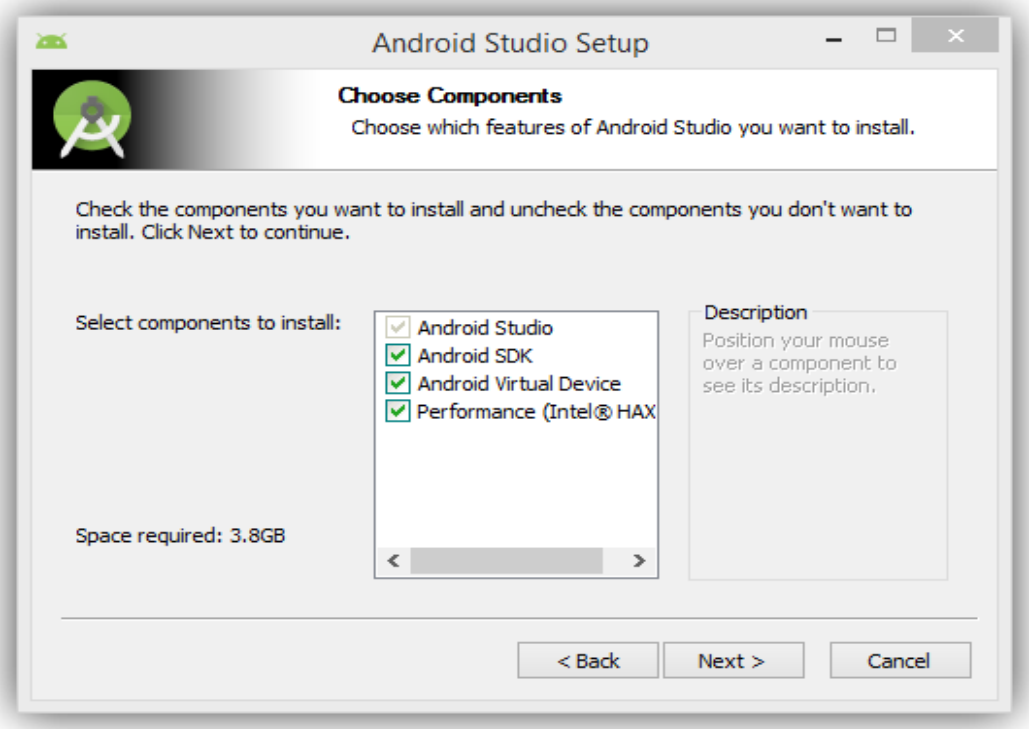

Ilustración 4: Instalación Android Studio

En este paso es importante seleccionar todos para que todas las funciones estén disponibles.

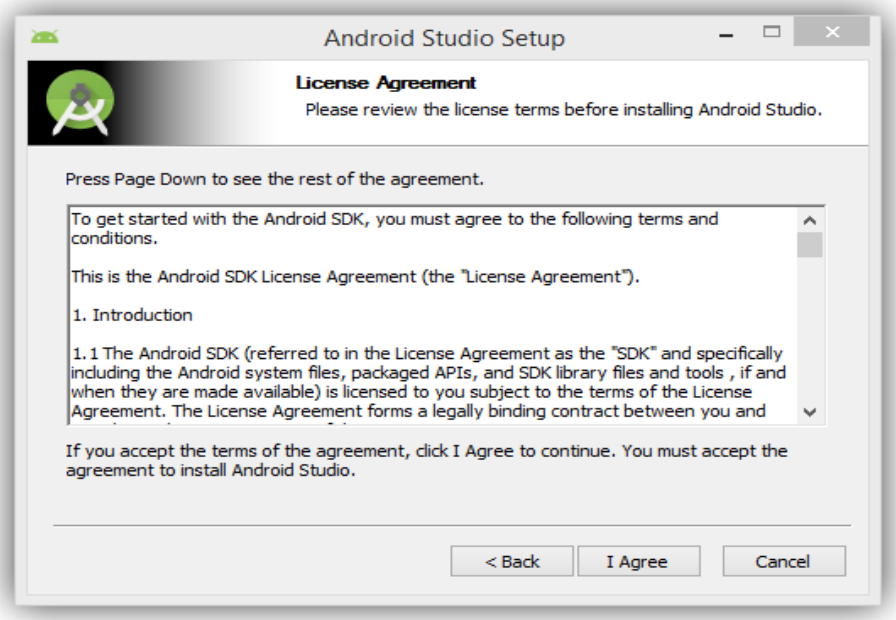

Ilustración 5: Aceptar Términos y condiciones Android studio para instalación.

Es necesario estar de acuerdo con los términos y condiciones para continuar.

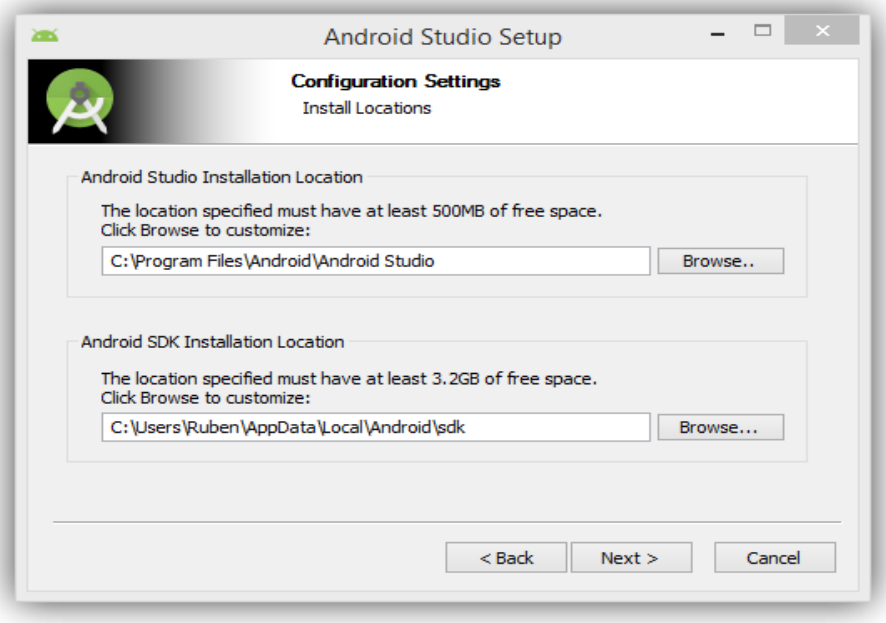

Ilustración 6: Escogiendo ruta para instalación de Android Studio.

