

Aplicación móvil interactiva para Android y en lenguaje de programación java que haga uso de la realidad aumentada para el mejoramiento, innovación y dinamización de la enseñanza de redes locales (LAN).

Karen Mayerly Macias Garcia

Universidad Nacional Abierta Y A Distancia
Escuela De Ciencias Básicas Tecnología E Ingeniería
Ingeniería De Sistemas
Junio 2021

Aplicación móvil interactiva para Android y en lenguaje de programación java que haga uso de la realidad aumentada para el mejoramiento, innovación y dinamización de la enseñanza de redes locales (lan).

Karen Mayerly Macias Garcia

Proyecto aplicado presentada(o) como requisito parcial para optar al título de:

Ingeniera de sistemas

Director (a):

Ingeniero de sistemas Jaime Junior Sedano Pinzon

Universidad Nacional Abierta Y A Distancia

Escuela De Ciencias Básicas Tecnología E Ingeniería

Ingeniería De Sistemas

Junio 2021

Resumen

La forma en que se relacionan las personas ha ido cambiando con el paso del tiempo; hoy se cuenta con herramientas que antes no se tenían; lo que ha facilitado que se interactúe con personas que se encuentran en otras partes del planeta. Ahora bien, para poder acceder a estas herramientas es necesario contar con una red que suministre el acceso a internet; esto a su vez redundaría en la necesidad de contar con personas expertas que realicen el diseño y la estructuración de dicha red y así alcanzar los objetivos planteados inicialmente. Por ese motivo, este proyecto busca facilitar el proceso de diseño y estructuración de una red local, y no sólo esto, sino que brindará al usuario la seguridad de diseñar esta red correctamente, aprovechando al máximo los recursos con los que cuente en el área que desea implementar la red. Además, el presente estudio tiene como objeto hacer uso de la realidad aumentada y lenguajes de programación de alto nivel como Java e implementación de librerías OpenCV para la creación de una aplicación interactiva de diseño y estructuración de redes LAN. Durante el desarrollo se direccionará su implementación desde la nube y su aplicación en dispositivos móviles. Así mismo se busca despertar en el estudiante el amor por la programación, el pensamiento computacional y a su vez las redes informáticas.

Palabras Clave: Realidad Aumentada, Redes Locales, Android, Java

Abstract

The way people relate to one another has changed over time; Today we have tools that we did not have before; which has made it easier to interact with people who are in other parts of the planet. Now, to be able to access these tools, it is necessary to have a network that provides access to the internet; This in turn results in the need to have experts who carry out the design and structuring of said network and thus achieve the objectives initially set. For this reason, this project seeks to facilitate the process of designing and structuring a local network, and not only this, but it will offer the user the security of designing this network correctly, making the most of the resources available in the area you want to implement the network. In addition, the present study aims to make use of augmented reality and high-level programming languages such as Java and implementation of Opencv libraries for the creation of an interactive application for the design and structuring of LAN networks. During development your implementation will be routed from the cloud and your application on mobile devices. It also seeks to awaken in the student the love for programming, computational thinking and in turn computer networks.

Keywords: Augmented Reality, Local Networks, Android, Java

Tabla De Contenido

Introducción	9
Definición Del Problema.....	10
Descripción Del Problema.....	10
Formulación Del Problema.....	11
Justificación	12
Objetivos	14
Objetivo General.....	14
Objetivos Específicos	14
Marco Conceptual Y Teórico.....	15
Realidad Aumentada	15
Redes LAN	17
Programación De Dispositivos Móviles	19
Ejemplos.....	23
Análisis.....	27
Análisis Inicial	27
Requerimientos De La Aplicación.....	27
Requerimientos Del Sistema.....	29
Diseño	30
Diseño Inicial De Los Modelos 3d	30
Codificación	34
Codificación Prototipo.....	34
Pruebas.....	55
Prueba De Capturar Coordenadas.....	56
Prueba De Reconocimiento De Planos.....	56
Prueba De Modelos 3d	57
Prueba Vista De Configuraciones.....	58
Prueba Final	60
Lineamientos De Calidad.....	60
Metodología	63
Etapa 1	64
Etapa 2	64
Etapa 3	64
Etapa 4	64
Recursos	65
Resultados.....	66
Conclusiones	67

Referencias.....68

Lista de Figuras

Figura 1.	30
<i>Vista Frontal de los modelos</i>	30
Figura 2.	31
<i>Vista lateral 1 de los modelos</i>	31
Figura 3.	31
<i>Vista trasera de los modelos</i>	31
Figura 4.	31
<i>Vista lateral 2 de los modelos</i>	31
Figura 5.	32
<i>Vista modelo de realidad aumentada en Cámara computador</i>	32
Figura 6.	32
<i>Vista Frontal de los modelos final</i>	32
Figura 7.	32
<i>Vista lateral 1 de los modelos final</i>	32
Figura 8.	32
<i>Vista trasera de los modelos final</i>	33
Figura 9.	33
<i>Vista lateral 2 de los modelos final</i>	33
Figura 10.	34
<i>Calculo del area de la sala</i>	34
Figura 11.	35
<i>Vista de las opciones del botón Capturar</i>	35
Figura 12.	49
<i>Vista programación grafica del panel</i>	49
Figura 13.	54
<i>Vista programación grafica del botón menú</i>	54
Figura 14.	54
<i>Vista programación grafica del botón ocultar</i>	54
Figura 15.	55
<i>Vista programación grafica del botón configuración del rack</i>	55
Figura 16.	56
<i>Evidencia captura de coordenadas</i>	56
Figura 17.	57
<i>Evidencia reconocimiento de plano</i>	57
Figura 18.	58
<i>Evidencia montaje de modelos 3d Router</i>	58
Figura 19.	58
<i>Evidencia de modelos 3d laterales</i>	58
Figura 20.	58
<i>Evidencia de sala modelos 3d</i>	58
Figura 21.	59
<i>Evidencia configuración del router</i>	59
Figura 22.	59
<i>Evidencia botón de ayuda</i>	59
Figura 21.	59
<i>Evidencia configuración del PC</i>	60

Lista De Tablas

Tabla 1.	61
<i>Lineamientos de calidad</i>	61
Tabla 2.	65
<i>Recursos</i>	65
Tabla 3.	66
<i>Resultados</i>	66

Introducción

Si hace 100 años se hubiese hablado de integrar un mundo virtual en la realidad, probablemente se pensaría que era algo ilógico, un simple sueño; sin embargo, hoy podemos dar cuenta de que esto es viable y está mucho más cerca de lo que parece. Gracias a la realidad aumentada hoy es posible lo que parecía imposible. Pero ¿Qué es la realidad aumentada? Bueno la realidad aumentada “se encarga de estudiar las técnicas que permiten integrar en tiempo real contenido digital con el mundo real” (Gonzales, Vallejo, Albusac, & Castro). Estas técnicas permiten no sumergirse totalmente en un mundo virtual sino superponer información adicional al mundo real, de modo que se obtenga una experiencia completa de integración y se haga de la realidad algo innovador y diferente. Ahora bien, es interesante notar que la realidad aumentada enriquece la percepción del mundo potenciando los sentidos que normalmente se usan para entender la realidad física; como lo son el gusto, el olfato, la vista y el tacto. (Telefonica, 2011)

Estas nuevas técnicas pueden ser de gran utilidad en el campo de la enseñanza (especialmente en áreas teórico-prácticas) pues ofrecerán a los estudiantes un conocimiento contextualizado que les permitirá potenciar sus habilidades y así cuando se enfrenten al mundo laboral estén al corriente de cómo aplicar sus conocimientos. Es por esto por lo que, este trabajo mostrara como diseñar una aplicación móvil basada en realidad aumentada que mejore la enseñanza de redes LAN. Para ello se implementarán 4 fases: planeación, diseño, codificación y pruebas; buscando obtener un análisis de los requerimientos, diseñar una interfaz de usuario, la creación de un código fuente y la implementación de pruebas de funcionamiento. Por último, es digno de mención que se espera que esta aplicación pueda ser usada inicialmente por los estudiantes de telecomunicaciones y sistemas del Cead Vélez de la UNAD; luego se espera que sea implementada en los demás centros zonales con los que cuenta la UNAD.

Definición Del Problema

Descripción Del Problema

A medida que pasa el tiempo y surgen nuevas tecnologías, los pilares de la educación están siendo transformados buscando renovar las formas de adquirir conocimiento contextualizado; a través de nuevas oportunidades para aprender en contextos distintos brindando nuevas perspectivas, con objetos virtuales, que de otra forma serían muy complejos de lograr.

Entre las contribuciones de la realidad aumentada a la educación encontramos:

- Motivación: esta tecnología innovadora hará que los estudiantes se encuentren motivados para aprender.
- Trabajo colaborativo: este nuevo método de aprendizaje facilita el trabajo en equipo y representa una técnica muy oportuna para realizar actividades entre alumnos.
- Construcción del conocimiento por parte del alumno: esta modalidad permitirá al estudiante aprender investigando paso a paso, siendo participe del proceso.
- Más información: permite acceder a más información, adentrarse en conocimientos que con tan solo observar su entorno no sería posible obtenerlos.
- Mayor accesibilidad: ahora es posible que el alumno desde un smartphone o Tablet tenga acceso a la realidad aumentada fácilmente.
- Perfeccionamiento de habilidades tecnológicas: al emplear la realidad aumentada el estudiante manejará un tipo de tecnología que estará potenciando su

aprendizaje y aplicación del idioma tecnológico que disponiendo de recursos habituales no lograría.

Así pues, como se analizó anteriormente este método de enseñanza reporta diversidad de beneficios a quienes la implementan; sin embargo, en muy pocas instituciones se implementa esta modalidad de estudio, y como consecuencia los estudiantes no se sienten satisfechos con su aprendizaje. Este es el caso de la UNAD; dado que en la enseñanza de materias teórico-prácticas como telecomunicaciones, redes y demás, no hace uso de dicha tecnología los estudiantes se están perdiendo de los beneficios mencionados inicialmente y su conocimiento esta siendo netamente teórico. De ahí que, la necesidad de contar con una aplicación interactiva que contextualice a estudiantes de la UNAD, en el aprendizaje, diseño y funcionamiento de redes LAN de una forma más atractiva y practica; ha repercutido en la iniciativa de desarrollar un software que satisfaga estas necesidades. Como resultado los estudiantes contarán con experiencias de educación innovadoras, no tradicionales que habiliten más y mejores oportunidades de aprender; y no solo esto, sino que potencien en ellos habilidades importantes para su desempeño laboral.

