

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



PROTOTIPO DE APLICACIÓN BASADO EN REALIDAD AUMENTADA
PARA EL APOYO DEL APRENDIZAJE DEL LENGUAJE DE
PROGRAMACIÓN DEL LENGUAJE EN C

Reporte Técnico de Investigación presentado por:

Rodrigo Emmanuel Sánchez Benítez 122432
Jaime Guerra Vazquez 116973

Requisito para la obtención del título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Dra. Vianey Guadalupe Cruz Sánchez

Ciudad Juárez, Chihuahua, a 7 de noviembre de 2019

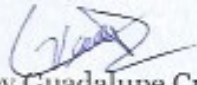
Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente.-

Por medio de la presente nos permitimos comunicarle que, después de haber realizado las asesorías correspondientes al reporte técnico Prototipo de Aplicación Basado en Realidad Aumentada para el Apoyo del Aprendizaje del Lenguaje de Programación en C, los alumnos Rodrigo Emmanuel Sánchez Benítez y Jaime Guerra Vázquez, de la Licenciatura en Ingeniería en Sistemas Computacionales, considero que lo han concluido satisfactoriamente, por lo que pueden continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente


Dra. Vianey Guadalupe Cruz Sánchez
Profesor Investigador de IIT



Ccp:
Coordinador del Programa de Sistemas Computacionales
Jaime Guerra Vázquez
Archivo

Ciudad Juárez, Chihuahua, a 7 de noviembre de 2019

Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente.-

Por medio de la presente nos permitimos comunicarle que, después de haber realizado las asesorías correspondientes al reporte técnico Prototipo de Aplicación Basado en Realidad Aumentada para el Apoyo del Aprendizaje del Lenguaje de Programación en C, los alumnos Rodrigo Emmanuel Sánchez Benítez y Jaime Guerra Vázquez, de la Licenciatura en Ingeniería en Sistemas Computacionales, considero que lo han concluido satisfactoriamente, por lo que pueden continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente


Dra. Viancy Guadalupe Cruz Sánchez
Profesor Investigador de IIT



Ccp:
Coordinador del Programa de Sistemas Computacionales
Rodrigo Emmanuel Sánchez Benítez
Archivo

Ciudad Juárez, Chihuahua, a 7 de Noviembre de 2019

Asunto: Autorización de publicación

C. Jaime Guerra Vázquez

Presente.-

En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la publicación del documento de Prototipo de Aplicación Basado en Realidad Aumentada para el Apoyo del Lenguaje de Programación en C, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería en Sistemas Computacionales.

Sin otro particular, reciba un cordial saludo.



Mtra. Ivonne Haydee Robledo Portillo
Profesor Titular de Seminario de Titulación II

Ciudad Juárez, Chihuahua, a 7 de Noviembre de 2019

Asunto: Autorización de publicación

C. Rodrigo Emmanuel Sánchez Benítez

Presente.-

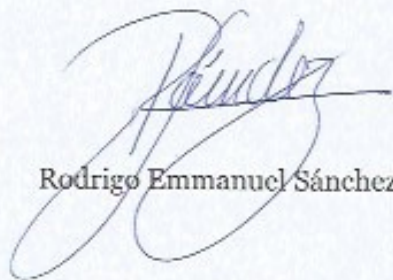
En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la publicación del documento de Prototipo de Aplicación Basado en Realidad Aumentada para el Apoyo del Lenguaje de Programación en C, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería en Sistemas Computacionales.

Sin otro particular, reciba un cordial saludo.

Mtra. Ivonne Haydee Robledo Portillo
Profesor Titular de Seminario de Titulación II

Declaración de Originalidad

Nosotros, Rodrigo Emmanuel Sánchez Benítez, Jaime Guerra Vázquez, declaramos que el material contenido en esta publicación fue elaborado con la revisión de los documentos que se mencionan en el capítulo de Bibliografía, y que la solución obtenida es original y no ha sido copiada de ninguna otra fuente, ni ha sido usada para obtener otro título o reconocimiento en otra institución de educación superior.



Rodrigo Emmanuel Sánchez Benítez



Jaime Guerra Vázquez

Agradecimientos

Agradezco a Dios, a mis padres y hermanas por haberme forjado como la persona que soy en la actualidad; muchos de mis logros también se los quiero agradecer a mis amigos que conocí en mi estancia en la Universidad, como también a esas personas que en su momento las consideré mis amigos, porque gracias a ellos y como lo mencioné antes he forjado como persona y he alcanzado mis logros, pero al final de cuentas, también me motivaron a ser alguien mejor. También, quiero agradecer a la Dra. Vianey Guadalupe Cruz Sanchez, por que a lo largo de este proyecto entre sus regaños y motivaciones estamos alcanzando una de nuestras metas como universitarios.

- Rodrigo Emmanuel Sanchez Benitez

Agradecimientos

Agradezco a Dios por guiarme en mi camino y por permitirme concluir con mi objetivo. A mis padres y hermanos quienes son mi motor y mi mayor inspiracion, que a traves de su amor, paciencia, buenos valores, ayudan a trazar mi camino. A mi esposa por ser el apoyo incondicional en mi vida, que con su amor y respaldo, me ayuda alcanzar mis objetivos. Por supuesto a mis amigos de la Universidad y a mi asesora Vianey Guadalupe Cruz Sanchez, por acompañarnos a concluir con concluir el proyecto, gracias por la paciencia, orientacion y guiarnos en el desarrollo de esta investigacion.

- Jaime Guerra Vazquez

Dedicatoria

Dedico todo este esfuerzo aquellas personas que estuvieron conmigo a lo largo de esta travesía llamada universidad, a esas personas que a lo largo de estos 6 años me hicieron retroceder en mis estudios, pero también a esas personas que me ayudaron avanzar cuando creía que no podría y que ahora los considero mis grandes amigos Jaime Guerra y Karla Jiménez. También, quiero agradecer a mi familia por este apoyo que me dieron a lo largo de estos 6 años, pero en especial a una persona que, aunque ya no está en este mundo siempre lo he llevado en mi mente y en mi corazón, a mi abuelo Francisco Sánchez Jiménez.

- Rodrigo Emmanuel Sánchez Benítez

Dedicatoria

Quiero dedicar este Proyecto de titulación especialmente a mis padres; Jaime Guerra Andrade y Erika Vazquez Jimenez que me han apoyado durante este proceso de formación profesional, a mi querida esposa Brenda Lizeth Vargas Rosales por su apoyo incondicional durante mi carrera y a mis amigos de la Universidad, que juntos emprendimos este viaje y juntos hemos logrado llegar hasta el final.

- Jaime Guerra Vázquez

Resumen

Se elaboró un prototipo de aplicación móvil basado en realidad aumentada para el apoyo del aprendizaje del lenguaje de programación en C, las actividades seleccionadas para el prototipo fueron conforme a una encuesta realizada para conocer los requerimientos y los temas a desempeñar. El prototipo se desarrolló a través del IDE Xcode mediante el lenguaje de programación Swift 5.2 anexando el paquete de realidad aumentada ARKit 2.0, los marcadores se generaron con la herramienta QR Generador para el enfoque de los objetos 3D que se mostraron y los objetos 3D de cada una de las actividades que se mostraron en el prototipo se elaboraron en la herramienta SketchUp. Para probar su funcionamiento se realizó una prueba piloto con alrededor de 34 estudiantes de las carreras de ingeniería que tuvieran en su mapa curricular un curso de programación, los estudiantes probaron el prototipo con sus diferentes actividades de cada uno de los temas elegidos para el prototipo, una vez terminada las actividades, los estudiantes contestaron una encuesta para tener una retroalimentación sobre el uso del prototipo. Los resultados obtenidos fueron que el 60 % de los usuarios prefiere tener una herramienta como el prototipo de aplicación desarrollado para esta investigación. Alrededor del 50 % considera que la retroalimentación en cada una de las actividades que se realizaron en el prototipo es esencial para que el estudiante no deje inconcluso su aprendizaje de el lenguaje de programación y no tenga una idea errónea de como realizarlo posteriormente. En conclusión, los estudiantes realizaron las actividades de cada uno de los temas planteados en el prototipo de aplicación y teniendo así la retroalimentación para poder apoyar en el aprendizaje del lenguaje de programación en C teniendo una idea sobre lo que realizaron en cada una de las actividades y como sería una mejor práctica para la elaboración a futuro.

Palabras clave: Programación en C, Realidad aumentada, Swift, Xcode, prototipo de aplicación, móvil, Marcadores, cámara.

Abstract

A mobile application prototype based on augmented reality was developed to support the learning of C programming language, the activities selected for the prototype were included in a survey conducted to learn about the requirements and the issues to be carried out. The prototype was developed through the Xcode IDE using the Swift 5.2 programming language, attaching the ARKit 2.0 augmented reality package, the markers were generated with the QR Generator tool for the focus of the 3D objects that were shown and the 3D objects of each One of the activities that were shown in the prototype were developed in the SketchUp tool. To test its operation, a pilot test was carried out with around 34 students of engineering careers who had a programming course on their curriculum map, the students tested the prototype with their different activities on each of the themes chosen for the prototype, Once the activities were finished, the students conducted a survey to have a feedback on the use of the prototype. The results obtained were that 60 % of users prefer to have a tool like the prototype application developed for this investigation. Around 50 % believe that feedback on each of the activities that were carried out in the prototype is essential so that the student does not leave his learning of the programming language unfinished and does not have a misconception of how to do it later. In conclusion, the students carried out the activities of each of the topics raised in the application prototype and thus having the feedback to be able to support in the learning of the programming language in C having an idea about what they did in each of the activities and how it would be a better practice for future elaboration.

Keywords: C Programming, Augmented Reality, Swift, Xcode, application prototype, mobile, Markers, camera.