Se selecciona el destino donde quedara instalado el programa y continuar con la instalación.

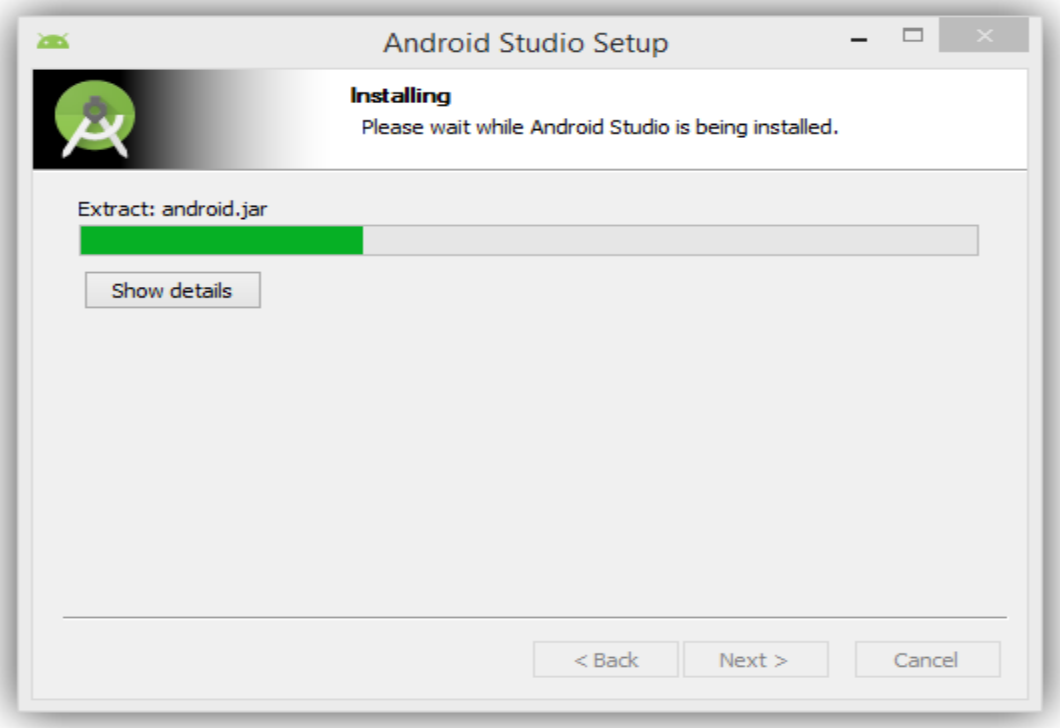

Ilustración 7: Extrayendo repositorios e instalando recursos para el sistema..

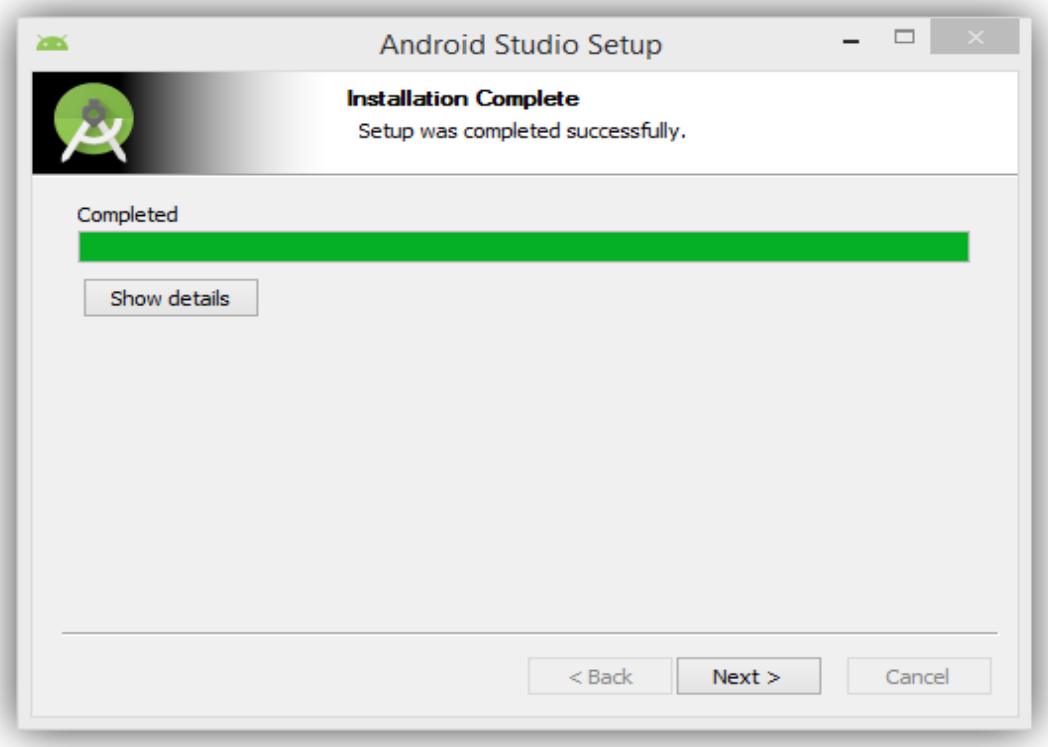

Ilustración 8: Instalación Exitosa.