Formulación Del Problema

¿Cómo diseñar una aplicación móvil interactiva que mejore la forma de enseñanza de redes LAN en la UNAD a partir de la realidad aumentada?

Justificación

La expansión y el auge del internet ha innovado el modo en que las personas se comunican, se relacionan, trabajan, etc., repercutiendo en que el internet sea una herramienta indispensable para trabajar, entretenerse, comunicarse o estudiar. Ahora bien, para poder disfrutar de esta herramienta es necesario contar con una red LAN que permita acceder a la nube; sin embargo, la estructuración y el diseño de dicha red no se puede tomar a la ligera, debe ser cuidadosamente planeada. Por esto, este proyecto cobra relevancia pues permite al usuario sobre un espacio definido ver el diseño y la estructura de una red LAN perfectamente planificada y funcional; y no solo esto, sino que le proporcionara la codificación específica de cada aparato en dicha red.

Además, es importante resaltar que en ocasiones quienes estructuran una red LAN no hacen uso adecuado de los recursos con los que cuentan y como consecuencia se desperdician materiales y en ocasiones equipos. Es por esta razón que, se hace inevitable contar con un software que aparte de proporcionar la codificación de los equipos permita al usuario identificar cual seria la mejor manera de estructurar su red de modo que se optimicen los recursos disponibles y se aproveche el espacio con el que cuenta el usuario.

También el proyecto cobra notabilidad debido a que en la UNAD a pesar de que se trabajan simuladores de redes estos no satisfacen totalmente las necesidades de los estudiantes en cuanto a conocimiento innovador y aplicable. Como consecuencia los estudiantes, aunque están adquiriendo una preparación verídica de la materia este conocimiento no es del todo aplicable en

situaciones reales y eso provoca que se limite su aprendizaje y posterior práctica. Repercutiendo entonces en que se haga necesario contar con un simulador que ubique a los estudiantes en un entorno real que les permita afianzar sus conocimientos y cree en ellos la seguridad necesaria para usarlos en el ámbito laboral.

Por otra parte, mediante la implementación de metodologías ágiles en realidad aumentada se podrá adelantar una aplicación interactiva para la estructuración de redes LAN utilizando lenguajes de programación de alto nivel; lo que sentará las bases para publicar información sobre la programación en la realidad aumentada que cuenta con pocas publicaciones actualmente. Esto a su vez ayudara a los estudiantes de sistemas para que cuenten con información importante sobre programación que les pueda orientar en sus proyectos posteriores o les permita aumentar su conocimiento en el ámbito de la realidad aumentada y su implementación en una aplicación móvil interactiva.

Objetivos

Objetivo General

Desarrollar una aplicación Android interactiva basada en realidad aumentada usando el lenguaje de programación java para la enseñanza de redes LAN.

Objetivos Específicos

Realizar el análisis y la identificación de las necesidades, para la determinación de los requerimientos del aplicativo, usando los cuestionarios como técnica para obtener el documento de requisitos.

Diseñar los elementos de la aplicación móvil, para la enseñanza de redes LAN, haciendo uso del lenguaje de programación java para sistema operativo Android.

Evaluar la funcionalidad y usabilidad del aplicativo, para establecer un nivel aceptable de calidad, aplicando los lineamientos básicos de calidad para desarrolladores Android.

Marco Conceptual Y Teórico

Realidad Aumentada

La Realidad Aumentada es una tecnología en ascenso que los maestros deben dominar y tener en cuenta aprovechando las oportunidades que ofrece para la educación. Su empleo asiste un rol dinámico del alumnado, consintiendo situar en su entorno modelos 3D y simulaciones virtuales para que interactúen, vean, exploren y analicen. Se trata de una tecnología que concede aprender de un método más práctico a nuestros alumnos. (Blázquez A, 2017)

Un sistema de Realidad Aumentada cumple los parámetros descritos a continuación:

- Armoniza mundo virtual y real. El sistema integra información sintetizada a las imágenes captadas del mundo real.
- Interactivo en tiempo real. Así, los efectos exclusivos de películas que añaden intachablemente imágenes 3D fotorrealistas con imagen real no son computadas de manera interactiva y no se considera Realidad Aumentada
- Alineación 3D. La información del mundo virtual debe ser tridimensional y debe estar exactamente organizada con la imagen del mundo real. De esta forma, las aplicaciones que sobreponen capas gráficas 2D sobre la imagen del mundo real no son consideradas de Realidad Aumentada.

En la realidad aumentada los elementos que pueden intervenir son:

- Dispositivo con cámara: Wearable con cámara (relojes, gafas, etc.), Ordenador portátil con webcam, PC con webcam, Tablet, y Smartphone,.

- Un software comisionado para realizar las transiciones precisas con el fin de proporcionar la información adicional.
- Un activador de la información o disparador, llamado “trigger”. Este disparador puede ser una Imagen, un Entorno material (espacio urbano, paisaje y medio observado), un Marcador, un Objeto, o un Código QR.

Herramientas para la Realidad Aumentada

- ARToolKit

Es una librería para la fundación de softwares de Realidad Aumentada que hace uso de algoritmos de enfoque computacional para solventar el inconveniente del tracking. Las bibliotecas para tracking de video de ARToolKit emplean variados modelos físicos para estimar la orientación y posición real de la cámara en tiempo real; lo que proporciona el despliegue de un extenso paquete de aplicaciones de Realidad Aumentada.

- NyARToolkit

Es una librería visual de Realidad Aumentada establecida en ARToolKit; ejecuta sobre variadas plataformas virtuales: Action script 3, Java, C#, flash, Silverlight, Processing y Android.

- SLARToolkit

SLARToolkit es una librería de Realidad Aumentada utilizada para el celular de la empresa windows Silverlight. Su finalidad es verificar que las apk de Realidad Aumentada en tiempo real se caractericen por ser sencillas y rápidas de utilizar; se basa en NyARToolkit y ARToolkit. SLARToolkit aplica un piloto de licencia dual y alcanza a ser destinado para softwares de código cerrado o abierto limitado por restricciones.

- osgART

Es una biblioteca que simplifica el desarrollo de apk de Realidad Aumentada, pues combina la librería de búsqueda ARToolKit con Open SceneGraph; la librería prescribe 3 labores primordiales: el patrón (marcador), el registro fotométrico (oclusión, sombra) y la integración de alto nivel de la entrada de video (objeto de vídeo).

- AndAR (Android ARToolkit)

AndAR es una librería apoyada en Java que accede a usar Realidad Aumentada para dispositivos Android. Android captura y muestra el video; por su parte el algoritmo del ARToolkit ejecuta el reconocimiento, análisis geométricos y rastreo del patrón.

- FLARToolkit

Es una aplicación de Realidad Aumentada que se apoya en la biblioteca ARToolkit, admite el lenguaje AS3, se recomienda específicamente para utilizar realidad aumentada en la web.

- FLARManager

Esta librería facilita el desarrollo de softwares de Realidad Aumentada para Flash. Es semejante con variadas bibliotecas (FLARToolkit) y los marcos de rastreo 3D (Papervision3D) que suministra un método más vigoroso fundado en sucesos para la gestión de marcadores. Admite la gestión y detección de variados modelos y distintos marcadores de un patrón específico.

Redes LAN

Una red LAN se define como una red de área local que posibilita la comunicación entre los dispositivos que se encuentren dentro de dicha red. Ahora bien, para la interconexión de dichos dispositivos es necesario definir el direccionamiento IP que permita enviar y recibir información entre los dispositivos. Esta configuración ip requiere al menos de dirección ip la

cual puede ser versión 4 o versión 6, máscara de subred, Dirección IP de un router por defecto y Dirección IP de un servidor DNS. (Boronat, s. & Montagud, c. 2013)

Este tipo de red enlaza dispositivos y trasfiere datos por medio de cables basados en el estándar Ethernet. Es excelente para la administración de magnas porciones de datos a velocidades altas. Se determina por poseer dos componentes esenciales: switches y routers.

- Los switches admiten que las maquinas que se encuentran dentro de la red se notifiquen entre sí, absorbiendo conjuntos de datos y guiándolos al destinatario correcto. Los switches favorecen economizar capital e acrecientan la productividad pues hacen viable compartir la información y los recursos con un mínimo costo.
- Los enrutadores vinculan variedad de redes entre sí o con Internet. Una vez que obtiene los datos el enrutador los examina y los remite por la ruta más adecuada. Además, permiten proteger los datos de los ataques de seguridad y solventar qué dispositivos de cómputo poseen prioridades con respecto a otros.

Características de una red

- Escalabilidad: Se refiere a la facultad de ajustarse y reaccionar ágilmente al incremento de los usuarios y de las cargas de labor, asegurando la inversión realizada en la red y estableciendo la continuidad de la operación.
- Seguridad: hace referencia a la facultad de una red de poseer prácticas y políticas necesarias para advertir e inspeccionar el acceso no concedido, así como la utilización ilegal, de los datos de la empresa y sus recursos.

- **Automatización:** Se refiere a la facultad de configurar, aprovisionar, gestionar y probar los dispositivos de manera automática. Lo que a su vez incrementa la eficacia, evita faltas humanas y mitiga los gastos operativos.
- **Inteligencia:** se refiere a la capacidad de obtener información relevante de las aplicaciones, los dispositivos y los usuarios; de esta manera será más eficiente la operación de la red y se facilitará la toma de decisiones para el negocio.

Programación De Dispositivos Móviles

Un dispositivo móvil se conoce como aquel aparato que puede trasladarse de un lugar a otro sin perder su configuración y funcionalidad; además, están proyectados para brindar acompañamiento a sus beneficiarios en actividades de rutina. A la hora de implementar la programación en un dispositivo móvil se deben tener en cuenta ciertos factores que son importantes analizar a diferencia de la programación en computadores. Por ejemplo, pueden variar los lenguajes de programación, el aforo de la interfaz de usuario y capacidad de almacenamiento. Por otro lado, es importante notar que las aplicaciones que se realicen deben cumplir con los estándares de seguridad que garantizaran la confidencialidad, la disponibilidad y la integridad de los datos del usuario. (Carriles, O.)