Índice general

1. Introducción	1
1.1. Antecedentes	2
1.1.1. Un entorno de aprendizaje de realidad mixta	2
1.1.2. La influencia de la educación de programación basada en el juego en el pensamiento algorítmico	2
1.1.3. La realidad aumentada en la educación: las tecnologías actuales y el potencial de la educación	3
1.1.4. Verificator: herramienta educativa para la programación de aprendizaje	3
1.1.5. Juego de ordenador como herramienta de aprendizaje y enseñanza para la programación orientada a objetos en la institución de educación superior	4
1.2. Definición del problema	4
1.3. Objetivos	5
1.4. Justificación	5
1.5. Alcances y limitaciones	6
1.5.1. Limitaciones	6
1.5.2. Alcances	6
2. Marco Teórico	7
2.1. Clasificación de los medios para el uso de realidad aumentada	8
2.1.1. <i>Tracking</i> basado en marcadores	8
2.1.2. Tracking sin marcadores	9
2.2. Niveles de realidad aumentada	9
2.3. Campos de aplicación	10
2.3.1. Realidad aumentada en juegos	10
2.3.2. Realidad aumentada en ventas	11
2.3.3. Realidad aumentada en mantenimiento	12
2.4. Marco tecnológico	12

3. Desarrollo	17
3.1. Descripción del área de estudio	17
3.2. Herramientas	17
3.2.1. Herramientas primarias	17
3.2.2. Herramientas secundarias	18
3.3. Metodología de desarrollo	19
3.4. Análisis de datos para el prototipo	21
3.5. Diseño del prototipo	22
3.6. Desarrollo del prototipo	23
3.6.1. Interfaz de usuario	23
3.6.2. <i>Main storyboard y view vontrollers</i>	24
3.6.3. Ventanas	26
3.6.4. Desarrollo de figuras 3D	26
3.6.5. Implementación de realidad aumentada	28
3.6.6. Desarrollo físico del manual e interacción con el usuario	34
3.6.7. Diagrama de usabilidad	35
4. Resultados y discusiones	39
4.1. Prototipo desarrollado	39
4.2. Resultados obtenidos, interpretación y discusión	39
5. Conclusiones	48
5.1. Objetivos cumplidos	48
5.2. Trabajos a futuro	49
Bibliografía	50
A. Manual de usuario	53
B. Código del prototipo móvil	68

Índice de figuras

2.1. Animación con realidad aumentada.	7
2.2. Marcas empleadas para sistemas de tracking.	8
2.3. Ejemplo sistemas de tracking sin marcas.	9
2.4. Ejemplo de juego con realidad aumentada.	10
2.5. Realidad aumentada en las tiendas.	11
2.6. Gafas de realidad aumentada identificando componentes.	12
3.1. Pasos para la implementación del prototipo.	19
3.2. Timeline de desarrollo.	20
3.3. Temas de programación de C bases para aprender a programar.	21
3.4. Lenguajes de programación predominantes en ingeniería.	21
3.5. Modelo de prototipo en la IDE.	23
3.6. <i>View Controllers</i>	24
3.7. Código general que se utiliza en los <i>View Controllers</i>	25
3.8. <i>Navegation controller</i>	25
3.9. Código para representar alertas.	26
3.10. Creación de esfera en software SketchUP.	27
3.11. Laptop modelada en 3D.	27
3.12. Función que analiza y muestra la figura acorde al marcador reconocido.	28
3.13. Objeto 3D aplicando la realidad aumentada.	29
3.14. Menú principal con los temas principales del prototipo de aplicación.	29
3.15. Menú secundario mostrando de los subtemas del prototipo de aplicación.	30
3.16. Mensaje de bienvenida y tema seleccionado.	30
3.17. Menú de los subtemas seleccionados - if / else.	31
3.18. Menú de los subtemas seleccionados - Operaciones en C.	32
3.19. Objeto 3D diseñado para el subtema de operaciones matemáticas.	32
3.20. Objeto 3D diseñado para el subtema secuencia de escape.	33
3.21. Ejemplo del manual.	34
3.22. Caso de uso del Tema 1 ->Subtema 1	36
3.23. Caso de uso del Tema 1 ->Subtema 2	37

3.24. Caso de uso del Tema 1 ->Subtema 3	37
3.25. Caso de uso del Tema 2	38
3.26. Caso de uso del Tema 3	38
4.1. Género.	40
4.2. Semestres que actualmente los 34 alumnos encuestados cursan.	41
4.3. Ingenierías encuestadas.	41
4.4. Uso de tecnología como complemento en los cursos.	42
4.5. Considerar si la tecnología de RA es suficiente para complementar los cursos impartidos de programación.	43
4.6. Nuevas tecnologías.	43
4.7. El prototipo fue facil de utilizar.	44
4.8. Comprensión del tema.	45
4.9. La retroalimentación ayudó a terminar las actividades.	45
4.10. Cambios en el prototipo.	46
4.11. Utilizarían como complemento para el curso de programación.	47

Índice de tablas

3.1. Herramientas primarias.	18
3.2. Herramientas secundarias.	18

Capítulo 1

Introducción

En la actualidad, la educación en México ha estado en constante cambio, en donde el estudiante ha dejado de ser una persona pasiva a una activa. Es decir, el estudiante no sólo recibe la información del profesor, sino que participa activamente en su proceso de aprendizaje buscando alternativas que permitan complementar su educación. No obstante, el modelo educativo [1], pide a las escuelas una mayor autonomía de gestión para que el alumno pueda buscar herramientas que complementen su aprendizaje. Por lo tanto, es en las aulas donde se observa cierto desinterés en las clases, derivado de factores tales como, la falta de interactividad existente en el aula.

La tecnología está presente en todos lados, en la educación se ha vuelto fundamental para el aprendizaje y adquisición de habilidades de los alumnos. Incorporarla en la educación aporta recursos que ayudan a mejorar la eficiencia y la productividad en el aula, así como a lograr aumentar el interés de los alumnos en sus actividades en clase. La enseñanza tradicional ha quedado atrás, ya que actualmente se tiene acceso a la información, comunicación con los profesores, actividades y clases en espacios virtuales, dando un giro completo a los métodos de enseñanza [2].

Actualmente, se usan distintas técnicas para que los alumnos aprendan, adquieran habilidades y se retroalimenten sobre los distintos temas de los lenguajes de programación que cada uno estudia. Algunas de las técnicas son: reescribir los algoritmos con lápiz y hoja de papel, usar diagramas de flujo, gestores de algoritmos y nuevas técnicas que aprovechan las tecnologías de realidad virtual y realidad aumentada. Una de las tecnologías que está teniendo impacto en la educación es la realidad aumentada ya que añade contenido digital al mundo real y se puede aprovechar como estrategia educativa para producir novedades en la forma de enseñar [3]. En [4] se menciona lo siguiente: numerosas han sido las investigaciones que sugieren que la realidad aumentada refuerza el aprendizaje e incrementa la motivación por aprender. Lo que se menciona en el artículo se debe a que aplicaciones de esta naturaleza crean interacciones con el usuario, ya que por lo general combinan modelos 3D, texto, imagen, audio y vídeo en la percepción del mundo real del usuario.

En el presente documento se plantea el desarrollo de un prototipo de aplicación móvil el

cual integra realidad aumentada para poder aportar una retroalimentación a los métodos de enseñanza de el lenguaje de programación en C.

1.1. Antecedentes

Actualmente, existen muy pocas aplicaciones que utilizan la realidad aumentada como una herramienta para la enseñanza de algún lenguaje de programación. A continuación, se describen aplicaciones y proyectos que, aunque algunos no cuentan con la tecnología de realidad aumentada, su objetivo es la enseñanza de la programación.

1.1.1. Un entorno de aprendizaje de realidad mixta

En [5], habla sobre la interacción del ser humano y la tecnología en la industria maquiladora (producción). Precisamente, en la industria se aplica el concepto de realidad mixta, donde los entornos de aprendizaje abordan desafíos, así como, la retroalimentación de información. Lo que se propuso en la investigación fue incorporar a principiantes y personas, que no estaban directamente relacionadas con el entorno de producción, así, el contenido que se propuso no fue solo para encontrar una solución, se utilizó para encontrar varias soluciones y la más factible para la producción.

La investigación llegó a la conclusión de que los nuevos paradigmas y las tecnologías, pueden abrir paso a nuevos entornos, en los cuales las empresas maquiladoras inviertan en la capacitación avanzada de los empleados. De lo contrario, no se podrá aprovechar al 100 % las habilidades que los empleados tengan para la empresa y los beneficios resultantes. El artículo presentó una arquitectura del concepto de realidad aumentada y un espacio de aprendizaje de una realidad. La realidad aumentada permite manejar una complejidad experimentable con estos usuarios, que fue comprensible al entorno de producción, lo mismo que se quiere lograr con el presente proyecto aplicado a los alumnos.

1.1.2. La influencia de la educación de programación basada en el juego en el pensamiento algorítmico

En [6], menciona que el método tradicional que se usa para aprender a programar no es tan eficaz. La solución que proponen es utilizar los juegos para enseñar a programar en C y demuestran a través de exámenes como resulta ser eficaz el método empleado, con respecto al método tradicional. El enfoque es interesante, ya que los juegos captan la atención de los jóvenes universitarios, a la par de que aprovechan a enseñar cómo programar. El juego en mención se llama *LEGO Mindstorm*, el cual consiste en programar un robot, para que realice acciones que el usuario quiere que ejecute. El juego ayuda a mejorar las habilidades de programación, ya que corrige errores de sintaxis y es esencial para alumnos con dificultades de programación.

En conclusión, la idea presentada en el artículo es buena, utilizar juegos para enseñar a programar aporta otra variable a los distintos métodos para aprender un lenguaje de programación. La desventaja, es que no es flexible para transportarlo a diferentes lugares, ya que cada juego tiene diferentes estructuras; en cambio la presente propuesta pretende utilizar un dispositivo móvil para que la aplicación pueda ser utilizada en cualquier lugar.