Una vez finalizada la instalación se puede descargar unity 3d.

Instalación de Unity

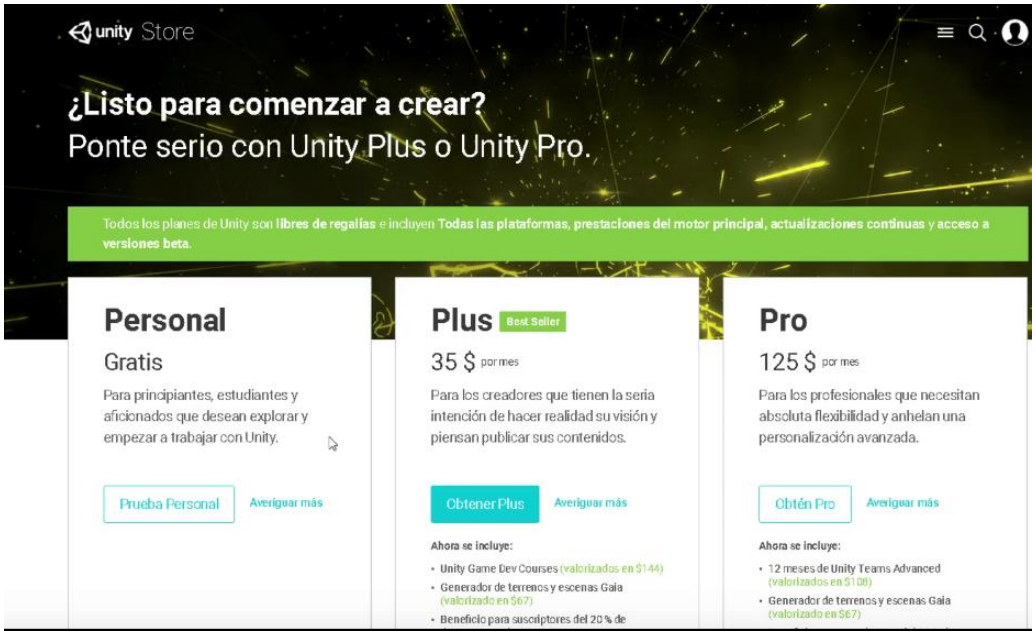

Ilustración 9:Página oficial Unity 3D

Se ingresa a la página oficial y se selecciona la opción personal después se debe buscar la versión Unity 2018.1.0 y se descarga el instalador de unity.

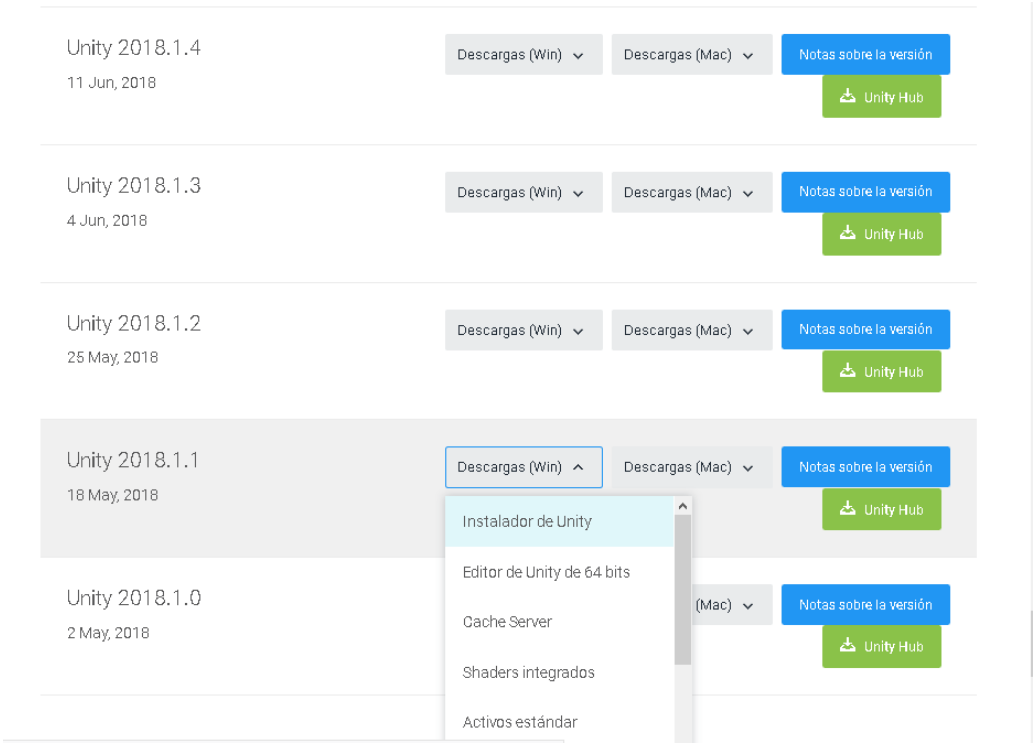

Ilustración 10: Versiones de UNITY.

Una vez descargado el archivo se ejecuta y aceptan condiciones de uso.