Cabe resaltar que para realizar la programación de cualquier aplicación JAVA es necesario contar con un entorno de desarrollo; para este fin existen varios entornos con características diferentes y es a criterio del programador que se elegirá el más adecuado. Eclipse es uno de estos entornos, se caracteriza por su entorno de software multilenguaje de programación que envuelve un ambiente de desarrollo integrado IDE, por permitir instalar materiales de desarrollo para cualquier lenguaje, por permitir cambiar, mejorar o agregar

funcionalidades y permite integrar diversos lenguajes sobre un mismo IDE. De este modo, es posible ejecutar el entorno de desarrollo de eclipse y las librerías de Android instalando en el ordenador las librerías de desarrollo de java necesarias.

Android

¿Qué es Android? Es un sistema operativo basado en Linux, contiene un foco de sistema operativo multiplataforma gratuito y libre; proporcionando todas las interfaces imprescindibles para perfeccionar sencillamente aplicaciones que puedan acceder a distintitas funciones del celular como las llamadas, la geolocalización, la agenda, la cámara, etc., manejando el lenguaje de programación java. Ahora bien, al usar Android las aplicaciones que son programadas en java corren sobre una máquina virtual llamada Dalvik y su seguridad depende en gran manera del programador; sin embargo, con el enfoque de programación segura es posible encontrar apoyo para implementar programas libres de vulnerabilidades. Este enfoque de programación segura engloba las siguientes técnicas: validación de entrada (validación de caracteres inválidos, corrección de cadenas invalidas), hardcodeo de credenciales, autenticación, autorización y asignación de permisos.

Realidad aumentada en Android

- NyARToolki (Serrano, 2010)

Fue diseñada por usuarios japoneses de Android de ahí que la documentación existente este la mayoría en este idioma. Además, si desea usar esta librería es necesario en el entorno de trabajo editar las preferencias de configuración de todos los archivos a UTF-8 debido a que la documentación del código esta creada en japonés. Cuenta con un exportador para Blender a formato MQO, también permite incluir formatos OBJ y MD2 compatibles con Blender. La distribución de un software básico es equivalente al de AndAR, este SDK emplea un motor de

renderizado externo [MIN3D] en vez de cargar los objetos directamente. Min3D es una biblioteca ligera 3D para Android que utiliza Java con OpenGL ES; esta librería brinda afinidad con la versión 1.5 de Android, OpenGL ES 1.0 y posteriores. Su documentación está compuesta en bloques adentro del mismo código.

- Vuforia (Serrano, 2010)

Vuforia es un SDK desarrollado por Qualcomm, en 2010 puso a disposición del público sus frameworks de desarrollo y los llamaron Vuforia. Este SDK se encuentra aprovechable para Android e iOS y se fundamenta en la identificación de imágenes fundadas en rasgos especiales, de modo que admite RA sin marcadores o marcadores naturales (targets). También, tiene instalado un plugin que le permite interactuar con Unity3D y brinda la posibilidad de producir botones virtuales que amplíen las rutas de interacción con el usuario. Para que una aplicación basada en RA funcione en Vuforia debe tener los siguientes componentes fundamentales:

Cámara: Este módulo se cerciora de que cada frame capturado supere al tracker. Aquí es necesario anunciar cuándo la aplicación comienza la captura y cuando finaliza. Además, es importante tener cuenta que tanto el formato como el tamaño de cada frame penderá del aparato móvil en uso.

Convertidor de imagen: En este espacio se transforma el formato de la cámara a un formato interoperable con OpenGL y para el tracking de los marcadores. Esta conversión comprende la disminución de la tasa de muestreo y así disponer de la imagen de la cámara en variedad de resoluciones.

Activador: Esta unidad envuelve los códigos de visión artificial que se facultan en la localización y búsqueda de los objetos de cada frame. Variedad de algoritmos se facultan del descubrimiento de nuevos “targets” o “markers” y de la evaluación de los botones virtuales. Los alcances se almacenan en un objeto de estado. Este módulo consigue llevar

variados conjuntos de objetos, pero no permite que haya activo más de un módulo simultáneamente. Video Background Renderer: Esta unidad procesa la fotografía contenida en la entidad de estado. La productividad de la representación de vídeo de base está mejorada para instrumentos determinados. Cada uno de estos elementos deben ser inicializados en nuestro software. En cada frame se renueva el objeto de estado y se llama a los servicios de renderizado. Por otra parte, nuestro software debe realizar los siguientes 3 pasos:

1. Examinar el objeto de estado para probar nuevos targets o markers detectados.
2. Reemplazar la lógica de la apk con los recientes datos de entrada.
3. Renderizar los elementos virtuales: instaurando los marcadores por medio de un sistema online. Una vez organizado el retrato que valdrá como target o marcador, se accede a este sistema. Se produce una nueva propuesta y se sube la imagen. El sistema considera la imagen y le determina una apreciación que muestra la eficacia del marcador en función de la cifra de peculiaridades individuales expuestas por el sistema. Luego se convierte la imagen a formatos distinguidos por la librería. El sistema retorna dos registros: un .xml con la configuración del target y un archivo binario que entraña los datos rastreables.

- AndAR(Serrano, 2010)

AndAR es una biblioteca fundamentalmente para equipos Android que funciona por medio de una API escrita en Java, fue creada en 2010. Para utilizarla es necesario adquirir la licencia dual de ARToolKit y se encuentra licenciada bajo GNU GPL v3; no obstante, se halla la probabilidad de disponer de ella en apk de código sellado realizando el pago de licencia a ARToolworks. La biblioteca trabaja con marcadores básicos y manipula un patrón de texto que gestiona los marcadores. Estas plantillas pueden formarse por medio del software `mk_patt` de ARToolKit, que nos ayuda a transformar una figura patrón en una plantilla que pueda leída por la

biblioteca. En la documentación de ARToolKit se puede hallar la plantilla para realizar nuestro propio target, el cual tiene unas dimensiones exactas para que llegue a ser reconocido. El proceso consiste básicamente en: primero, tenemos la actividad primordial AndARActivity; luego se crea como mínimo una acción extra CustomActivity, que hereda de esta clase abstracta. Esta clase efectúa las actividades concernientes al mando de la cámara, la localización de marcadores y la visualización de vídeo. También permite administrar las labores relativas a inserción de objetos, desde onCreate. Luego, se encuentra ARRenderer que es la clase encargada de lo concerniente con OpenGL. Esta clase será usada para realizar el renderizado, y nos permitirá combinar la realidad con los objetos aumentados. También en esta clase será posible tramitar los parámetros de posición, iluminación y cualquier trabajo que tenga que ver con la inicialización de OpenGL. Una extensión de esta clase permite crear la clase ARObjects que pueda enseñar modelos en formato .obj. de este modo, ARObject será la clase encargada de proveer asiduamente la información sobre el target.

Ejemplos

- Aplicación Móvil de Realidad Aumentada para la Enseñanza de la Clasificación de los Seres Vivos a Niños de Tercer Grado. (Solano Villanueva, Casas Díaz, & Guevara Bolaños, 2015):

Esta aplicación de uso estudiantil consiste en dos módulos; el de autenticación y el registro de usuarios, el segundo el de realidad aumentada, cuyo objetivo es proyectar los modelos en 3D y demás contenidos multimedia en la pantalla del dispositivo. Esta aplicación se realizó con el SDK (Kit de Desarrollo de Software) Vuforia. La aplicación funciona de la siguiente manera: si el contenido es un modelo en 3D, se visualiza inmediatamente, puesto que ya se

encuentra almacenado en el dispositivo para dar un mejor rendimiento. En caso de que sea cualquier otro contenido multimedia, se establece en una comunicación hacia los servicios web por medio de JavaScript con el fin de recuperar del servidor dichos recursos y presentarlos al usuario, en este caso el estudiante. Dichos servicios web están creados bajo la arquitectura Representational State Transfer (REST) para dar una mayor flexibilidad y escalabilidad.

Los lenguajes de programación que se usaron en este proyecto fueron los siguientes: Java EE 6 para la aplicación web usando tecnologías como JAX-RS 1.1, JavaServer Faces (JSF) 2.0, Enterprise JavaBeans (EJB) 3.1, Java Persistence (App) 2.0 y Context and Dependency Injection (CDI) 1.0. El segundo es Android para realizar la aplicación móvil. Y, por último, JavaScript, con el cual se realizaron las mejoras a la interfaz. Se utilizó el servidor de aplicaciones GlassFish, cuyo código es compatible con Java EE y que proporciona todas las funciones para la implementación de Java EE 6. El motor de RA fue Vuforia así como Unity 5 una plataforma para crear experiencias interactivas en 2D y 3D. La base de datos que se utilizó fue PostgreSQL que es un sistema de gestión de bases de datos objeto-relacional. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema.

- Programación de una aplicación Android para la visualización de modelos 3D en realidad aumentada. (Arias, 2020)

En este documento se puede encontrar básicamente los fundamentos y diferentes conceptos de la programación en realidad aumentada; además brindan tutoriales que permiten al usuario aprender a usar herramientas como entornos de desarrollo integrado que se utilizarán en la creación de las aplicaciones. Luego también mediante tutoriales se brindan al lector las herramientas para que consiga programar distintas apk que permitan introducir modelos propios

y animaciones 3D, además de visualizarlos en su dispositivo usando lo aprendido de la realidad aumentada. Los temas de los tutoriales que se trabajan allí son los siguientes: Explicación del funcionamiento y conceptos de ARCore, Descargar ARCore, android Studio, preparación del Hardware (preparación de dispositivo, comprobación de requisitos, opciones de desarrollador , depuración usb), aplicación base (Descarga de la aplicación soporte, Conceptos de programación, Explicación del código base), Inclusión de modelos 3D propios (Importar y modificar archivos 3D en una aplicación, Aplicación con varios modelos 3D, Aplicación de un modelo 3D con animaciones), Instalación en dispositivo (Cambiar ID de la aplicación, Cambiar Nombre de la aplicación, Cambiar Icono de la aplicación).