1.1.3. La realidad aumentada en la educación: las tecnologías actuales y el potencial de la educación

En [7], proporciona una introducción a la tecnología de la realidad aumentada y sus posibilidades para la educación. Una parte importante que se menciona es que una gran cantidad de interfaces carecen de ser instruccionales, esto es parte fundamental en la educación y sirven para que el usuario se familiarice y mejore la percepción con la aplicación. La información transmitida por los objetos virtuales ayuda a los usuarios a realizar tareas del mundo y es una forma importante de mejorar el aprendizaje.

El artículo, menciona de forma resumida las siguientes tecnologías para realidad aumentada:

- Head Mounted Displays: Gafas que incluyen una pantalla en la parte frontal del ojo para poder visualizar el entorno.
- Handheld Displays: Son dispositivos móviles que pueden ofrecer realidad aumentada gracias a que cuentan con cámaras integradas y un procesador capaz de lograr la realidad aumentada.
- Pinch Gloves: Son guantes que contienen sensores para permitir la interacción con los objetos 3D que se generan con la realidad aumentada, dándole un enfoque realista.

En conclusión, utilizar los dispositivos móviles en el presente proyecto es buena idea, ya que una gran parte de la población en México dispone de uno, la decisión de no agregar las gafas o los guantes es por el alto costo para el usuario. Además, sería incómodo para el usuario traer los guantes o las gafas en un lugar público.

1.1.4. Verificator: herramienta educativa para la programación de aprendizaje

En [8], es una interfaz para computadoras de escritorio que ayuda a adquirir habilidades de programación, el cual funciona como verificador de los errores de sintaxis que se cometen en el código. Solo funciona con el lenguaje de programación C++, y al igual que el presente proyecto el software está destinado para los estudiantes de primeros semestres de pregrado de las carreras de ingeniería que en su retícula tengan clases de programación. El objetivo

del trabajo fue anular los malos hábitos de los estudiantes cuando se trata de programar, así como, tratar de escribir programas sin tomar en cuenta la sintaxis ni la comprobación lógica.

Para finalizar, la desventaja del software es que ya existen verificadores de sintaxis para programar como *Visual Studio*, además en el presente proyecto de realidad aumentada, se incluirá en cada capítulo la verificación de sintaxis para logra fortalecer el aprendizaje del alumno.

1.1.5. Juego de ordenador como herramienta de aprendizaje y enseñanza para la programación orientada a objetos en la institución de educación superior

En [9], es un juego para usuarios que desean aprender a programar, las dinámicas que se encuentran en este juego son para la programación orientada a objetos. Este juego tiene la peculiaridad de asemejarse en su dinámica a un diagrama de flujo, en el que el usuario toma las decisiones y, conforme a lo elegido, se va creando el juego, lo que hace agradable el uso de la herramienta en su entorno que incluye efectos de sonido y el manejo de los personajes. El juego termina cuando los jugadores vencen al último jefe, lo cual les otorga recompensas a los usuarios para aprender con entusiasmo, de esta forma, se anima a los usuarios que sigan aprendiendo y mejoren sus habilidades en la programación.

A la conclusión que llegaron los desarrolladores del proyecto, fue que los juegos de computadora pueden ser buenas herramientas de aprendizaje y de enseñanza para la programación, en específico a la de orientada a objetos. Sin embargo, para el desarrollo pleno de este videojuego existieron muchos desafíos, como es la mejora continua, ya que cada día se actualizan los métodos ya existentes.

1.2. Definición del problema

En la Universidad Autónoma de Ciudad Juárez(UACJ), específicamente en el Instituto de Ingeniería y Tecnología(IIT), existen programas educativos que en su mapa curricular cuentan con algún curso de programación, lo cual ayuda a fomentar habilidades tales como la resolución de problemas, la creatividad, entre otras. Sin embargo, en la mayoría de las instituciones el método tradicional de enseñanza es aún mediante un libro, o bien, con clases de programación donde la mayoría de los casos son presenciales. Factores tales como:

- Un mal entendimiento de los conceptos de programación.
- Aprendizaje auditivo, no visual.
- Poco tiempo destinado al tema, sujeto a horarios de clase.

Son algunos de los factores encontrados en las aulas, destinadas al aprendizaje de un lenguaje de programación que dificultan la comprensión de temas tan relevantes como son las funciones, sentencias, etc.

1.3. Objetivos

A continuación, se mencionarán los objetivos a desarrollar en este prototipo de aplicación.

Objetivo general:

Desarrollar un prototipo de aplicación para dispositivos móviles en iOS utilizando la realidad aumentada para apoyar el aprendizaje del lenguaje de programación en C para estudiantes de pregrado de las carreras de ingeniería.

Objetivos específicos:

- Desarrollar un prototipo de aplicación en iOS para dispositivos móviles.
- Diseñar objetos 3D y marcadores para el reconocimiento en los dispositivos mediante la tecnología de la realidad aumentada.
- Realizar pruebas de funcionalidad del prototipo con estudiantes de pregrado de las carreras de ingeniería.
- Elaboración de un material de apoyo para el uso del prototipo.

1.4. Justificación

Actualmente, se usan libros para aprender un lenguaje de programación como también asisten a clases de programación. Parte de la justificación es hacer la combinación de estos dos factores y apoyar al alumno a comprender los conceptos del lenguaje de programación en C, como se mencionó en la definición del problema, este lenguaje es uno de los temas que se imparten inicialmente en las carreras de ingeniería de nivel superior de la UACJ. El presente prototipo de aplicación apoyará a los estudiantes de nivel superior de pregrado a tener una retroalimentación de algunos temas vistos en clase, el alumno interactuará con el prototipo de aplicación dependiendo donde lo permita el prototipo. La realidad aumentada integrará estilos de aprendizaje para crear diferentes métodos prácticos, es por ello la elección referente a la realidad aumentada para este proyecto. El prototipo de aplicación será de fácil acceso en cualquier dispositivo móvil con el sistema operativo iOS, ofreciendo una forma interactiva de aprender a programar en el lenguaje C, desarrollando habilidades y destrezas del usuario.

Para finalizar este punto, se tiene conocimiento de diferentes herramientas para el apoyo al usuario en la enseñanza de lenguajes de programación, lo que hace la diferencia de nuestro

prototipo con los demás es sobre la combinación de la tecnología de la realidad aumentada y la movilidad que se tendrá ya que estará destinada para un dispositivo móvil.

1.5. Alcances y limitaciones

En esta sección se puntualizarán los alcances y limitaciones que el prototipo tiene para este proyecto. Aspectos que se lograrán en la investigación y por parte de limitaciones se mencionan los aspectos quedan fuera de la cobertura del proyecto.

1.5.1. Limitaciones

1. El reconocimiento del marcador dependerá del enfoque, distancia e iluminación presente en el entorno.
2. El prototipo no podrá interactuar con otros libros del lenguaje C.
3. El lenguaje de programación en C es muy extenso en su campo de estudio, no podrán ser abarcados todos los temas.

1.5.2. Alcances

1. El prototipo sólo detectará marcadores diseñados para este proyecto.
2. El presente proyecto abarca el apoyo a estudiantes de nivel superior en la rama de ingeniería.
3. El prototipo solo funciona con un material de apoyo diseñado para este proyecto.
4. El prototipo de aplicación sólo se realizará en iOS.
5. La investigación se realizará en alumnos de la UACJ del instituto IIT de los primeros semestres de las carreras de ingeniería que cuenten con algún curso de programación en C.

Capítulo 2

Marco Teórico

En esta sección se hablará de la realidad aumentada y sus clasificaciones según la forma de detección de los marcadores en el entorno real, también se mencionan datos sobre el lenguaje de programación en C.

Realidad aumentada

La realidad aumentada es una tecnología en donde se trabaja en un entorno donde el usuario es capaz de observar el entorno físico, pero en este entorno se agrega el entorno virtual donde coexisten en uno solo, como lo es en un dispositivo capaz de soportar y presentar toda esa información virtual adicional.

Esta tecnología está haciendo introducida en nuevas áreas como lo son en la arquitectura, mercadotecnia y procesos en la industria [10]. Una de las formas para entender a la realidad aumentada es a través de los sentidos, específicamente de la vista, esto para percibir el mundo con otro enfoque. La realidad aumentada potencia los cinco sentidos complementando el mundo físico con el mundo digital [11]. La Figura 2.1 muestra un ejemplo de una aplicación de realidad aumentada.



Figura 2.1: Animación con realidad aumentada.

2.1. Clasificación de los medios para el uso de realidad aumentada

Existen tres medios para presentar la realidad aumentada, la computadora tradicional, los dispositivos móviles y equipos específicos de realidad aumentada [12]. En los dispositivos anteriores, se pueden usar diferentes tecnologías de seguimiento para realizar una aplicación de realidad aumentada, estas técnicas se les llama *tracking*. Este tipo de método tratan de obtener una estimación de la trayectoria en el espacio realizada por un objeto o sensor. Se pueden emplear diferentes tipos de sensores, basados en campos magnéticos, ondas sonoras o cámaras de visión.

2.1.1. *Tracking* basado en marcadores

Es uno de los métodos más utilizados en la tecnología de la realidad aumentada. Se utilizan marcadores por su eficiencia a la hora de ser detectados por la cámara, ya sea por una computadora, o bien, un dispositivo móvil, por su alto contraste de tonos negros y blancos. En la Figura 2.2 se muestran ejemplos de marcadores [13].

La fidelidad de estos marcadores ayuda en la precisión de los objetos que se presenta en los dispositivos, ya sea una presentación 2D o 3D [14]. Los marcadores deben contener un patrón único, esto permitirá que el dispositivo que contenga la cámara pueda observar y reconocer dicho marcador, esto permitirá mostrar el objeto u objetos que indique el marcador [15].