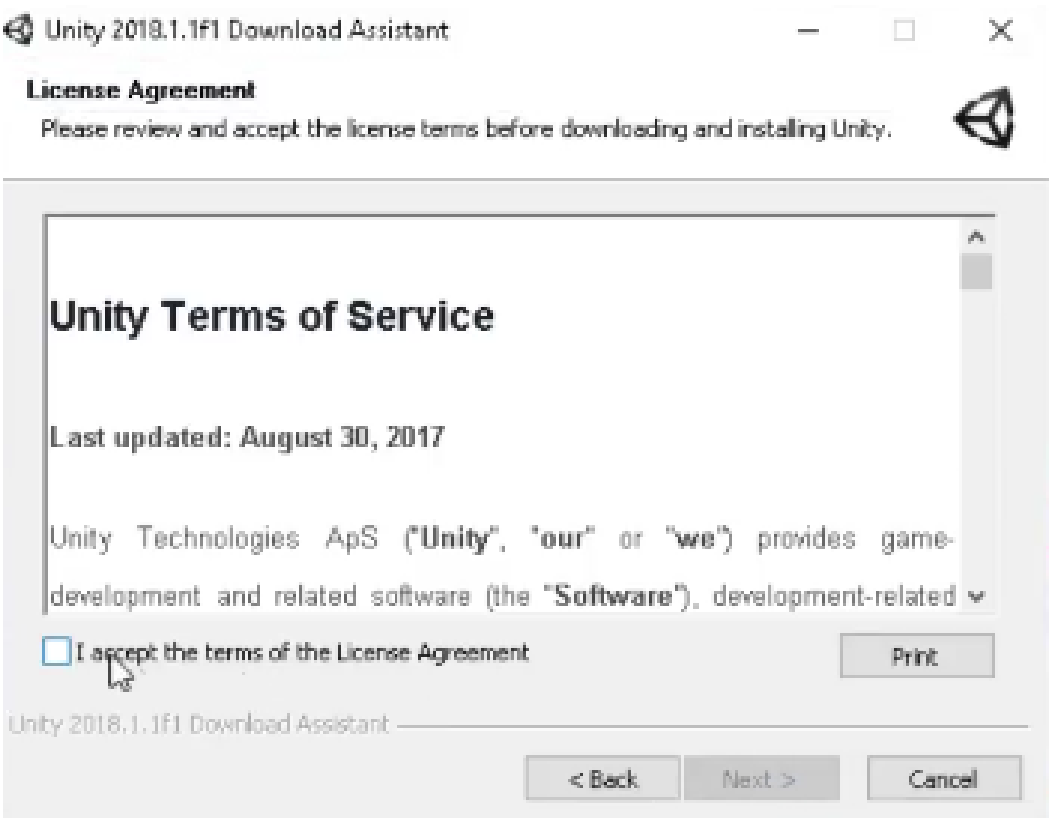

Ilustración 11: Paso 1 instalación Unity.

Se deben seleccionar todas las casillas que están en la imagen esto permitirá hacer uso de las funciones que más se utilizarían para el desarrollo de la aplicación.

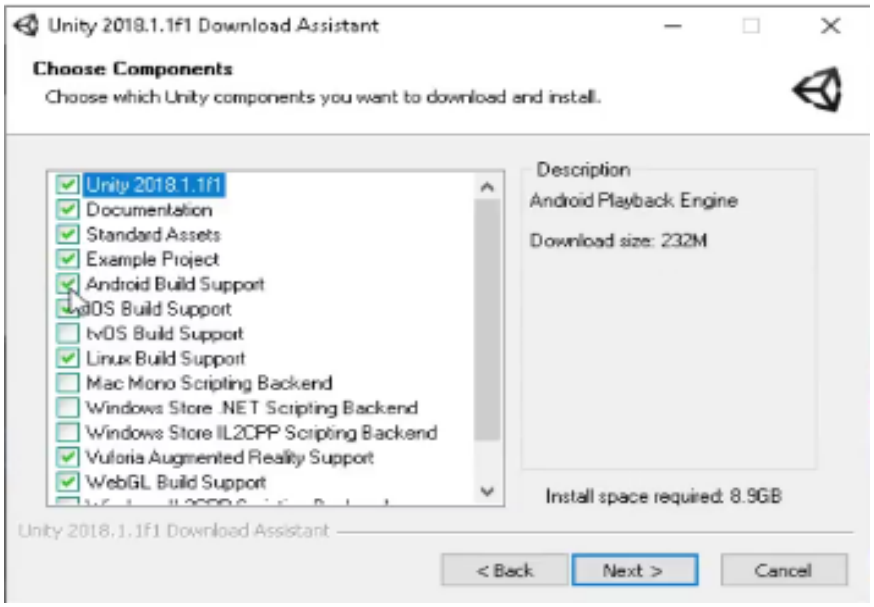

Ilustración 12: Paso dos el más importante.

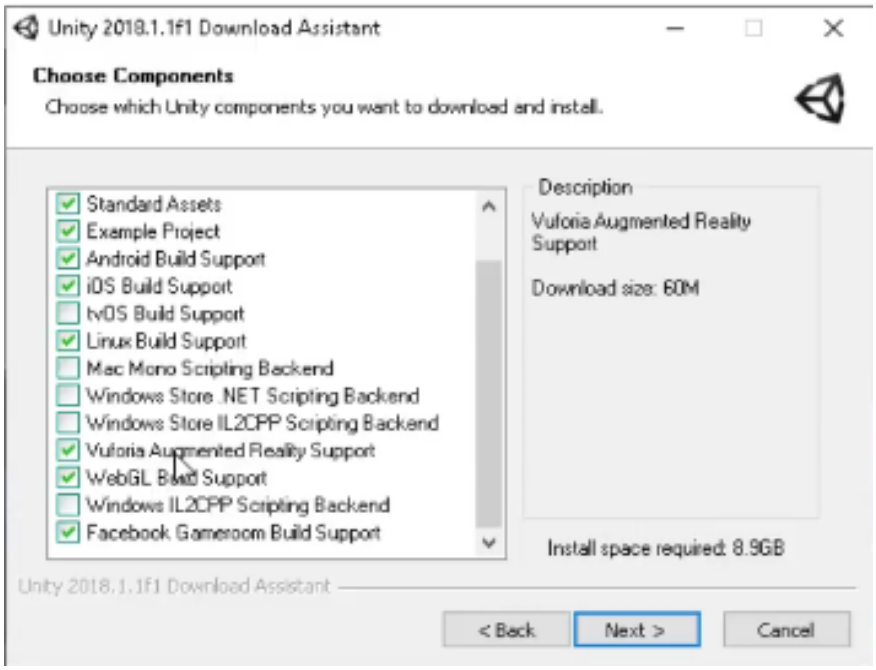

Ilustración 13: Paso2 confirmando funciones a instalar.