- Realidad aumentada aplicada a objetos de aprendizaje para asignaturas de ingeniería informática. (Mateus, 2013)

En este escrito se describe y ejemplifica la aplicación de la Realidad Aumentada en Objetos de Aprendizaje, especialmente en asignaturas de Ingeniería Informática. Para ello se describen los métodos para la creación de los objetos de aprendizaje, en especial la metodología SAM que es la elegida a implementar en este proyecto. Luego se procede a desarrollar los objetos de aprendizaje por medio de 3 iteraciones a la aplicación donde en cada una se hace una evaluación del problema, se realiza un diseño y se proceden a explicar los tópicos de programación utilizados. Una vez realizada esta caracterización se procede a explicar la implementación de los patrones de realidad aumentada para la producción del prototipo computacional. Allí se explica que La herramienta que se utilizó tanto para realizar las animaciones y apk de realidad fue Unity 3D; además se implementó el SDK de Vuforia para la aplicación de Realidad Aumentada y Blender para las animaciones. Por último, el documento

procede a mencionar como funciona el prototipo funcional; explicando cómo se ejecuta en la aplicación el área de Algoritmos y Programación con los temas de listas simplemente ligadas, matrices, y apuntadores; en el área de Matemáticas Aplicadas para implementar la enseñanza de cada uno de las areas. Concluyendo con una validación de los patrones de realidad aumentada que se aplicaron en los objetos de aprendizaje, haciendo uso de un grupo piloto y un grupo control.

Análisis

Análisis Inicial

El primer paso que se siguió para la obtención de los requerimientos de la aplicación fue realizar una encuesta a los estudiantes de telecomunicaciones de la UNAD con el fin de identificar las características mas adecuadas para un software como el que se desea realizar. De acuerdo con las respuestas se identificó que las siguientes son características esenciales del software:

- Practico, sencillo, intuitivo y gráfico.
- Uso de dispositivos actuales.
- Implementación de nuevas tecnologías como IPV6.
- Asignación de host y direccionamiento IP manual y automática.

Requerimientos De La Aplicación

Una vez identificados los aspectos que se consideran esenciales para la realización de un software que satisfaga las necesidades, se procede a plantear la idea que solucione dicha problemática.

- El sistema en el módulo principal deberá capturar 3 puntos de geolocalización que permitan medir el espacio de la habitación y así montar el modelo basándose en estas coordenadas y medidas.

- El sistema en el módulo principal deberá hacer uso de la realidad aumentada para mostrar al usuario como se organizará una red LAN en el espacio que capturó con su cámara.
- El sistema en el módulo principal deberá permitir al usuario seleccionar un dispositivo de la red y ver la configuración básica del mismo.
- El sistema deberá estar configurado en el lenguaje de programación JAVA e implementarse para dispositivos con sistema operativo android.
- El sistema deberá contar con una interfaz gráfica sencilla e intuitiva que facilite la navegación por la aplicación.
- El sistema en la opción de configuración básica deberá mostrar al usuario la configuración de básica de cada dispositivo de red y el código correspondiente para ejecutarla.
- El sistema en el módulo principal deberá organizar los siguientes dispositivos en el espacio real proporcionado: switches, routers, computadores, rack, y (de ser necesario) access point.
- El sistema deberá hacer uso del protocolo IPV4 para cada uno de los dispositivos presentes en la red LAN.
- El sistema deberá mostrar como configuración básica en un computador los siguientes aspectos: dirección IPV4, máscara de subred, Dirección IP de un router por defecto.
- El sistema deberá mostrar como configuración básica en un router los siguientes aspectos: nombre del router, contraseña, configuración de interfaces y de línea.

- El sistema deberá realizar las siguientes funciones: tomar captura del entorno donde se desea implementar la red, dibujar un modelo 3d en este espacio de como quedaría la red ya implementada, permitir que el usuario elija un dispositivo y mostrarle la configuración básica del mismo.
- El sistema realizara el cálculo de las direcciones Ip, las máscaras de subred, los hosts permitidos en el área especificada; así como el número de routers y switchs necesarios.
- El sistema deberá implementar como máximo 30 hosts en el espacio asignado.

Requerimientos Del Sistema

Para que sea posible utilizar la realidad aumentada en el dispositivo móvil será necesario comprobar los siguientes requisitos mínimos de hardware:

- Versión Android superior a 8.0.
- Dispositivo compatible con Vuforia Ground Plane.
- Acceso a cámara del móvil.
- Acceso a la geolocalización del móvil.

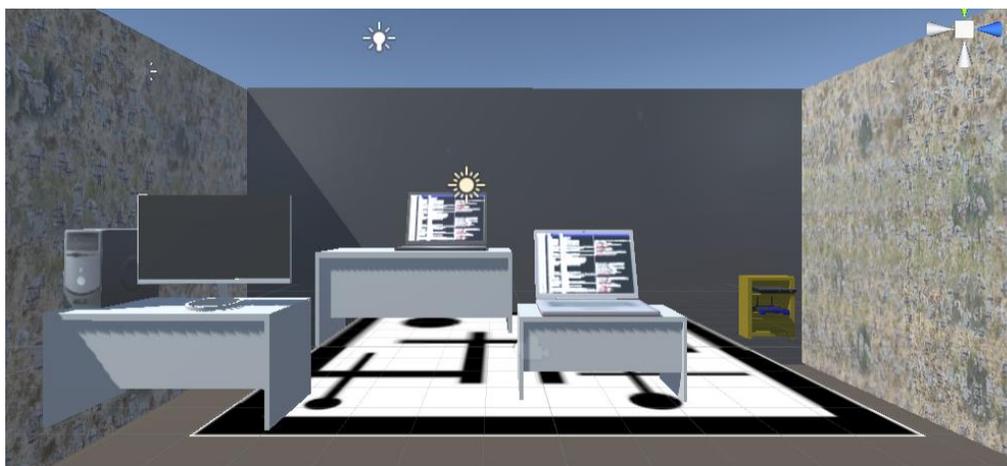
Diseño

Diseño Inicial De Los Modelos 3d

Para el diseño inicial de la red se seleccionaron modelos .obj ya elaborados anteriormente que se encontraron gratuitamente en internet. Estos modelos en formato .obj permitieron realizar un diseño inicial de la red que contara con portátiles, computadores de mesa, y un rack con switch y routers. Ahora bien, en el caso de las paredes y el techo de la sala se diseñaron completamente utilizando las herramientas de modelado que ofrece Unity. También es importante recalcar que se realizó un segundo diseño de la red en el cual los modelos se plasmaron no ajustándose a la visibilidad de un target sino distribuyéndose a lo largo de un plano que es reconocido utilizando herramientas de realidad aumentada. A continuación, se muestran los diseños iniciales y finales.

Figura 1.

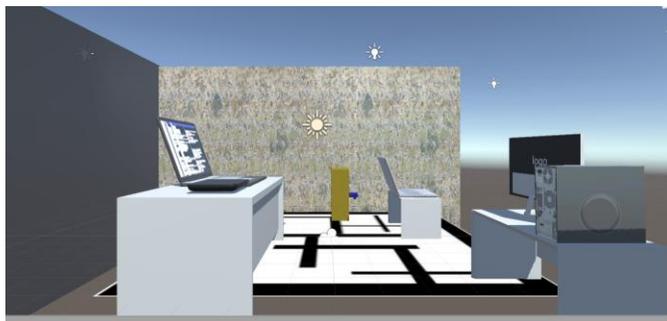
Vista Frontal de los modelos



Fuente. El autor

Figura 2.

Vista lateral 1 de los modelos



Fuente. El autor

Figura 3.

Vista trasera de los modelos



Fuente. El autor

Figura 4.

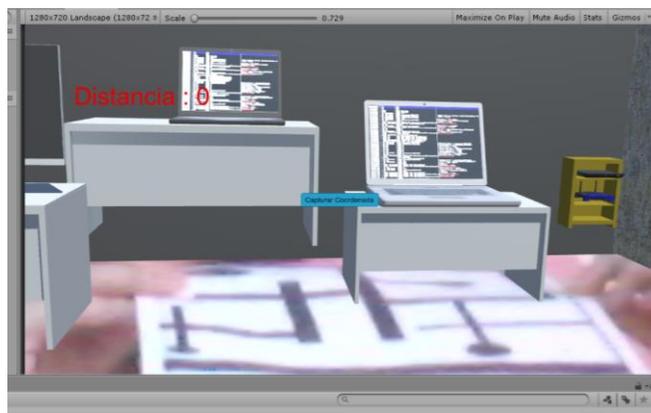
Vista lateral 2 de los modelos



Fuente. El autor

Figura 5.

Vista modelo de realidad aumentada en Cámara computador



Fuente. El autor

Figura 6.

Vista Frontal de los modelos final



Fuente. El autor

Figura 7.

Vista lateral 1 de los modelos final



Fuente. El autor

Figura 8.

Vista trasera de los modelos final



Fuente. El autor

Figura 9.

Vista lateral 2 de los modelos final



Fuente. El autor

Codificación

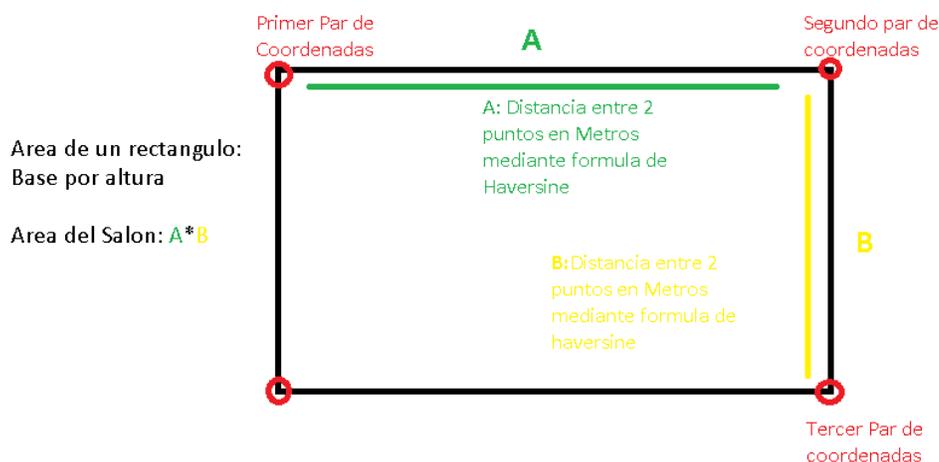
Codificación Prototipo

Codificación captura de coordenadas

El primer bloque de codificación de este trabajo se centró en la codificación del algoritmo para capturar las coordenadas del GPS del dispositivo móvil y mediante ellas usando la fórmula de haversine es posible obtener el tamaño del área de la sala en la que se quiere montar la red. La fórmula haversine determina la distancia del círculo máximo entre dos puntos en una esfera dadas sus longitudes y latitudes; de esta forma, teniendo las coordenadas de longitud y latitud es posible determinar la distancia entre dos puntos y en consecuencia determinar el área total del salón como se muestra en la figura 10.