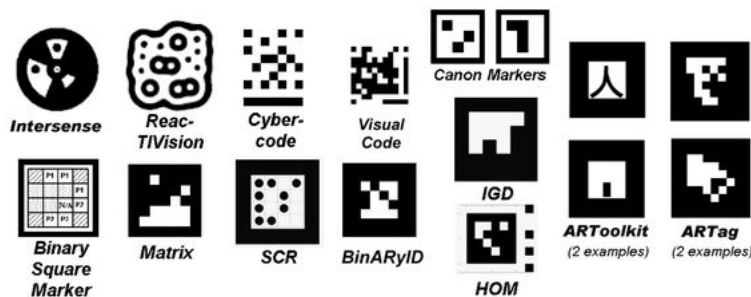


Figura 2.2: Marcas empleadas para sistemas de tracking.

2.1.2. Tracking sin marcadores

El método utiliza características del entorno real, un ejemplo son las estructuras físicas que se encuentren en la visión de la cámara. El procedimiento de captura de profundidad ese mediante un algoritmo que determina tres puntos en la imagen (en tipo de triángulo) para determinar primero en dos puntos el horizonte y el tercer punto la profundidad, mediante el concepto del estado del arte se puede calcular la posición del tercer punto [16]. Ver Figura 2.3.



Figura 2.3: Ejemplo sistemas de tracking sin marcas.

2.2. Niveles de realidad aumentada

La realidad aumentada posee diferentes niveles para representar la información, se clasifican según sea el grado de dificultad de integración y el tipo de tecnología para la implementación [4].

- Nivel 0:
En este nivel se realiza un escaneo de un código de barras o un código QR, el cual activa un hyperlink que contiene información sobre el tema en donde se identifica el código.
- Nivel 1:
En este nivel se escanea un marcador impreso y el dispositivo representa una figura 3D sobre el marcador.
- Nivel 2:
A diferencia del nivel 1, en este nivel no se utiliza un marcador para desplegar la realidad aumentada, se utilizan objetos del mundo real que actúan como activadores.

- Nivel 3:

Este nivel es llamado visión aumentada, se necesitan gafas especiales que ayudan a representar la realidad aumentada sin necesidad de un dispositivo móvil.

En el actual proyecto se combina el nivel 1 y el nivel 2 dependiendo del tema a tratar.

2.3. Campos de aplicación

La adaptación de la realidad aumentada a las acciones que se realizan en la vida cotidiana son amplias y han evolucionado con el paso del tiempo, a continuación se mencionará algunos de los diferentes ámbitos en donde a destacado esta tecnología [17].

2.3.1. Realidad aumentada en juegos

En los inicios del año 2000, se empezó a notar el potencial que tenía la realidad aumentada para los juegos y distintas universidades empezaron a desarrollar videojuegos clásicos de esa época. Pero tenía un inconveniente, para los usuarios era incómodo ya que tenían que llevar varios dispositivos para lograr experimentar la realidad aumentada, dificultando la usabilidad y experiencia de los usuarios como se ve en la Figura 2.4.

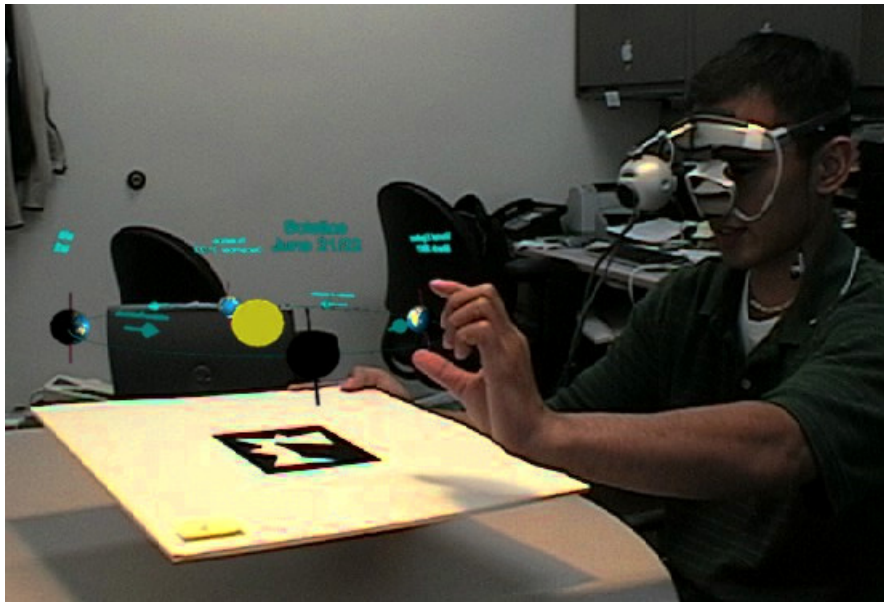


Figura 2.4: Ejemplo de juego con realidad aumentada.

Gracias al avance tecnológico, es posible crear videojuegos de realidad aumentada para los dispositivos móviles y lograr eliminar de tantos elementos que anteriormente se necesitaban. Las 3 empresas más reconocidas productoras de videojuegos Microsoft, Nintendo y Sony están apostando para que en los videojuegos ya no se utilicen los mandos, o que no sean el principal control, si no que el usuario utilice su cuerpo como mando principal.

2.3.2. Realidad aumentada en ventas

El ramo del marketing es donde se está explotando la realidad aumentada, ya que ofrece al usuario la posibilidad de acceder a experiencias visuales llamativas. La ventaja de usar realidad aumentada en las ventas es que el usuario puede comprobar el resultado sin necesidad de probar.



Figura 2.5: Realidad aumentada en las tiendas.

2.3.3. Realidad aumentada en mantenimiento

En el ámbito de mantenimiento en producción industrial o automovilístico la realidad aumentada es empleada para la reparación y mantenimiento de autos o maquinaria, la funcionalidad es dar como guía los pasos a seguir en las actividades que se desempeñen en el objeto que se este entrenando. En conclusión, la realidad aumentada tiene un sinfín de aplicaciones que ayudan a los usuarios en su vida cotidiana, es por eso que se optó por esta tecnología para apoyar al aprendizaje del lenguaje C.

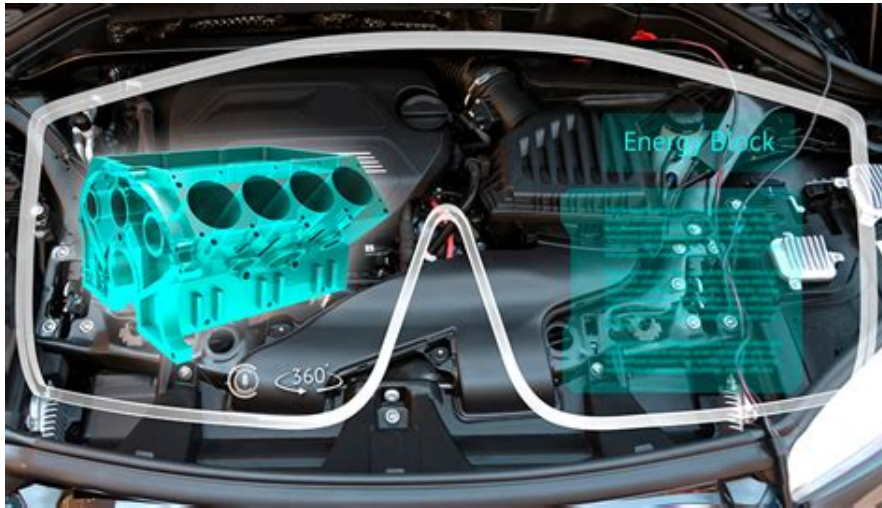


Figura 2.6: Gafas de realidad aumentada identificando componentes.

2.4. Marco tecnológico

Para el desarrollo del prototipo se evaluaron diferentes IDE dependiendo de las características que el proyecto requería para su elaboración. A continuación, se enunciarán las diferentes IDE evaluadas para el proyecto.

- **Microsoft Visual Studio**

Microsoft Visual Studio tiende a mejorar habilidades y destreza en el manejo de las herramientas para el desarrollo de aplicaciones ya que permite el desarrollo en múltiples plataformas como lo son Android, iOS, Windows Phone, entre otros. Esta herramienta también ofrece diferentes lenguajes de programación para las plataformas antes mencionadas, el lenguaje predominante de Visual Studio es el lenguaje C#.

- **Xcode by Apple**

XCode IDE creada por la empresa Apple para el desarrollo de aplicaciones móviles y de escritorio exclusivas de la empresa (Mac mini, iMac, iPad, iPod, iPhone, MacBook) mediante el lenguaje Swift 4.0 y Objective-C [18]. En el año 2017 Apple introdujo su librería de desarrollo de realidad aumentada ARKit para el sistema operativo móvil iOS 11, aunque esto limita el desarrollo de alguna aplicación en realidad aumentada de sistemas operativos anteriores a iOS, como también dispositivos móviles de la compañía que no tengan este sistema operativo iOS 11 [19]. ARKit de Apple tiene como base lo antes visto en otros kits de desarrollo como la detección de profundidad en imagen para el posicionamiento de los objetos, y gracias a los dispositivos móviles con el acelerómetro tendrá una mejor calibración al momento de interactuar con esta tecnología.

- **Android Studio**

Android Studio y SDK Tool, es una herramienta que ayuda a la creación de aplicaciones para el sistema operativo de Android. Android Studio es una plataforma gratuita donde se permitirá el desarrollo del prototipo ya que contiene una extensa variedad de librerías y una de esas librerías es ARKit de Android [20], una de las desventajas de este software es por su alto nivel de requerimiento de recursos de la computadora, esto hace tardado el uso para el desarrollo.