Estas opciones incluyen el SDK de vuforia lo cual permite redireccionar y descargar repositorios automáticamente.

Configuración de aplicaciones

Para que las aplicaciones cumplan con todas las funciones se debe configurar desde el motor de unity accediendo directamente a las rutas de estas.

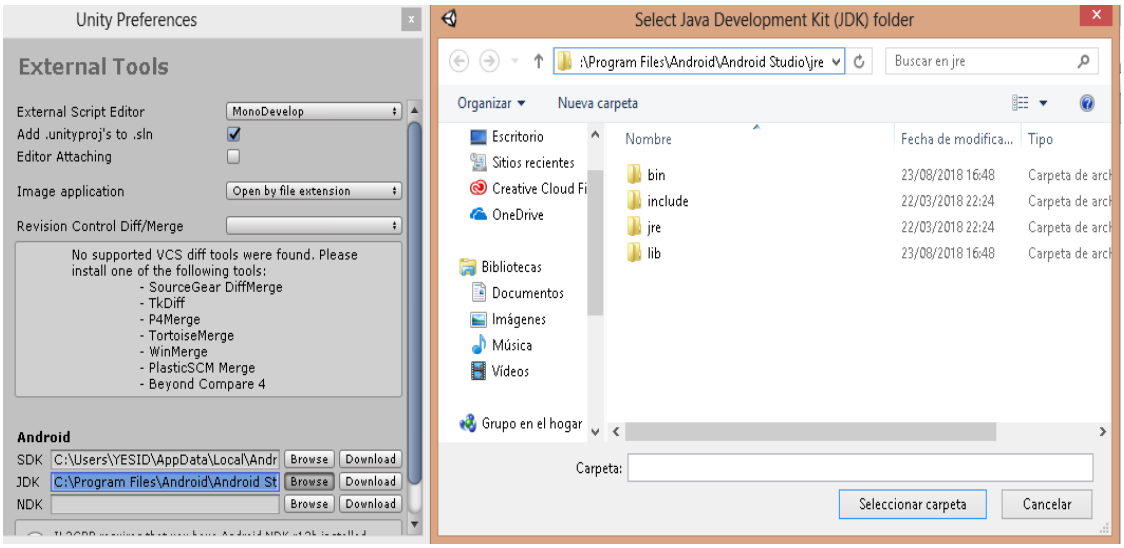

Ilustración 14: Configuración de Unity Con JDK.

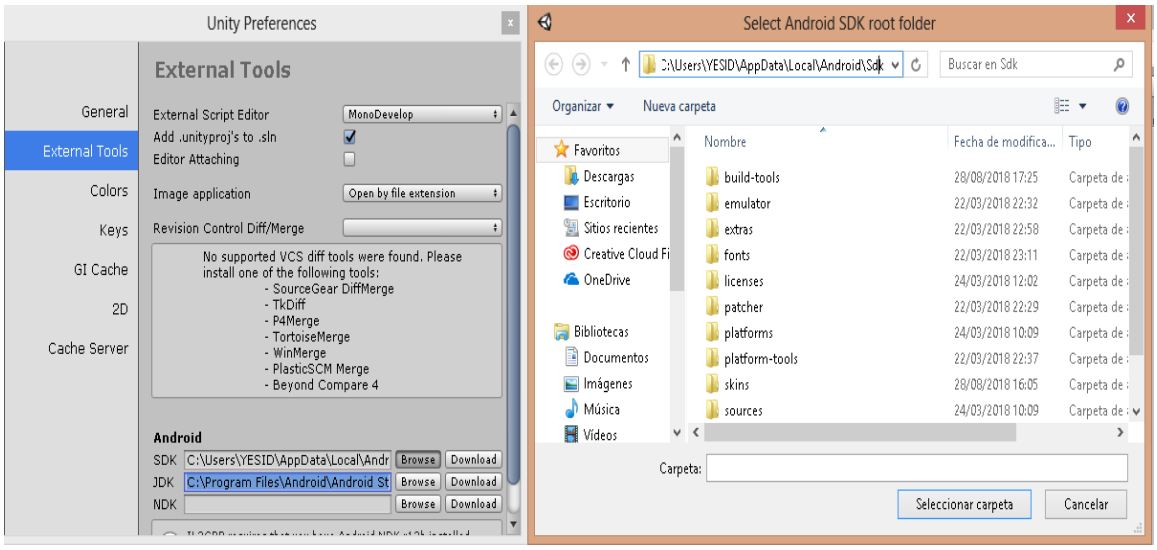

Ilustración 15: Configuración de UNITY con SDK Android.

Funcionamiento de la aplicación.

En los siguientes Scripts se explica solo lo esencial para que funcione la aplicación, por ejemplo: Ubicación de objetos dentro del mapa, activación de realidad aumentada, navegación entre escenarios y GPS.

Funcionamiento del sistema de objetos dentro del campus.

Este sistema por medio de coordenadas permite ubicar un punto específico dentro del mapa del campus universitario.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using GoShared;

namespace Unigeo {

    public class GODropFeatures : MonoBehaviour {

        public GOMap Unigeo ;

        public Material testLineMaterial;
        public GameObject player;
        public Material testPolygonMaterial;
        public GOUVMappingStyle uvMappingStyle = GOUVMappingStyle.TopFitSid
esRatio;
        // Iniciando el sistema
        IEnumerator Start () {

            //configuración del origen.
            yield return StartCoroutine (goMap.locationManager.WaitForOrigi
nSet ());
```

```
]

            // Soltar un punto en el mapa

            dropTestPin();

            Laboratorio ();

            cancha ();

            dropTestLine();

            dropTestLine2();

            //eliminar a polygon
            dropTestPolygon();

            arboles ();

            EntradaBC();

        }

        void EntradaBC(){

            GameObject EntradaBC = GameObject.FindGameObjectWithTag("Entrad
aBC");
            EntradaBC.transform.localScale = new Vector3 (12,6,13);