Figura 10.

Calculo del area de la sala



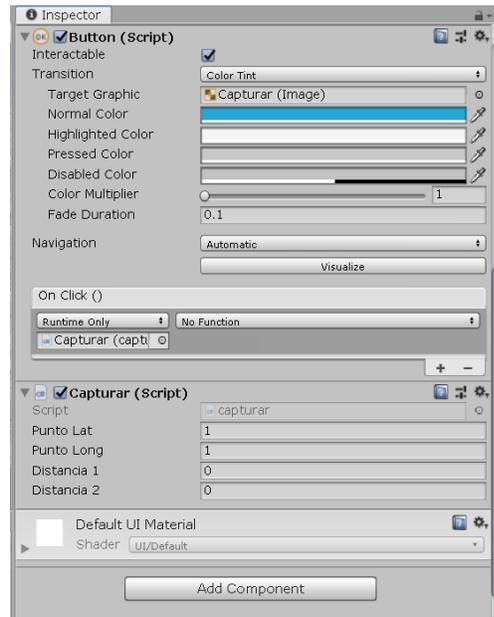
Fuente. El autor

Para poder capturar las coordenadas en el momento específico se agregó un botón que permita tomar estas coordenadas cuando el usuario así lo desee. Luego con el código que se

muestra a continuación se realizó la configuración y programación del botón para capturar y guardar cada par de coordenadas.

Figura 11.

Vista de las opciones del botón Capturar



Fuente. El autor

Código captura de coordenadas

```
public class capturar : MonoBehaviour
{
    // Inicializamos todas las variables a usar en la clase
    //Arreglos para guardar los datos de cada coordenada
    private string myFloatKey = "dato1";
    private string myFloatKey2 = "dato2";
    private string myFloatKey3 = "dato3";
```

```
private string myFloatKey4 = "dato4";
private string myFloatKey5 = "dato5";
private string myFloatKey6 = "dato6";

// Variables que toman el valor actual de cada coordenada
float datola1;
float datolo1;
float datola2;
float datolo2;
float datola3;
float datolo3;

public static float actualLat;
public static float actualLong;

//Variables para guardar la distancia entre cada par de coordenadas
public static float distancia1;
public static float distancia2;
public static float distanciaf;

//Variables para llevar la cuenta de cada click al boton
int buttonPressedCounter = 0;

float FormulaHaversine(float lat1, float long1, float lat2, float long2)
{
    //Formula para calcular la distancia en metros entre dos puntos de coordenadas
    float earthRad = 6371;
    float lRad1 = lat1 * Mathf.Deg2Rad;
```

```

float lRad2 = lat2 * Mathf.Deg2Rad;

float dLat = (lat2 - lat1) * Mathf.Deg2Rad;

float dLong = (long2 - long1) * Mathf.Deg2Rad;

float a = Mathf.Sin(dLat / 2) * Mathf.Sin(dLat / 2) +
          Mathf.Cos(lRad1) * Mathf.Cos(lRad2) *
          Mathf.Sin(dLong / 2) * Mathf.Sin(dLong / 2);

float c = 2 * Mathf.Atan2(Mathf.Sqrt(a), Mathf.Sqrt(1 - a));

return earthRad * c;
}

IEnumerator Start()
{
    //Llama el servicio de geolocalizacion tan pronto carga la escena
    // First, check if user has location service enabled
    if (!Input.location.isEnabledByUser)
        yield break;

    // Start service before querying location
    Input.location.Start();

    // Wait until service initializes
    int maxWait = 20;
    while (Input.location.status == LocationServiceStatus.Initializing && maxWait > 0)
    {

```

```
        yield return new WaitForSeconds(1);
        maxWait--;
    }

    // Service didn't initialize in 20 seconds
    if (maxWait < 1)
    {
        print("Timed out");
        yield break;
    }

    // Connection has failed
    if (Input.location.status == LocationServiceStatus.Failed)
    {
        print("Unable to determine device location");
        yield break;
    }
    else
    {
        // Access granted and location value could be retrieved
        print("Location: " + Input.location.lastData.latitude + " " +
Input.location.lastData.longitude + " " + Input.location.lastData.altitude + " " +
Input.location.lastData.horizontalAccuracy + " " + Input.location.lastData.timestamp);
    }
}
```

```
}  
  
// Update is called once per frame  
  
public void Update()  
{  
  
    StartCoroutine(Start());  
  
    actualLat = Input.location.lastData.latitude;  
  
    actualLong = Input.location.lastData.longitude;  
  
    // Establece estas variables como la longitud y latitud respectivas del dispositivo  
    movil segun el GPS  
  
}  
  
public void Save()  
{  
  
    //Evalua cuantas veces se ha hecho click en el boton y de  
  
    //acuerdo a esto tma una decision  
  
    buttonPressedCounter++;  
  
    if (buttonPressedCounter == 1)  
    {  
  
        PlayerPrefs.SetFloat(myFloatKey, actualLat);  
  
        PlayerPrefs.SetFloat(myFloatKey2, actualLong);  
  
    }  
  
}
```

```
}
```

```
else
```

```
    if (buttonPressedCounter == 2)
    {
        PlayerPrefs.SetFloat(myFloatKey3, actualLat);
        PlayerPrefs.SetFloat(myFloatKey4, actualLong);
    }
```

```
else
```

```
    if (buttonPressedCounter == 3)
    {
        PlayerPrefs.SetFloat(myFloatKey5, actualLat);
        PlayerPrefs.SetFloat(myFloatKey6, actualLong);
        bool result = PlayerPrefs.HasKey(myFloatKey);
        if (result)
        {
            datola1 = PlayerPrefs.GetFloat(myFloatKey);
            datolo1 = PlayerPrefs.GetFloat(myFloatKey2);
            datola2 = PlayerPrefs.GetFloat(myFloatKey3);
            datolo2 = PlayerPrefs.GetFloat(myFloatKey4);
            datola3 = PlayerPrefs.GetFloat(myFloatKey5);
            datolo3 = PlayerPrefs.GetFloat(myFloatKey6);
            distancia1 = FormulaHaversine(datola1, datolo1, datola2, datolo2);
            distancia2 = FormulaHaversine(datola1, datolo1, datola3, datolo3);
        }
    }
```

```

        distanciaf = (distancia1 * distancia2) * 1000;
    }
}

private void OnGUI()
{
    string mensaje =
        "\nDistancia : " + distanciaf +
        "\na lat: " + actualLat +
        "\na lon: " + actualLong;

    GUI.contentColor = Color.red;
    GUI.skin.label.fontSize = 50;
    GUI.Label(new Rect(100, 60, 1000, 900), mensaje);
}
}

```

Codificación modelos

El segundo bloque de codificación de este trabajo se centró en la codificación del algoritmo para la identificación del número de computadoras a organizar dependiendo del tamaño de la sala. De esta forma se estableció que si el área de la sala es menor a 5 metros solo se mostrarán 7 ordenadores, si el área es mayor a 5 metros y menor a 10 se mostrarán 11

computadoras, si el área es mayor a 10 metros y menor a 15 se mostraran 14 computadoras y por último si el área es mayor a 15 metros se mostraran 17 computadoras.

```
public class Modelos : MonoBehaviour

{

    public GameObject modelo1;

    public GameObject modelo2;

    public GameObject modelo3;

    public GameObject modelo4;

    // Update is called once per frame

    void Update()

    {

        if (capturar.distanciaf < 0.05)

        {

            modelo1.SetActive(true);

            modelo2.SetActive(false);

            modelo3.SetActive(false);

            modelo4.SetActive(false);

        }

        else

        if (capturar.distanciaf < 1)

        {

            modelo1.SetActive(true);
```

```
        modelo2.SetActive(true);
        modelo3.SetActive(false);
        modelo4.SetActive(false);
    }
else
if (capturar.distanciaf < 1.5)
{
    modelo1.SetActive(true);
    modelo2.SetActive(true);
    modelo3.SetActive(true);
    modelo4.SetActive(false);
}
else
if (capturar.distanciaf > 1.5)
{
    modelo1.SetActive(true);
    modelo2.SetActive(true);
    modelo3.SetActive(true);
    modelo4.SetActive(true);
}
}
}
```

Codificación Escalar

El tercer bloque de codificación de este trabajo se centró en diseñar un algoritmo que permite ubicar el número de computadores designados de acuerdo con el área de la sala distribuidos equitativamente en este espacio. Esto incluye cambiarlos de posición y tamaño dependiendo de cuantos ordenadores es posible ubicar en la sala: este cálculo se realizó mediante el supuesto de que un ordenador con su mesa ocupa un tamaño de 1 metro cuadrado.

```
public class Scale : MonoBehaviour
{
    float x, y, z, p, p2, p3;
    Vector3 newSize;
    // Update is called once per frame
    void Update()
    {
        x = 15.37328f;
        y = 15.37328f;
        z = 15.37328f;
        p = -13.3f;
        p2 = -18.65587f;
        p3 = -4f;
        if (capturar.distanciaf < 0.05)
        {
            transform.localPosition = new Vector3(p, p2, p3);
        }
    }
}
```

```
        transform.localScale = new Vector3(x, y, z);
    }
else
if (capturar.distanciaf < 1)
{
    transform.localPosition = new Vector3(p, p2, p3);
    transform.localScale = new Vector3(x, y, z); } } }

public class Scale2 : MonoBehaviour
{
    float x, y, z, p, p2, p3;
    Vector3 newSize;
    void Update()
    {
        x = 15.37328f;
        y = 15.37328f;
        z = 15.37328f;
        p = -13.3f;
        p2 = -18.65587f;
        p3 = -32f;
        if (capturar.distanciaf < 0.05)
        {
        }
    }
else
```

```

if (capturar.distanciaf < 1)
{
    transform.localScale = new Vector3(x, y, z);
    transform.localPosition = new Vector3(p, p2, p3);
}
}
}