- **Unity 3D** Unity es un motor gráfico de reciente generación enfocado a los videojuegos y entornos de aprendizaje. Esta herramienta provee una gran accesibilidad para el desarrollo de gráficos, como también de programación. El lenguaje de programación más usual de este entorno gráfico es C#, eso da una buena ventaja para el desarrollo del material que se presentará en este prototipo de aplicación ya que se tiene noción de este lenguaje para el desarrollo [21], este programa cuenta con su propio *plugin* de ARKit, esto incluye la tecnología de Realidad Aumentada. Unity cuenta con una documentación de ARKit que ayudará para la implementación, la única desventaja el alto requerimiento de hardware y la suscripción para obtener mejores paquetes de actualización y *assets*.

- **ARkit**

Librería que permite la creación de aplicaciones de realidad aumentada, en las que sobreponer imágenes virtuales al mundo real. ARToolKit funciona mucho mejor con las cuatro herramientas antes mencionadas ya que por sí sola no se podrá desarrollar el prototipo de manera coherente. A continuación, se describen las clases utilizada para el desarrollo del prototipo las cuales forman parte de la librería Arkit:

- **class ARSession** Maneja los principales procesos que ARKit realiza para crear una experiencia de realidad aumentada. Estos procesos incluyen la lectura de datos del hardware de detección de movimiento del dispositivo, el control de la

cámara integrada del dispositivo y la realización de análisis en las imágenes de la cámara capturadas. La sesión sintetiza todos estos resultados para establecer una correspondencia entre el espacio del mundo real que habita el dispositivo y un espacio virtual donde se modela contenido de realidad aumentada [22].

- **class ARConfiguration** Objeto que define las características de ARKit concretas que se habiliten en la sesión en un momento dado [23].
- **class ARAnchor** Funciona para realizar un seguimiento de las posiciones estáticas y orientaciones de objetos reales o virtuales en relación con la cámara [24].
- **class ARFrame** Captura continuamente fotogramas de vídeo de la cámara del dispositivo, mientras que ARKit los analiza para estimar la posición del usuario en el entorno físico. Además, ARKit le proporciona esta información en forma de ARFrame y a la frecuencia de la velocidad de fotogramas de la aplicación [25].
- **class ARCamera** Obtiene información de la cámara de cada ARFrame que ARKit entrega[26].
- **class ARSCNView** Proporciona la forma más fácil de crear experiencias de realidad aumentada que combinan contenido 3D virtual con una vista de cámara de dispositivo del mundo real. Al ejecutar el objeto ARSession proporcionado por la vista[27].

■ UIKit

Para el desarrollo del prototipo se utilizará la librería UIKit, esta librería nos permitirá utilizar los recursos del IDE Xcode. Estos recursos son:

- View Controller

Nos permitirá tener una navegación completa entre view controllers (Escenas de presentación) mediante la clase de UINavigationController, esta clase enlazada con el view controller nos permite pasar de escena a escena sin la pérdida de información.

- Touches, presses y Gesturs

Estos tres recursos nos permitirán la interacción entre el usuario y el prototipo, mediante las clases de UITouch, UIResponder y UIPress.

- Keyboards y inputs

Keyboard y inputs son dos protocolos de ingreso de información por parte del usuario, mediante UITextInputDelegate podremos contener la información que el usuario ingrese en el prototipo.

Estos recursos están de la mano con el diseño del prototipo en la adaptación he implementación del código de los recursos.

- **SketchUp**

La característica principal de esta herramienta de desarrollo, se pueden realizar objetos 3D de forma sencilla, SketchUp en su recursos cuenta con tutoriales como apoyo para el desarrollo del diseño y modelado en el ambiente, además de programar incluye galerías de objetos, texturas e imágenes listas para su uso [28]. El desarrollo en este ambiente se emplea en el lenguaje de programación Ruby [29], Ruby es una lenguaje de programación definido por Yukihiro Matsumoto, donde mezcló partes de sus lenguajes de programación favoritos((Perl, Smalltalk, Eiffel, Ada y Lisp). Esta herramienta más el complemento de ARToolKit es una gran ayuda para el desarrollo del entorno para el prototipo.

- **Lenguaje de programación Swift**

En [30] nos describe la manera muy sencilla de utilizar el lenguaje de programación Swift y de como este lenguaje define las clases de errores en la programación que se pueden cometer comúnmente. También, este lenguaje está optimizado para poder aprovechar los recursos del hardware moderno. La sintaxis de este lenguaje de programación como las bibliotecas estándar manejadas están diseñadas para un mejor soporte al momento de crear una "Hola Mundo" hasta crear un sistema completo.

Las versiones estables de Swift son 4.0, 4.2 y 5.1 en estas versiones se puede hacer uso de el paquete de realidad aumentada ARKit sin tener problemas en la compilación. Es por esta razón que este lenguaje de programación es elegido para la creación del prototipo.

- **Lenguaje de programación Objective-C**

Como nos menciona en [31] el lenguaje de programación en Objective-C esta orientado a objetos que da el soporte al desarrollo de aplicaciones móviles y de escritorio de un lenguaje intuitivo al diseño que se realice para dicho sistema o prototipo. El lenguaje de programación es influenciado fuertemente por el lenguaje de programación en C y el viejo lenguaje de programación Smalltalk. Este lenguaje de programación es elegido por compatibilidad con el lenguaje de programación Swift en todas sus versiones.

- **Power BI**

Para el análisis de la información que obtenedemos en la seccion de resultados se utilizará la herremienta Power BI (Power Business Intelligence). Como menciona en [22] la herramienta de Power BI es dedicada para una solución para el análisis de datos a nivel empresarial. Power BI será utilizada para el análisis de la información de los resultados obtenidos en la prueba del prototipo. Esta herreminta permite una presnetación de los datos de manera mas dinámica que gráficas que podemos generar en otras herramientas de análisis de datos.

Las herramientas vistas en los anteriores puntos cuentan con el perfil necesario para soportar la tecnología de realidad aumentada. Pero una de los IDE que tiene mayor compatibilidad con el sistema operativo móvil que se utilizará, así como los dispositivos es la IDE Xcode. Esta IDE además de ser desarrollada por la compañía Apple con una lógica sencilla en la librería de realidad aumentada empleada para el uso óptimo de esta tecnología, así como también el lenguaje de programación Swift que es muy intuitivo y que no requiere demasiados recursos de hardware y de software.

Capítulo 3

Desarrollo

En esta sección se describe cómo se desarrolló el prototipo de aplicación esperada para el proyecto, empezando con una breve descripción del área de estudio que abarcó el proyecto y posteriormente se mencionarán las herramientas que se utilizaron, finalizando con los temas que se eligieron para presentarlo en el prototipo.

3.1. Descripción del área de estudio

El proyecto de investigación es de desarrollo tecnológico, enfocado en el área de investigación de realidad aumentada, con esta tecnología se desarrolló un prototipo de aplicación móvil para el apoyo en el aprendizaje del lenguaje de programación en C utilizando software gratuito en el desarrollo. La investigación está enfocada principalmente para alumnos de educación superior que estén cursando los primeros semestres de la carrera de ingeniería y que tengan como tronco común la materia de programación.

3.2. Herramientas

Las herramientas que se utilizaron en el proyecto se dividieron en dos secciones, herramientas primarias y herramientas secundarias. Esta división se empleó por el grado de impacto que la herramienta tuvo en la elaboración del prototipo de aplicación. A continuación se enunciarán las diferentes herramientas.

3.2.1. Herramientas primarias

Las herramientas implementadas en el desarrollo del prototipo de aplicación fue software gratuito, el hardware utilizado es de uso propio. A continuación, en la Tabla 3.1 se presentará el software y hardware utilizado en el desarrollo del prototipo de aplicación. Parte del software utilizado es Xcode para la programación del prototipo mediante el lenguaje de programación

Swift 4.3, Xcode cuenta con un simulador de cualquier dispositivo que se encuentre dentro de la familia Apple. También, Xcode cuenta con el sistema operativo necesario como es iOS. El Hardware utilizado este compuesto por terminales a usuario final, como Mac mini, MacBook, iPhone y cable USB Lightning.

Tabla 3.1: Herramientas primarias.

Software	Hardware
<i>Xcode 10</i>	Mac mini (<i>Mid 2010</i>)
<i>iOS 12</i>	MacBook (<i>Late 2009</i>)
Simulador <i>Xcode</i>	<i>iPhone 6S</i>
<i>MacOS 10.13.6</i>	Cable USB <i>Lightning</i>

3.2.2. Herramientas secundarias

Como parte de las herramientas utilizadas en el desarrollo del prototipo, las herramientas secundarias es un complemento adicional a lo listado en la Tabla 3.1 A continuación, en la Tabla 3.2 se mostrará el software adicional que se utilizó para el efecto visual del prototipo de aplicación.

Tabla 3.2: Herramientas secundarias.

Software
Adobe Illustrator
Adobe photoshop
QR generator
Sketchup 3D

El paquete Adobe fue utilizado para el diseño gráfico de toda la interfaz del prototipo de aplicación, por otro lado, *QR Generator* se utilizó como base para generar los marcadores y Sketchup 3D se utilizó para generar las figuras 3D utilizadas en este prototipo.

3.3. Metodología de desarrollo

La metodología en cascada se desarrolló en cinco fases, como se muestra en la Figura 3.1. Para una mejor comprensión de la metodología se utilizó el estándar de la IEEE 830-1998 [32], también conocido como modelo clásico, modelo tradicional o modelo lineal secuencial. El método de la cascada es considerado como el enfoque clásico para el ciclo de vida del desarrollo de sistemas, se puede decir que es un método puro que implica un desarrollo rígido. En el análisis se consideró reunir las necesidades del producto (prototipo de aplicación). En el diseño se describe la estructura interna del producto. En la implementación se describirá la etapa de codificación.