            Coordinates coordenadasEntradaBC = new Coordinates (4.578267f,
-74.223334f);
            goMap.dropPin(coordenadasEntradaBC,EntradaBC);

        }


        void arboles(){

            GameObject arboles = GameObject.FindGameObjectWithTag("arboles"
);
            arboles.transform.localScale = new Vector3 (6,6,6);


            //2)  clase de coordenadas con la latitud deseada

            Coordinates coordenadasarboles = new Coordinates (4.577283f,-
74.223884f);

    //3) llama al pasador desplegable que pasa las coordenadas y objeto de
juego
            goMap.dropPin(coordenadasarboles,arboles);



        }


        void cancha(){
```

```
            GameObject cancha = GameObject.FindGameObjectWithTag("cancha");
            cancha.transform.localScale = new Vector3 (6,6,6);
            //aBigRedSphere2.GetComponent<MeshRenderer> ().material.color =
 Color.green;


            //2)  clase de coordenadas con la latitud deseada

            Coordinates coordenadascancha = new Coordinates (4.577556f,-
74.224328f);

            //3) llama al pasador desplegable que pasa las coordenadas y tu
objeto de juego

            goMap.dropPin(coordenadascancha,cancha);



        }

        void Laboratorio(){

            GameObject aBigRedSphere3 = GameObject.FindGameObjectWithTag("O
bjeto2");
            aBigRedSphere3.transform.localScale = new Vector3 (6,6,6);



            Coordinates coordinates3 = new Coordinates (4.577959f,-
74.224281f);


            goMap.dropPin(coordinates3,aBigRedSphere3);

        }

        void dropTestPin() {

            GameObject aBigRedSphere2 = GameObject.FindGameObjectWithTag("O
bjeto1");
            aBigRedSphere2.transform.localScale = new Vector3 (6,6,6);


            Coordinates coordinates2 = new Coordinates (4.578611f,-
74.223347f);


            goMap.dropPin(coordinates2,aBigRedSphere2);

        }

        void dropTestLine() {


            List <Coordinates> polyline = new List<Coordinates> ();
            polyline.Add(new Coordinates (4.578488,-74.222587));
            polyline.Add(new Coordinates (4.578541,-74.222658));
            polyline.Add(new Coordinates (4.578610,-74.222717));
            polyline.Add(new Coordinates (4.578678,-74.222776));
```

```
            // punto de y cruce
            polyline.Add(new Coordinates (4.578773,-74.222865));
            polyline.Add(new Coordinates (4.578858,-74.222954));
            polyline.Add(new Coordinates (4.578916,-74.223019));
            polyline.Add(new Coordinates (4.578951,-74.223110));
            polyline.Add (new Coordinates (4.578959, -74.223205));
            polyline.Add (new Coordinates (4.578959, -74.223256));
            polyline.Add(new Coordinates(4.578954, -74.224536));



            float width = 4;


            float height = 3;


            Material material = testLineMaterial;


            goMap.dropLine(polyline,width,height,material,uvMappingStyle);

        }

        void dropTestLine2() {


            List <Coordinates> polyline = new List<Coordinates> ();
            polyline.Add(new Coordinates (4.578859,-74.222974));
            polyline.Add(new Coordinates (4.578780,-74.223062));
            polyline.Add(new Coordinates (4.578720,-74.223164));
            polyline.Add(new Coordinates (4.578701,-74.223200));
            polyline.Add(new Coordinates (4.578687,-74.223226));
            polyline.Add(new Coordinates (4.578679,-74.223256));
            polyline.Add(new Coordinates (4.578669,-74.223266));
            polyline.Add(new Coordinates (4.578659,-74.223276));
            polyline.Add(new Coordinates (4.578639,-74.223276));
            polyline.Add(new Coordinates (4.578530,-74.223220));
            polyline.Add(new Coordinates(4.5784290, -74.223145));


            //2) Establecer ancho de linea
            float width = 4;

            //3) Establecer la altura de la linea
            float height = 3;

            //4) Elija un material para la línea (esta vez
vinculamos el material del inspector)

              Material material = testLineMaterial;

              //5) Lnea de llamada
              goMap.dropLine(polyline,width,height,material,uvMa
ppingStyle);                    }

         void dropTestPolygon() {
```

```
        }

    }
}
```

Funcionamiento de vuforia para enfocar Objetos.

Para el uso de Realidad Aumentada vuforia es el participante debido a que aprovechara al máximo los recursos de cámara del dispositivo móvil para que se cumpla el objetivo se implementara un auto enfoque a través del siguiente Script.

```
using UnityEngine;
using System.Collections;
using Vuforia;

public class CameraFocusController : MonoBehaviour {


    private bool mVuforiaStarted = false;
    private bool mFlashEnabled = false;

    void Start()
    {
        VuforiaARController vuforia = VuforiaARController.Instance;

        if (vuforia != null)
            vuforia.RegisterVuforiaStartedCallback(StartAfterVuforia);
    }

    private void StartAfterVuforia()
    {
        mVuforiaStarted = true;
        SetAutofocus();
    }

    void OnApplicationPause(bool pause)
    {
        if (!pause)
        {
            // La aplicación se reanudó

            if (mVuforiaStarted)
            {
                // La aplicación se reanudó y vuforia ya comenzó

                // vamos a empezar de nuevo

                SetAutofocus(); // Esto se hace porque algunos dispositivos
Android pierden el enfoque automático después de reanudar
                // esto fue un error en Vuforia 4 y 5. No he marcado 6,
pero el código es inofensivo de todos modos
```

```
            }
        }
    }

    private void SetAutofocus()
    {
        if (CameraDevice.Instance.SetFocusMode(CameraDevice.FocusMode.FOCUS
_MODE_CONTINUOUSAUTO))
        {
            Debug.Log("Autofocus set");
        }
        else
        {
            // Nunca he visto un dispositivo que no admita esto, pero por
si acaso