```

Codificación de panel

El cuarto bloque de codificación de este trabajo se centró en la codificación del algoritmo para configurar el panel que se insertó en la aplicación de modo que el menú solo sea visible cuando se solicite y se oculte cuando no sea necesario. Para esto se implementó una clase ocultar botón que esconde todos los botones del menú tan pronto se carga la escena, otra clase que oculta el panel cuando se presiona un botón y por último otra clase que muestra los elementos en el menú dependiendo del botón seleccionado; por ejemplo, si se selecciona el botón de configuración del pc1 será visible la configuración del pc, si se selecciona el botón de configuración del pc2 se mostrará la configuración del pc2 y así sucesivamente.

```

public class OcultarBoton : MonoBehaviour
{
    public Button Track;

    public Button TC1, TC2, TC3, TC4, TC5, TC6, TC7, TC8, TC9, TC10, TC11, TC12,
    TC13, TC14, TC15, TC16, TC17;

    public GameObject Pan1;

```

```
public Image MIIMAGEN, ip1, ip2, ip3, ip4, ip5, ip6, ip7, ip8, ip9, ip10, ip11, ip12,  
ip13, ip14, ip15, ip16, ip17;
```

```
void Start()
```

```
{ Track.gameObject.SetActive(false);
```

```
  TC1.gameObject.SetActive(false);
```

```
  TC2.gameObject.SetActive(false);
```

```
  TC3.gameObject.SetActive(false);
```

```
  TC4.gameObject.SetActive(false);
```

```
  TC5.gameObject.SetActive(false);
```

```
  TC6.gameObject.SetActive(false);
```

```
  TC7.gameObject.SetActive(false);
```

```
  TC8.gameObject.SetActive(false);
```

```
  TC9.gameObject.SetActive(false);
```

```
  TC10.gameObject.SetActive(false);
```

```
  TC11.gameObject.SetActive(false);
```

```
  TC12.gameObject.SetActive(false);
```

```
  TC13.gameObject.SetActive(false);
```

```
  TC14.gameObject.SetActive(false);
```

```
  TC15.gameObject.SetActive(false);
```

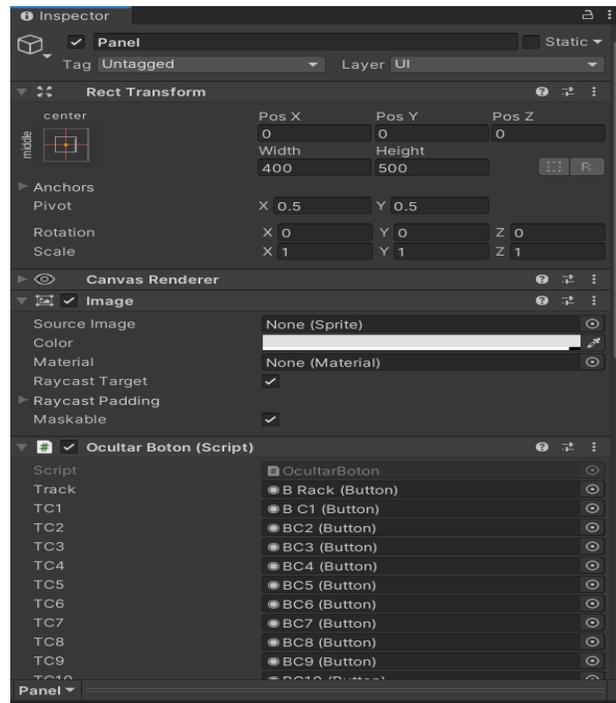
```
  TC16.gameObject.SetActive(false);
```

```
  TC17.gameObject.SetActive(false);
```

```
  ip1.gameObject.SetActive(false);
```

```
  ip2.gameObject.SetActive(false);
```

```
ip3.gameObject.SetActive(false);  
ip4.gameObject.SetActive(false);  
ip5.gameObject.SetActive(false);  
ip6.gameObject.SetActive(false);  
ip7.gameObject.SetActive(false);  
ip8.gameObject.SetActive(false);  
ip9.gameObject.SetActive(false);  
ip10.gameObject.SetActive(false);  
ip11.gameObject.SetActive(false);  
ip12.gameObject.SetActive(false);  
ip13.gameObject.SetActive(false);  
ip14.gameObject.SetActive(false);  
ip15.gameObject.SetActive(false);  
ip16.gameObject.SetActive(false);  
ip17.gameObject.SetActive(false);  
MIIMAGEN.gameObject.SetActive(false);  
}  
}
```

Figura 12.*Vista programación grafica del panel***Fuente.** El autor

```

public class OcultarBotones : MonoBehaviour

{

    public Button Track;

    public Button TC1, TC2, TC3, TC4, TC5, TC6, TC7, TC8, TC9, TC10, TC11, TC12,
TC13, TC14, TC15, TC16, TC17;

    public GameObject Pan1;

    public Image MIIMAGEN, ip1, ip2, ip3, ip4, ip5, ip6, ip7, ip8, ip9, ip10, ip11, ip12,
ip13, ip14, ip15, ip16, ip17;

    public void OcultarPanel()

    {

```

```
Pan1.gameObject.SetActive(false);

}

public void MostrarPanel()
{
    ip1.gameObject.SetActive(false);
    ip2.gameObject.SetActive(false);
    ip3.gameObject.SetActive(false);
    ip4.gameObject.SetActive(false);
    ip5.gameObject.SetActive(false);
    ip6.gameObject.SetActive(false);
    ip7.gameObject.SetActive(false);
    ip8.gameObject.SetActive(false);
    ip9.gameObject.SetActive(false);
    ip10.gameObject.SetActive(false);
    ip11.gameObject.SetActive(false);
    ip12.gameObject.SetActive(false);
    ip13.gameObject.SetActive(false);
    ip14.gameObject.SetActive(false);
    ip15.gameObject.SetActive(false);
    ip16.gameObject.SetActive(false);
    ip17.gameObject.SetActive(false);

    MIIMAGEN.gameObject.SetActive(false);

    Pan1.gameObject.SetActive(true);
}
```

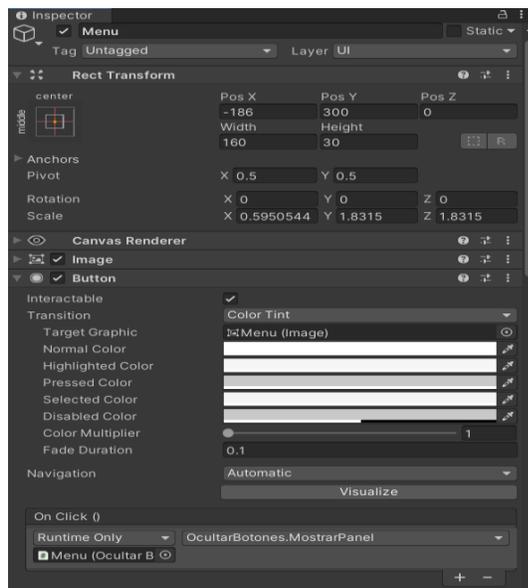
```
if (capturar.distanciaf < 0.05)
{
    Track.gameObject.SetActive(true);
    TC1.gameObject.SetActive(true);
    TC2.gameObject.SetActive(true);
    TC3.gameObject.SetActive(true);
    TC4.gameObject.SetActive(true);
    TC5.gameObject.SetActive(true);
    TC6.gameObject.SetActive(true);
    TC7.gameObject.SetActive(true);
    TC8.gameObject.SetActive(false);
    TC9.gameObject.SetActive(false);
    TC10.gameObject.SetActive(false);
    TC11.gameObject.SetActive(false);
    TC12.gameObject.SetActive(false);
    TC13.gameObject.SetActive(false);
    TC14.gameObject.SetActive(false);
    TC15.gameObject.SetActive(false);
    TC16.gameObject.SetActive(false);
    TC17.gameObject.SetActive(false);
}
else
```

```
if (capturar.distanciaf < 1)
{
    Track.gameObject.SetActive(true);
    TC1.gameObject.SetActive(true);
    TC2.gameObject.SetActive(true);
    TC3.gameObject.SetActive(true);
    TC4.gameObject.SetActive(true);
    TC5.gameObject.SetActive(true);
    TC6.gameObject.SetActive(true);
    TC7.gameObject.SetActive(true);
    TC8.gameObject.SetActive(true);
    TC9.gameObject.SetActive(true);
    TC10.gameObject.SetActive(true);
    TC11.gameObject.SetActive(true);
    TC12.gameObject.SetActive(false);
    TC13.gameObject.SetActive(false);
    TC14.gameObject.SetActive(false);
    TC15.gameObject.SetActive(false);
    TC16.gameObject.SetActive(false);
    TC17.gameObject.SetActive(false);
}
else
```

```
if (capturar.distanciaf < 1.5)
{
    Track.gameObject.SetActive(true);
    TC1.gameObject.SetActive(true);
    TC2.gameObject.SetActive(true);
    TC3.gameObject.SetActive(true);
    TC4.gameObject.SetActive(true);
    TC5.gameObject.SetActive(true);
    TC6.gameObject.SetActive(true);
    TC7.gameObject.SetActive(true);
    TC8.gameObject.SetActive(true);
    TC9.gameObject.SetActive(true);
    TC10.gameObject.SetActive(true);
    TC11.gameObject.SetActive(true);
    TC12.gameObject.SetActive(true);
    TC13.gameObject.SetActive(true);
    TC14.gameObject.SetActive(true);
    TC15.gameObject.SetActive(false);
    TC16.gameObject.SetActive(false);
    TC17.gameObject.SetActive(false);
}
}
}
```

Figura 13.