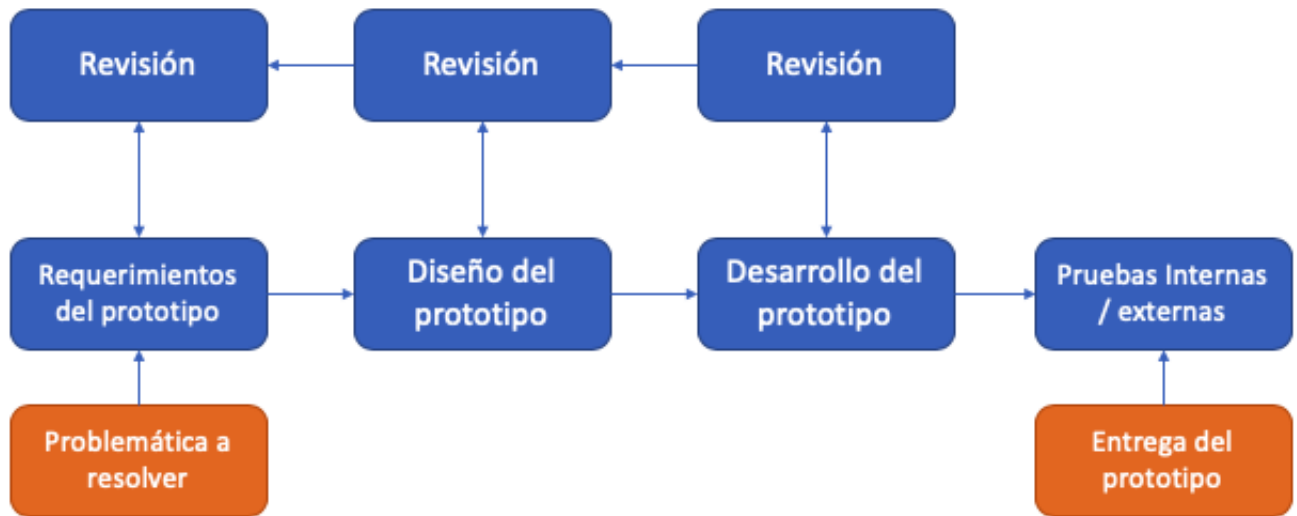


Figura 3.1: Pasos para la implementación del prototipo.

Parte del trabajo que se realizó para el desarrollo del prototipo se empleó bajo una cronología de trabajo que contó con 4 etapas. Estas etapas fueron, requerimientos, diseño del prototipo, desarrollo, y por último el uso de las pruebas internas y externas.

Primera etapa, constó saber sobre los requerimientos actuales, pero también basó en los trabajos ya realizados con realidad aumentada, esta etapa tuvo una duración de dos meses. La segunda etapa fue el diseño del prototipo y la parte de la funcionalidad esto fue de la mano con la etapa número tres del desarrollo basado en el diseño y los requerimientos necesarios para el uso, la etapa dos y tres tuvo una durabilidad de 4 meses de dos meses cada etapa. Por último, la cuarta etapa fue catalogada como de pruebas (*testing*) una vez que el desarrollo fue terminado se inició con las pruebas internas de las cuales se llevaron pruebas

de estrés a la aplicación para poder realizar las correcciones antes de las pruebas externas, estas pruebas internas tuvieron una duración de un mes, una vez que fueron corregidos los errores encontrados en las pruebas de estrés en el mes restante se llevó a cabo las pruebas con los usuarios. Parte de estas pruebas se podrán ver descritas en el Capítulo 4: Resultados y discusiones. En resumen, parte de las etapas fueron importantes unas con otras y esto lo podemos ver relacionado con la metodología que se empleó para este prototipo como se describió en el párrafo anterior. Las etapas antes descritas en la Figura 3.2.



Figura 3.2: Timeline de desarrollo.

3.4. Análisis de datos para el prototipo

La Figura 3.3 muestra los resultados obtenidos de una encuesta realizada con una muestra de 101 estudiantes, para conocer que temas de programación son base para aprender un lenguaje de programación. Conforme a los resultados, los temas que los usuarios eligieron que desearían aprender son los siguientes, Introducción a C, Operadores matemáticos en C, Sentencias en C.

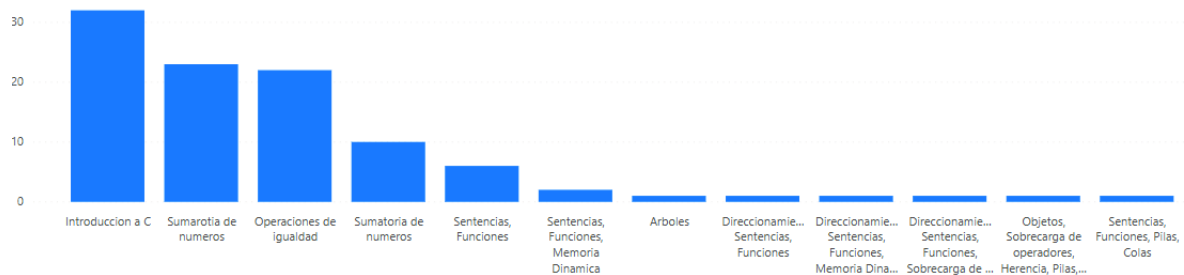


Figura 3.3: Temas de programación de C bases para aprender a programar.

Como se mencionó en uno de los puntos anteriores. Adicional a esto, se realizó una serie de encuestas a 101 personas dentro de las instalaciones de la UACJ en el IIT, parte de estos datos podemos constatar sobre el lenguaje de programación base para aprender a programar como se muestra en la Figura 3.4.

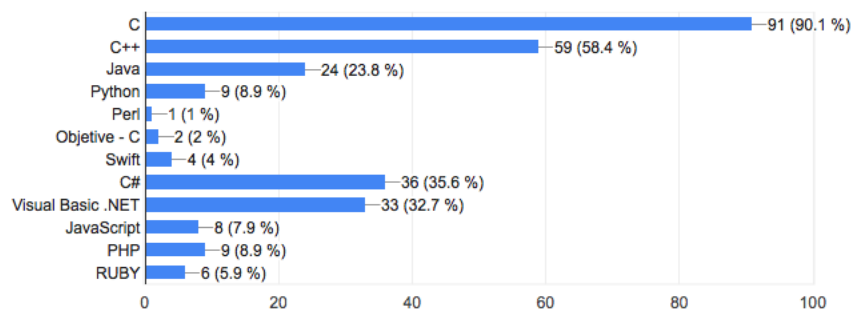


Figura 3.4: Lenguajes de programación predominantes en ingeniería.

3.5. Diseño del prototipo

El diseño del prototipo fue basado en la teoría de color para la creación de una interfaz llamativa, con el objetivo de que los usuarios estén cómodos.

Los temas que abarca el prototipo son:

- Tema 1: Introducción a la programación en C.
- Tema 2: Sumatoria de dos números enteros y conceptos de memoria.
- Tema 3: Operaciones de igualdad y relaciones.

Los temas anteriormente puntualizados fueron agregados al prototipo son de un nivel básico, la razón es porque el proyecto está destinado a estudiantes de pregrado interesados en el lenguaje de programación en C, sin conocimiento previo o con conocimiento básico. Estos temas tienen como propósito darles las herramientas básicas de programación y que ellos mismos continúen con su investigación de encontrar nuevas formas de programar, pero la única diferencia con la tecnología de RA sea más didáctica.

Como se mencionó en el párrafo anterior no se necesita tener conocimiento previo, y si el usuario cuenta con un conocimiento básico el prototipo es de gran ayuda para reforzar dicho conocimiento, es por eso que cada tema cuenta con subtemas para un mejor entendimiento del tema principal.

3.6. Desarrollo del prototipo

En el desarrollo del prototipo se llevó diferentes secciones de programación, esto se realizó para tener un orden para cada uno de los temas a elaborar en la IDE.

3.6.1. Interfaz de usuario

La IDE permite el *drag and drop* de componentes para la programación, permitiendo crear una interfaz amigable para el programador como se muestra en la Figura 3.5. Se utilizaron los softwares Illustrator y Photoshop del paquete de Adobe para diseñar las imágenes utilizadas en el prototipo y dar una apariencia agradable para la perspectiva del usuario.

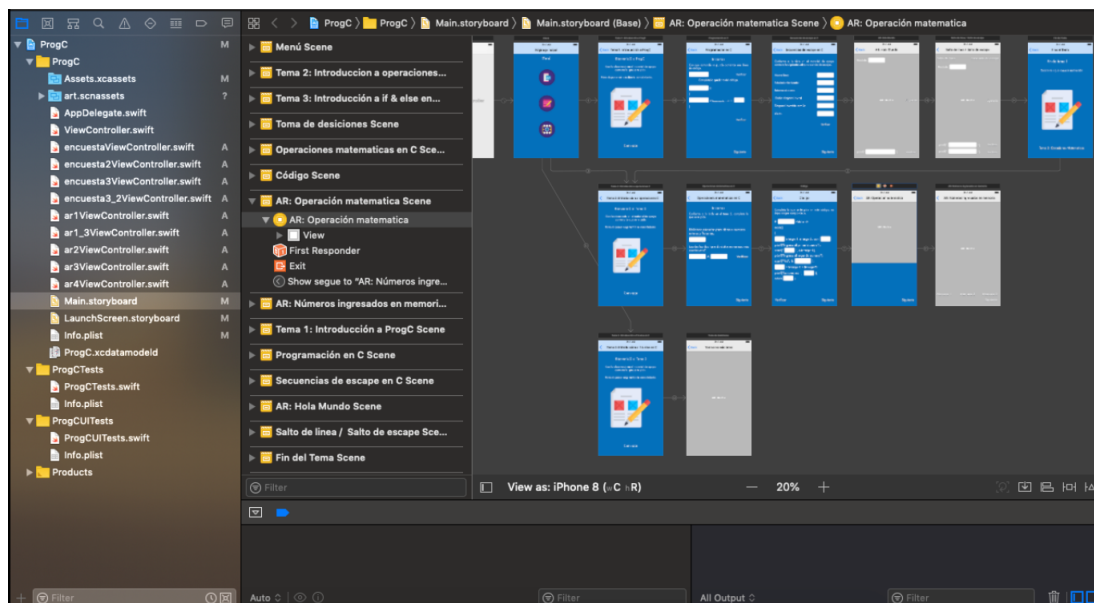


Figura 3.5: Modelo de prototipo en la IDE.