              Debug.Log("this device doesn't support auto focus"
);
         }
     }
}
```

Función de los objetos.

Cada objeto en la aplicación es un **GameObject**, desde el dispositivo ubicado y objetos hasta el escenario, cámara y diseños 3d. este necesita de propiedades antes de que pueda convertirse en parte de la escena, un entorno o un efecto especial en la aplicación recreara ante el usuario como se ven distribuidos los salones mapas y caminos para seguir.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AparecerPiso : MonoBehaviour {

// son los objetos de juego

    public GameObject PisoReconocer1;
    public GameObject Planopiso1;
    public GameObject PisoReconocer2;
    public GameObject Planopiso2;
    public GameObject Opciones;
    public GameObject ImagenPlanopiso1;
```

```
    public GameObject ImagenPlanopiso2;

// cada evento lo podre llamar desde un botón predeterminado e
n unity

    public void Evento(){

        PisoReconocer1.SetActive (true);
        Planopiso1.SetActive(true);
        Opciones.SetActive (false);

    }
    public void Evento2(){

        PisoReconocer2.SetActive (true);
        Planopiso2.SetActive(true);
        Opciones.SetActive (false);

    }

    public void PlanoPiso1(){

        ImagenPlanopiso1.SetActive (true);

    }

    public void PlanoPiso2(){

        ImagenPlanopiso2.SetActive (true);
    }

    public void CerrarPlanoPiso1(){

        ImagenPlanopiso1.SetActive (false);

    }
    public void CerrarPlanoPiso2(){

        ImagenPlanopiso2.SetActive (false);

    }
}
```

Servicios de localización.

Recuperado a través de Input.location.lastData. El servicio no comienza a enviar datos de ubicación inmediatamente. El código debe verificar el estado de servicio actual en Input.location.status. Indicadores de precisión deseados: precisión de servicio deseada en medidores. El uso de valores más altos, como 500, generalmente no requiere encender el chip GPS y, por lo tanto, ahorra energía de la batería. Se podrían usar valores como 5-10 para obtener la mejor precisión. El valor por defecto es de 10 metros. updateDistanceInMeters: la distancia mínima (medida en metros) que un dispositivo debe mover lateralmente antes de que se actualice la propiedad Input.location. Los valores más altos como 500 implican menos sobrecarga. El valor predeterminado es de 10 metros. En Android, el uso de este método en sus scripts agregará automáticamente el permiso ACCESS_FINE_LOCATION al manifiesto de Android. Si usa valores de baja precisión, como 500 o más, puede seleccionar "Ubicación de baja precisión" en la Configuración del jugador para agregar el permiso ACCESS_COARSE_LOCATION.

```
using UnityEngine;
using System.Collections;

public class TestLocationService : MonoBehaviour
{
    IEnumerator Start()
    {
        // Primero, verifica si el usuario tiene habilitado el
servicio de localización


        if (!Input.location.isEnabledByUser)
            yield break;

        // Iniciar el servicio antes de consultar la ubicación

        Input.location.Start();

        // Espere hasta que se inicie el servicio
```

```
        int maxWait = 20;
        while (Input.location.status == LocationServiceStatus.Initializing
&& maxWait > 0)
        {
            yield return new WaitForSeconds(1);
            maxWait--;
        }

        // El servicio no se inicializó en 20 segundos

        if (maxWait < 1)
        {
            print("Timed out");
            yield break;
        }

        // Conexión fallida
        if (Input.location.status == LocationServiceStatus.Failed)
        {
            print("Unable to determine device location");
            yield break;
        }
        else
        {
            // Acceso concedido y valor de ubicación puede ser recuperados

              print("Location: " + Input.location.lastData.latit
ude + " " + Input.location.lastData.longitude + " " + Input.lo
cation.lastData.altitude + " " + Input.location.lastData.horiz
ontalAccuracy + " " + Input.location.lastData.timestamp);
         }

         // Stop service if there is no need to query location
updates continuously
         Input.location.Stop();
    }
}
```

```
using System.Collections.Generic;
using UnityEngine;

public class GamePersitente : MonoBehaviour {

    public static GamePersitente Piso1B;
    public loadlevels loadlevela;
    public int ObjetoA = 3;
    public GameObject BloqueA;
    public GameObject BloqueB;
    public GameObject BloqueC;
```

```
    void Awake(){


        if (Piso1B == null) {

            Piso1B = this;
            DontDestroyOnLoad (gameObject);
            Debug.Log ("Soy el primero!");
            loadlevela = GetComponent<loadlevels> ();

            Traerdatos();


        }else if(Piso1B != this){

            Destroy (gameObject);
            Debug.Log ("yA EXISTE UN OBJETO Piso1B, Autodrestr
uir");
            loadlevela = GetComponent<loadlevels> ();

            //Traerdatos();


        }
    }

    public void Traerdatos(){

        //loadlevela = GetComponent<loadlevels> ();

        if (loadlevela != null && loadlevela.resultado == 3) {

            Debug.Log ("Cargando escenario con valor de
retorno 3");
            ObjetoA = 5;

        } else {

            Debug.Log ("Cargando escenario con valor de
retorno 5"+ObjetoA);
        }
    }
}
```