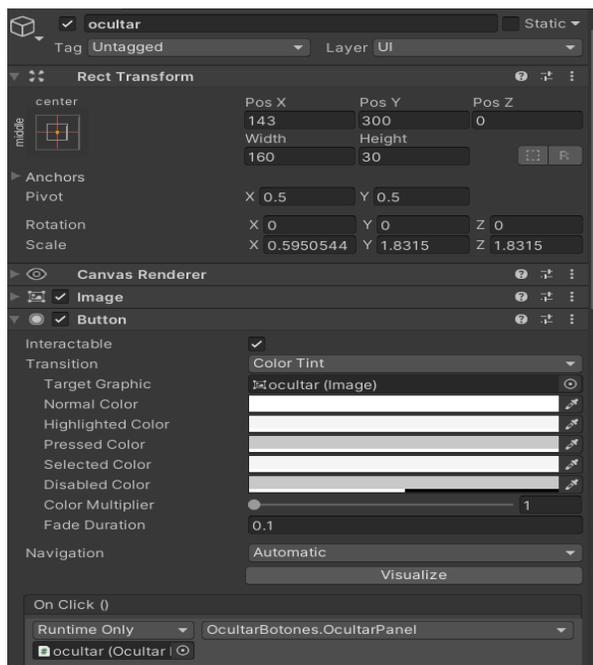
Vista programación grafica del botón menú



Fuente. El autor

Figura 14.

Vista programación grafica del botón ocultar



Fuente. El autor

```

public class TextoRack : MonoBehaviour
{
    public Image MIIMAGEN, MIIMAGEN1;

    public void TexRack()

    { MIIMAGEN.gameObject.SetActive(true);}

    public void TeC1()

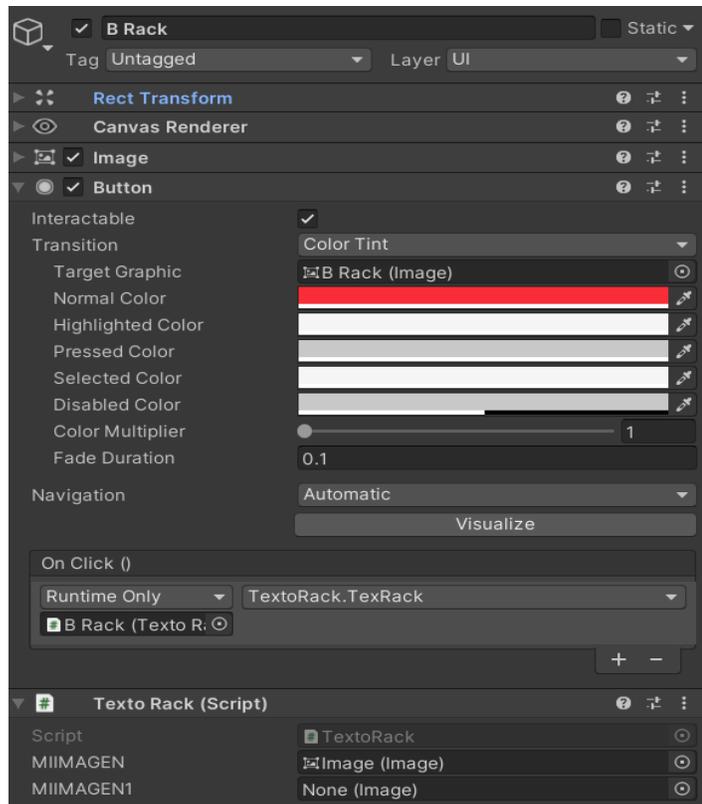
    { MIIMAGEN1.gameObject.SetActive(true);}

}

```

Figura 15.

Vista programación grafica del botón configuración del rack



Fuente. El autor

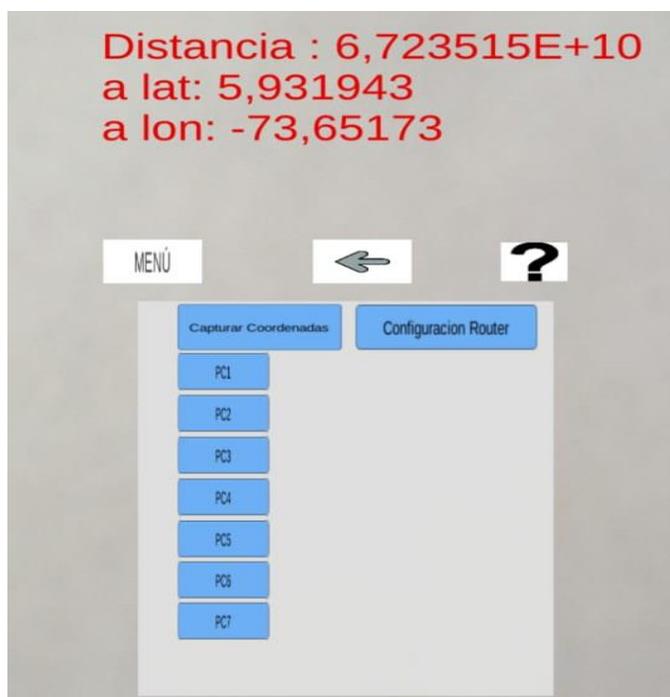
Pruebas

Prueba De Capturar Coordenadas

A continuación, se presenta la evidencia del funcionamiento del algoritmo de captura de coordenadas, que en este caso funciona tomando como referencia tres pares de coordenadas correspondientes a las esquinas de la sala que se utilizó como modelo.

Figura 16.

Evidencia captura de coordenadas



Fuente. El autor

Prueba De Reconocimiento De Planos

A continuación, se presenta la evidencia del funcionamiento del reconocimiento de plano ejecutado en la aplicación por medio del SDK de Vuforia Ground Plane. Este SDK reconoce una superficie y cuando la identifica proyecta un cuadrado sobre el que se da clic y se monta la realidad aumentada.

Figura 17.

Evidencia reconocimiento de plano



Fuente. El autor

Prueba De Modelos 3d

A continuación, se presenta la evidencia del funcionamiento del algoritmo utilizado para superponer los modelos 3d en la realidad dependiendo de la información de distancia obtenida gracias a los pares de coordenadas.

Figura 18.

Evidencia montaje de modelos 3d Router



Fuente. El autor

Figura 19.

Evidencia de modelos 3d laterales



Fuente. El autor

Figura 20.

Evidencia de sala modelos 3d



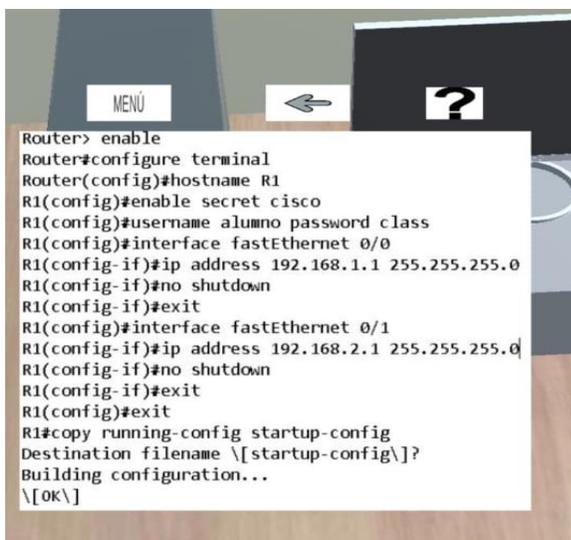
Fuente. El autor

Prueba Vista De Configuraciones

A continuación, se presenta la evidencia del funcionamiento de los botones correspondientes a la configuración del router, configuración de los equipos y el botón de ayuda.

Figura 21.

Evidencia configuración del router



```

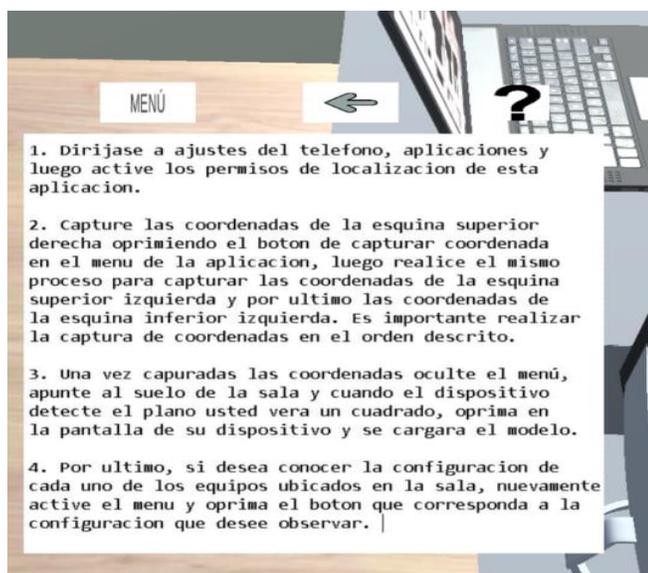
Router> enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#enable secret cisco
R1(config)#username alumno password class
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface fastEthernet 0/1
R1(config-if)#ip address 192.168.2.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#exit
R1#copy running-config startup-config
Destination filename \[startup-config\]?
Building configuration...
\[OK\

```

Fuente. El autor

Figura 22.

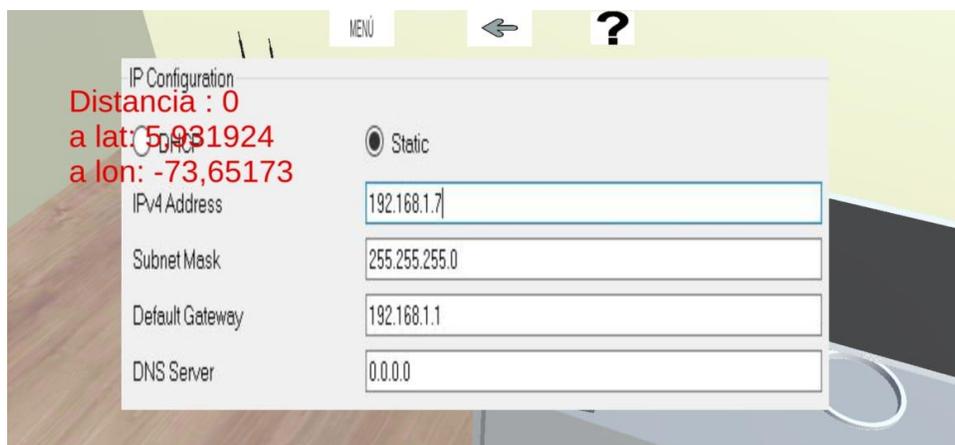
Evidencia botón de ayuda



Fuente. El autor

Figura 21.