3.6.2. Main storyboard y view vontrollers

Como se mencionó en el párrafo anterior, Xcode permite un desarrollo amigable para el programador, storyboard es una herramienta fundamental en la IDE para mantener un orden de los View Controllers que se manejaron en el prototipo. ¿Qué son los View Controllers? Son las pantallas que el usuario observa e interactúa del prototipo móvil como se muestra en la Figura 3.6.



Figura 3.6: *View Controllers*.

Cada *View Controller* tiene su propia estructura para cargarse en la escena, el código se puede ver en la Figura 3.7. El flujo de control se llevó a cabo con la herramienta del IDE del *Navegation Controller*. ¿Qué es el *Navegation Controller*? son flechas de continuidad que permiten determinar qué *view controller* se activa en determinado momento como se muestra en la Figura 3.8, cada *View Controller* tiene su propia estructura. El *View Controller* tiene su propia clase llamada igual que el archivo del *View Controller* pero referenciando a los archivos importados.

```
import Foundation
import ARKit
import UIKit

class ar1ViewController: UIViewController, ARSCNViewDelegate, UITextFieldDelegate {

    @IBOutlet weak var sceneView: ARSCNView!
    @IBOutlet weak var btnSig: UIButton!
    @IBOutlet weak var txtMensaje: UITextField!
    @IBOutlet weak var txtLibreria: UITextField!
    @IBOutlet weak var btnVerificar: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()
        btnSig.isHidden = true

        let scene = SCNScene()
        sceneView.scene = scene
    }
}
```

Figura 3.7: Código general que se utiliza en los *View Controllers*.

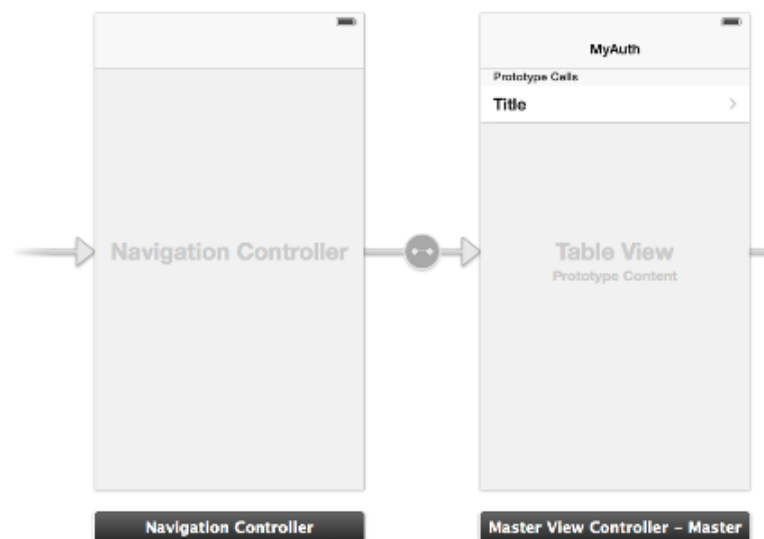


Figura 3.8: *Navegation controller*.

3.6.3. Ventanas

En el prototipo se usaron alertas de ventana para informar al usuario la acción que tiene que realizar según cada tema lo requiera. En Xcode se utiliza la función `UIAlertController` para representar las alertas de ventanas como se muestra en la Figura 3.9 donde se muestra tres ejemplos de funciones creadas para mostrar los tipos de mensajes que se les presentó a los usuarios del prototipo.

```
//Mensaje
func mensaje(){
    let alert = UIAlertController(title: "Nivel Avanzado", message: "Felicidades, Te encuentras en el nivel de Realidad
    Aumentada", preferredStyle: .alert)
    alert.addAction(UIAlertAction(title: "OK", style: .default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}

//Alertas
func correcto(){
    let alert = UIAlertController(title:"Correcto", message: "Felicidades, Correcto. \n 100 puntos.", preferredStyle: .alert)
    alert.addAction(UIAlertAction(title:"OK", style: .default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}

func error(){
    let alert = UIAlertController(title:"Incorrecto", message: "Intentalo de nuevo, algo salio mal en la respuesta.",
    preferredStyle: .alert)
    alert.addAction(UIAlertAction(title:"OK", style: .default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
```

Figura 3.9: Código para representar alertas.

Como se mencionó en el párrafo anterior, cada función hay una variable de tipo `UIAlertController` a la cual se le asigna el título de la ventana y el mensaje que se requiere mostrar al usuario. Esto permitió un control de mensajes para cada tema que se utilizó para el prototipo.

3.6.4. Desarrollo de figuras 3D

El software SketchUP Pro 2018, como se mencionó en el marco tecnológico, tiene una variedad de herramientas para crear figuras o modelos en 3D, con este *software* se desarrollaron las figuras 3D mostradas en el prototipo móvil al momento de escanear los marcadores.

Los objetos desarrollados con el software sketchup 2018, se crean con la extensión SCN, este tipo de extensión es el único que permite ingresar Xcode para el manejo de objetos 3D.

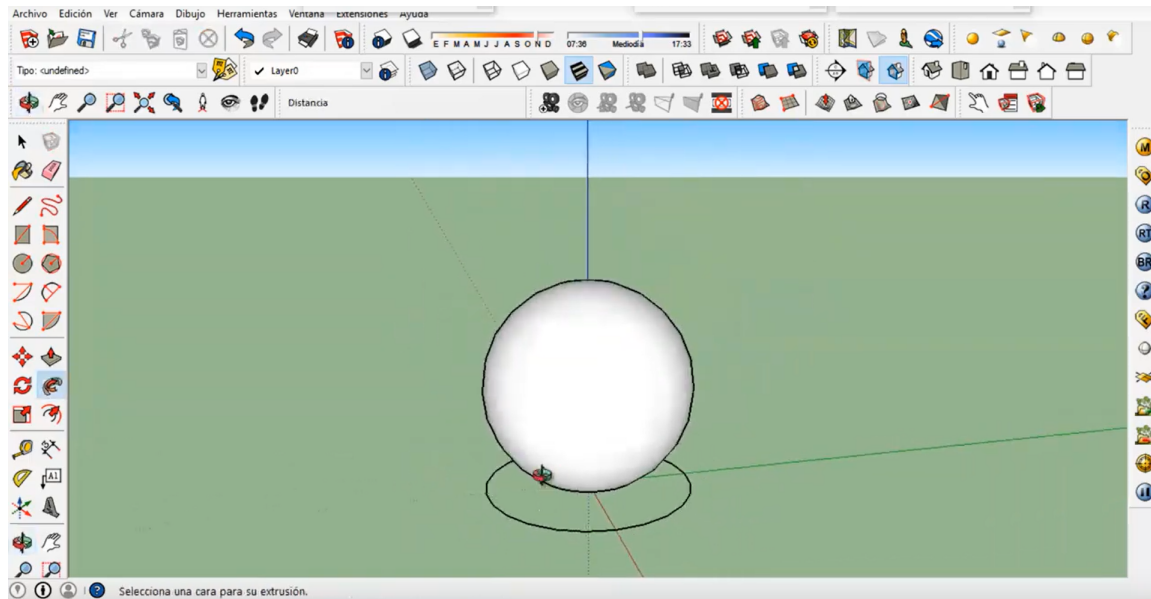


Figura 3.10: Creación de esfera en software SketchUP.



Figura 3.11: Laptop modelada en 3D.

3.6.5. Implementación de realidad aumentada

En la Figura 3.12 se muestra un aporte a la investigación, función elaborada a partir de varias librerías que aporta el lenguaje de programación Swift, el propósito de esta función fue analizar el marcador al cual se está enfocando mediante el dispositivo, el código analiza detalladamente patrones del marcador y las compara en una carpeta que se encuentra en la estructura del prototipo una vez analizado, si se encuentra el mismo patrón en uno de los objetos dentro de la carpeta, el código procede hacer la relación de marcado con objeto 3D y lo añade a la escena.

```
func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {
    let node = SCNNode()

    if let imageAnchor = anchor as? ARImageAnchor {
        let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
            imageAnchor.referenceImage.physicalSize.height)

        plane.firstMaterial?.diffuse.contents = UIColor(white: 1, alpha: 0.8)

        let planeNode = SCNNode(geometry: plane)
        planeNode.eulerAngles.x = -.pi / 2

        let shipScene = SCNScene(named: "art.scnassets/ship.scn")!
        let shipNode = shipScene.rootNode.childNodes.first!
        shipNode.position = SCNVector3Zero
        shipNode.position.z = 0.10

        planeNode.addChildNode(shipNode)

        node.addChildNode(planeNode)
    }

    return node
}
```

Figura 3.12: Función que analiza y muestra la figura acorde al marcador reconocido.

Como se mencionó, la función desarrollada es para analizar, mostrar y renderizar en tiempo real lo que el usuario está observando. Estas funciones predeterminadas por el IDE ayudan en la activación de la cámara y el preanálisis del marcador. En la unión de las funciones mencionadas en los párrafos anteriores da como resultado el objeto 3D en un entorno físico mediante un marcador como se muestra en la Figura 3.13.

Una vez que el usuario seleccionó el tema, se mostró un nuevo menú con subtemas del tema principal como se muestra en la Figura 3.15. En la implementación del prototipo de aplicación los usuarios interactuaron con el menú donde contienen los tres temas principales como se muestra en la Figura 3.14. Los tres temas conducen a la escena de realidad aumentada iniciando con un mensaje de bienvenida y describiendo el tema que seleccionó el usuario como se muestra en la Figura 3.16.