Funcionamiento del sistema de Ground plane de Vuforia.

```
/*==============================================================
All Rights Reserved.
Confidential and Proprietary - Protected under copyright and o
ther laws.
==============================================================
===============*/

using UnityEngine;
using System.Collections;
using Vuforia;

/// <summary>
/// A custom handler that implements the ITrackableEventHandle
r interface.
///
/// Changes made to this file could be overwritten when upgrad
ing the Vuforia version.
/// When implementing custom event handler behavior, consider
inheriting from this class instead.
/// </summary>
public class DefaultTrackableEventHandler : MonoBehaviour, ITr
ackableEventHandler
{
    public GameObject CuadrodeMira;
    public GameObject Ventana;
    #region PROTECTED_MEMBER_VARIABLES

    protected TrackableBehaviour mTrackableBehaviour;

    #endregion // PROTECTED_MEMBER_VARIABLES

    #region UNITY_MONOBEHAVIOUR_METHODS

    protected virtual void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>
();
        if (mTrackableBehaviour)
            mTrackableBehaviour.RegisterTrackableEventHandler(
this);
    }

    protected virtual void OnDestroy()
    {
        if (mTrackableBehaviour)
            mTrackableBehaviour.UnregisterTrackableEventHandle
r(this);
    }
```

```
    #endregion // UNITY_MONOBEHAVIOUR_METHODS

    #region PUBLIC_METHODS

    /// <summary>
    ///     Implementation of the ITrackableEventHandler funct
ion called when the
    ///     tracking state changes.
    /// </summary>
    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TR
ACKED)
        {
            Debug.Log("Trackable " + mTrackableBehaviour.Track
ableName + " found");
            OnTrackingFound();
            Ventana.SetActive(true);
            CuadrodeMira.SetActive(false);

        }
        else if (previousStatus == TrackableBehaviour.Status.T
RACKED &&
            newStatus == TrackableBehaviour.Status.NO_POSE)
        {
            Debug.Log("Trackable " + mTrackableBehaviour.Track
ableName + " lost");
            OnTrackingLost();



        }
        else
        {
            // For combo of previousStatus=UNKNOWN + newStatus
=UNKNOWN|NOT_FOUND
            // Vuforia is starting, but tracking has not been
lost or found yet
            // Call OnTrackingLost() to hide the augmentations
            OnTrackingLost();
        }
    }

    #endregion // PUBLIC_METHODS
```

```
    #region PROTECTED_METHODS

    protected virtual void OnTrackingFound()
    {
        var rendererComponents = GetComponentsInChildren<Rende
rer>(true);
        var colliderComponents = GetComponentsInChildren<Colli
der>(true);
        var canvasComponents = GetComponentsInChildren<Canvas>
(true);


        // Enable rendering:
        foreach (var component in rendererComponents)
            component.enabled = true;


        //Debug.Log("No se ha iniciado");

        // Enable colliders:
        foreach (var component in colliderComponents)
            component.enabled = true;
        //Debug.Log("No se ha iniciado");

        // Enable canvas':
        foreach (var component in canvasComponents)
            component.enabled = true;
        //Debug.Log("No se ha iniciado");
    }


    protected virtual void OnTrackingLost()
    {
        var rendererComponents = GetComponentsInChildren<Rende
rer>(true);
        var colliderComponents = GetComponentsInChildren<Colli
der>(true);
        var canvasComponents = GetComponentsInChildren<Canvas>
(true);

        // Disable rendering:
        foreach (var component in rendererComponents)
            component.enabled = false;
        Debug.Log("No se ha iniciado el render");

        // Disable colliders:
        foreach (var component in colliderComponents)
            component.enabled = false;
```

```
            Debug.Log("No se ha iniciado el collider");

            // Disable canvas':
            foreach (var component in canvasComponents)
                component.enabled = false;
            Debug.Log("No se ha iniciado el canvas");
        }
        #endregion // PROTECTED_METHODS
}
```

Prototipos de pantalla de aplicativo.

Ilustración 16:Inicio app Unigeo.

Ilustración 17: Menú de navegación.

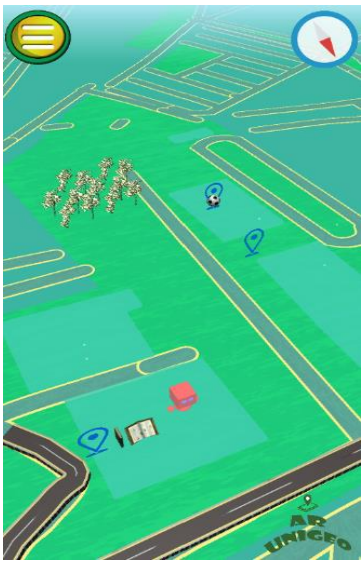

Ilustración 18: Navegación primer interfaz.

Ilustración 19: Busqueda de puntos.

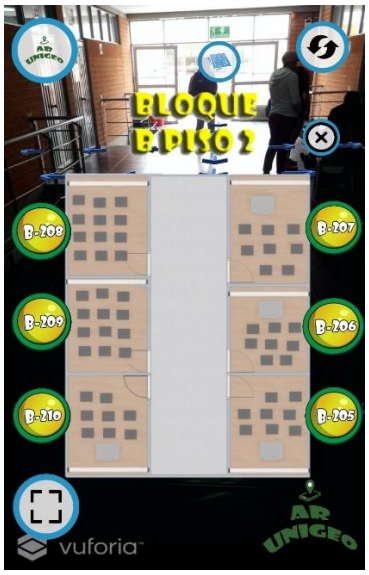

Ilustración 20: Mapa de navegación Bloque B piso2.

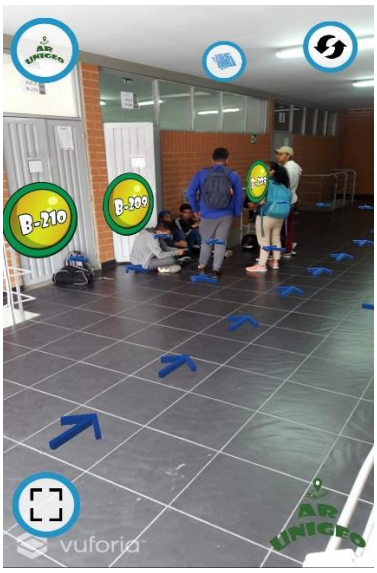

Ilustración 21: Navegación Interior con REALIDAD AUMENTADA.

Ilustración 22: Selección de piso.

Ilustración 23: Ubicación de Sitios SALA 2 Y 3 Bloque A.