Evidencia configuración del PC



Fuente. El autor

Prueba Final

En el siguiente link encontrará un video de prueba donde se observará el funcionamiento de la aplicación en tiempo real, mostrando cada uno de los aspectos esenciales de la misma y explicando detalladamente su funcionamiento. Además, se realizan pruebas en diferentes espacios para ver como se comporta la aplicación en diferentes entornos con diferentes coordenadas.

MACIAS GARCIA, K. RedAR. [Video]. Colombia, Youtube, 2021. 1 minuto 48 segundos. Recuperado de <https://www.youtube.com/watch?v=60dxbrZQjsw>

Lineamientos De Calidad

Los lineamientos presentados a continuación son los lineamientos establecidos por la comunidad de desarrolladores Android. Estos lineamientos permiten evaluar distintos aspectos que determinaran la calidad de la aplicación a implementar. Estos lineamientos evalúan los aspectos de experiencia visual, funcionalidad, rendimiento, estabilidad, privacidad y seguridad.

Se les asignó un porcentaje de cumplimiento que permite valorar el nivel de calidad de la aplicación y así determinar si cumple con los requisitos básicos de publicación.

Tabla 1.

Lineamientos de calidad

Área	Prueba	Descripción	%	De
				Cumplimiento
Navegación	CR-3	La app admite la navegación estándar del botón Atrás y no utiliza avisos personalizados en pantalla para este.	100	
Navegación	CR-1	La app preserva y restaura correctamente el estado del usuario o la app.	100	
Notificaciones	CR-9	Las notificaciones no se utilizan para realizar promoción cruzada ni para publicitar otro producto, ya que Play Store lo prohíbe de forma estricta.	100	
IU y gráficos	CR-5	La app admite tanto la orientación horizontal como la vertical (si fuera posible). Las orientaciones exponen ampliamente las mismas funciones y acciones, y conservan la paridad funcional. Se aceptan cambios mínimos en el contenido o las vistas.	90	

IU y gráficos	CR-5	La app controla correctamente las transiciones rápidas entre las orientaciones de la pantalla sin presentar problemas de renderización ni perder el estado.	85
Calidad visual	CR-all	La app muestra gráficos, texto, imágenes y otros elementos de la IU sin distorsión, esfumado ni pixelado notables.	80
Servicio en segundo plano	CR-10	La app admite correctamente las funciones de administración de energía que se introdujeron en Android 6.0 (Descanso y App Standby). Cuando se interrumpe la funcionalidad central por la gestión de energía, solo apps calificadas pueden solicitar una exención.	100
Estabilidad	CR-all SD-1	La app no falla ni bloquea el subproceso de IU que provoca errores ANR ("Android no responde").	80
Rendimiento	CR-all SD-1	La app se carga rápidamente o proporciona al usuario comentarios en pantalla (como un indicador de	80

		progreso o una señal similar) en el caso de que tarde más de dos segundos en cargarse.	
SDK	CR-0	La app se ejecutará en la última versión pública de la plataforma de Android sin que se produzca una falla y sin que haya un impacto sobre la funcionalidad principal.	100
Permisos	SC-4	La app solicita solo la mínima absoluta de permisos que necesita a fin de admitir el caso de uso en cuestión. Para algunos permisos, como la ubicación, usa la ubicación aproximada en lugar de la precisa si es posible.	100
Datos y archivos	SC-10	No se registran datos personales o sensibles de los usuarios en el registro del sistema ni en un registro específico de la app.	100
Bibliotecas	SC-U1	Todas las bibliotecas, los SDK y las dependencias están actualizados.	90

Fuente. El autor

Metodología

Etapa 1

Planeación: recopilar información que permita Identificar los requerimientos del sistema entre ellos: el lenguaje de programación más adecuado para su aplicación en la realidad aumentada, los dispositivos necesarios y los procedimientos a utilizar.

Etapa 2

Diseño: desarrollar un prototipo en el cual se incluya la interfaz gráfica bosquejando los elementos que se incluirán en el aplicativo entre ellos: menú, objetos, etc; así como la funcionalidad básica del software.

Etapa 3

Codificación: teniendo en cuenta el lenguaje de programación elegido se procede a escribir el código fuente de la aplicación; además se realiza la compilación del código fuente en una App Android.

Etapa 4

Pruebas: Prueba del funcionamiento del aplicativo en dispositivos móviles Android de tipo unitario, corrección de errores y aceptación.

Recursos

Tabla 2.

Recursos

Recurso	Descripción	Presupuesto
1. Equipo Humano	Ingeniero de Sistemas, estudiante de ingeniería de sistemas	0
2. Equipos y Software	Computadores, Java, Unity	2000000
3. Viajes y Salidas de Campo	Viajes al Cead Vélez	200000
4. Materiales y suministros	N/A	0
5. Bibliografía	Documentación electrónica de fuentes confiables	0
TOTAL: 2200000		

Fuente. El autor

Resultados

Tabla 3.

Resultados

Resultado	Indicador	Beneficiario
Entorno grafico para la interacción del usuario con el dispositivo móvil.	Entorno grafico	Estudiantes UNAD Cead Vélez
Código fuente que permita procesar la información recibida en el dispositivo.	Algoritmo	Estudiantes UNAD Cead Vélez
Aplicación móvil para dispositivos Android.	Software	Estudiantes UNAD Cead Vélez

Fuente. El autor

Conclusiones

Se realizó el análisis y la identificación de las necesidades del aplicativo utilizando los cuestionarios como técnica para obtener el documento de requisitos; se obtuvo un documento con los requisitos de la aplicación y este sentó las bases para el posterior diseño de esta. Se logró diseñar los elementos de la aplicación móvil, para la enseñanza de redes LAN, haciendo uso del lenguaje de programación Java para sistema operativo Android; cumpliendo con los requisitos propuestos inicialmente. Además, se evaluó la funcionalidad y usabilidad del aplicativo mediante los lineamientos básicos de calidad para desarrolladores Android lo que permitió que el software consiguiera una calidad aceptable.

Por otro lado, este trabajo permitió aprender los conceptos básicos de la realidad aumentada, entre ellos su definición, los elementos que la componen, los lenguajes de programación que permiten la creación de contenido con realidad aumentada y las librerías utilizadas para ello. Por otra parte, este trabajo permitió adquirir habilidades en el manejo de codificación cuyo fin es obtener información de entornos físicos como la adquisición de coordenadas y el cálculo de la distancia de una sala mediante estas.

También, logró mejorar las habilidades de diseño con el fin de realizar una interfaz gráfica atractiva para la aplicación, de modo que quienes la utilicen se sientan a gusto y visualmente les genere una sensación de bienestar. Por último, este trabajo permitió reforzar el conocimiento en las funciones básicas de una red y la forma en que debe diseñarse y estructurarse dicha red, así como los conceptos básicos a seguir para su correcta configuración.

Referencias

- Arredondo, S. J. (2010). *Desarrollo de un modelo de ambiente virtual que integre el uso de la Realidad Aumentada para el proceso de Enseñanza-Aprendizaje de la asignatura Fundamentos de Programación I*. Politécnico Colombiano Jaime Isaza Cadavid. <http://eduteka.icesi.edu.co/gp/upload/5fb29c87337686de2bc60fc7e4037338.pdf>
- Blázquez, Alegría (2017) *Realidad Aumentada En Educación*, Universidad Politécnica De Madrid. Gabinete de Tele-Educación. http://oa.upm.es/45985/1/Realidad_Aumentada_Educacion.pdf
- Bong, S (2012). *Nyartoolkit*. ArtoolWorks. <http://www.artoolworks.com/products/desk-top/nyartoolkit>
- Boronat, S. F., & Montagud, C. M. (2013). *Direccionamiento e interconexión de redes basada en tcp/ip : Ipv4/ipv6, dhcp, nat, encaminamiento rip y ospf*. Bblioteca virtual UNAD. <https://ebookcentral-proquest-com.bibliotecavirtual.unad.edu.co>
- Calidad básica de las apps. (2021). Developers. <https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=es-419>
- Carriles, O. (2012). *Desarrollo de aplicaciones JAVA para dispositivos móviles*. Universidad de Palermo. <https://www.palermo.edu/ingenieria/downloads/CyT%204/CYT407.pdf>

- Cava (2016) *Recursos Educativos Aumentados Una Oportunidad Para La Inclusión*, Editorial Tecnológico Comfenalco. <https://tecnologicocomfenalco.edu.co/wp-content/uploads/librosinvestigacion/recursos%20educativos%20abiertos.pdf>
- Craig, A. Mcgrath, R. Gutierrez, A (2011) *Technical Note: Augmented Reality Software Kits for SmartPhones*. Illinios.edu.
https://www.ideals.illinois.edu/bitstream/handle/2142/27688/AR_Smart_Phone_Note_rev3.pdf?sequence=2
- García, R. P. (2014). *Métodos de programación segura en Java para aplicaciones móviles en Android*. CIENCIA ergo-sum. <https://dialnet.unirioja.es/servlet/articulo?codigo=5035131>
- Gonzalez C., Vallejo D. & Albusac J (2012) *Realidad Aumentada, Un Enfoque Practico con ARToolkit y Blender*. Editorial IdenTIC.
http://www.librorealidadaugmentada.com/descargas/Realidad_Aumentada_1a_Edicion.pdf
- Sacristán, C. R., & Fernández, D. R. (2012). *Programación en Android*. Ministerio de Educación. <https://cutt.ly/tvbbeM0>
- Sanchez, J (2011) *Realidad Aumentada para el Aprendizaje de Ciencias en niños de Educación General Básica*. Universidad de Chile. <http://repositorio.uchile.cl/handle/2250/104314>