Cuando el usuario seleccionó “OK”, se realizó la actividad y terminó satisfactoriamente, enseguida, enfocó la cámara para iniciar con el análisis del marcador mediante la función de

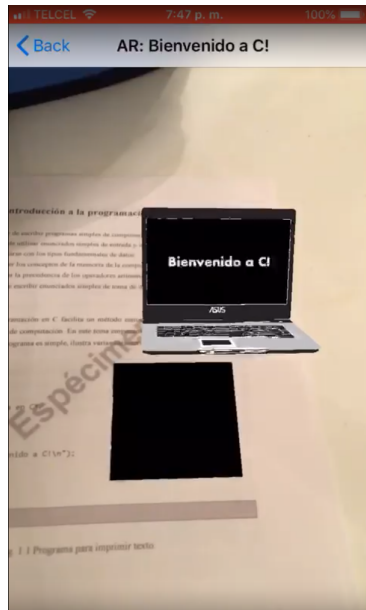


Figura 3.13: Objeto 3D aplicando la realidad aumentada.



Figura 3.14: Menú principal con los temas principales del prototipo de aplicación.

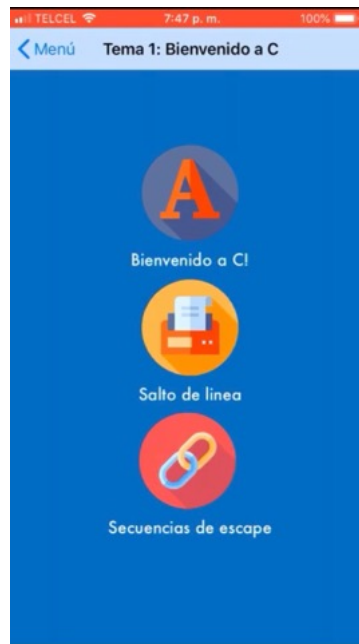


Figura 3.15: Menú secundario mostrando de los subtemas del prototipo de aplicación.

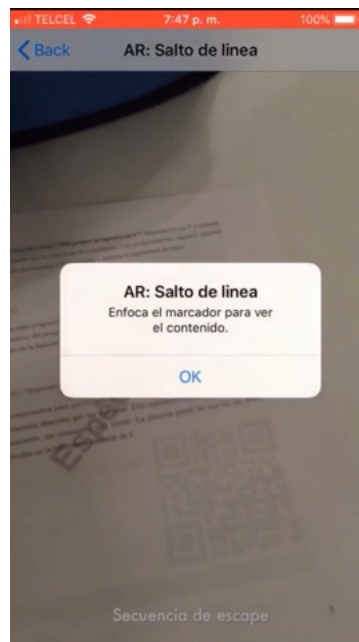


Figura 3.16: Mensaje de bienvenida y tema seleccionado.

renderizado en tiempo real. Esta función descrita anteriormente, analizó el marcador enfocado por el usuario, realizó una comparativa con los registros en la base de datos y lo relaciona con un objeto 3D alojado en la base de datos. En la Figura 3.13 se realizaron las primeras pruebas de detección de marcadores relacionados con objetos 3D, en las pruebas realizadas se observó que el alto consumo de recursos sobre el hardware no hacía factible el desempeño para futuras implementaciones. Corrigiendo el código inicial de la función desarrollada se llegó a una función que no consume tantos recursos de software y hardware. A continuación, dio como resultado un objeto 3D sobrepuesto en el marcador antes analizado como se muestra en la Figura 3.13. Este mismo procedimiento está programado en los demás temas propuestos para este prototipo.

En las Figuras 3.17 y 3.18 se observa los menús utilizados para los subtemas de los temas principales antes mencionados, los View Controllers están programados de la misma manera como se mencionó en la sección 3.6.2 main storyboard y view controllers que mediante botones se puede dirigir a la escena de realidad aumentada para el escaneo del marcado.



Figura 3.17: Menú de los subtemas seleccionados - if / else.

Las Figuras 3.19 y 3.20 son objetos 3D diseñados mediante el programa SketchUP diseños elaborados para los subtemas del prototipo. Los objetos 3D mencionados son de gran volumen y consumen alto recurso en cualquier dispositivo como se mencionó en la Sección 3.6.5 implementación de la realidad aumentada, pero con la función desarrollada se redujo el consumo de recursos un 34 % y ayudó a un mejor renderizado en tiempo real de los objetos 3D.



Figura 3.18: Menú de los subtemas seleccionados - Operaciones en C.

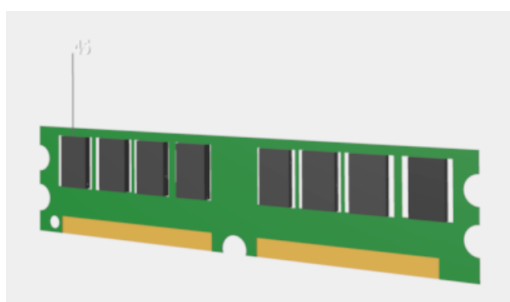


Figura 3.19: Objeto 3D diseñado para el subtema de operaciones matemáticas.



Figura 3.20: Objeto 3D diseñado para el subtema secuencia de escape.

3.6.6. Desarrollo físico del manual e interacción con el usuario

Los temas que se agregaron al prototipo son de un nivel básico, la razón es porque el proyecto está destinado a estudiantes de pregrado interesados a iniciar el conocimiento del lenguaje de programación en C sin conocimiento previo o básico de los temas en general. Se optó por el desarrollo de un material de apoyo para el aprendizaje de lenguaje en C, el cual tiene contenido introductorio a los temas mencionados, además, ayuda a complementar el prototipo con el despliegue de objetos 3D sobre el mismo material con la ayuda de marcadores, como se muestra en la Figura 3.21.

Tema 1: Introducción a la programación en C

Objetivos

- Ser capaz de escribir programas simples de computación en C.
- Se capaz de utilizar enunciados simples de entrada y salida.
- Familiarizarse con los tipos fundamentales de datos.
- Comprender los conceptos de la memoria de la computadora.
- Comprender la precedencia de los operadores aritméticos.
- Ser capaz de escribir enunciados simples de toma de decisiones.

Introducción

El lenguaje de programación en C facilita un método estructurado y disciplinado para el diseño de programas de computación. En este tema empezamos analizando un programa en C. Aun cuando este programa es simple, ilustra varias características importantes del lenguaje C.

Iniciemos

```
/*Mi primer programa en C*/  
  
main ()  
{  
    printf("Bienvenido a C!\n");  
}
```



Resultado

Bienvenido a C!

Figura 3.21: Ejemplo del manual.

Los temas que abarcaron el prototipo son:

- Tema 1: Introducción a la programación en C.
- Tema 2: Sumatoria de dos números enteros y conceptos de memoria.
- Tema 3: Operaciones de igualdad y relaciones.

Cada tema inicia con un enunciado de objetivos. Esto le indica al usuario que es lo que puede esperar, y le da una oportunidad, una vez que haya leído el capítulo, de determinar si él o ella han cumplido con estos objetivos. Genera confianza y es fuente de refuerzo positivo.

En las pruebas realizadas como parte del objetivo, se puso a prueba el prototipo de aplicación en un grupo de 60 personas, se dividió en 30 personas a quienes se les aplicó un ejercicio por tema sin el prototipo de aplicación solo basándose en los conocimientos adquiridos en las clases de programación. Las personas se les indicó los pasos a seguir con el prototipo para poder realizar las actividades de cada uno de los temas. El resultado fue variado ya que dependió de otros factores como lo son el rendimiento del prototipo, instalación, interacción, entre otros.

3.6.7. Diagrama de usabilidad

En esta sección está dedicada a la usabilidad del prototipo, aquí se verán cuales fueron los pasos que se llevaron a cabo para poder desplegar la realidad aumentada y la retroalimentación.

Como se muestra en la Figura 3.22 es el caso de uso del prototipo cuando el usuario selecciona el primer tema y el subtema 1. Lo que se logra es que fuera un flujo lineal al momento de mostrar la información. Este flujo se aplica para cada uno de los subtemas del tema 1 como se muestra en la Figura 3.23 y la Figura 3.24. El diseño y el desarrollo de el prototipo fue realizada para que el usuario tuviera una retroalimentación de una manera que fuera comprensible para el usuario y supiera la manera correcta de realizar las actividades que el prototipo le fue marcado.

Cada uno de los temas contiene una sección de retroalimentación si el usuario comete algún error al momento de ingresar la información y verificarla. Fue programado para que el usuario pudiera tener un dato adicional y poder completar la actividad satisfactoriamente. También, contiene una sección de aprobación cuando el usuario sin retroalimentación completo la actividad. Esto se puede verificar en las Figuras 3.22 en el apartado 5.1 y 5.2. Además, a diferencia del tema 1 entre los temas 2 y 3 no contienen subtemas el flujo lineal de la información es mas directo (corto) a la actividad que deben de desarrollar como se muestra en la Figura 3.25

En conclusión, el desarrollo y la implementación no fue fácil a un inicio ya que la función de renderizado ocupaba muchos recursos de hardware y software, pero en el transcurso del desarrollo se llegó a una versión estable, capaz de administrar adecuadamente los recursos del dispositivo donde se encuentra instalado. Además, incluyendo las secciones de retroalimentación de cada una de las actividades que se emplearon en el prototipo, como también,

agregando la sección de evaluación donde el usuario ya no tendría que ir a otro equipo para poder realizar esta actividad esto hizo mas eficiente para el prototipo en la retroalimentación desde el uso. Parte de los problemas encontrados al inicio de las pruebas fueron las detecciones de los marcadores ya que no se encontraban en zonas con luz, fue corrigió con marcadores más grandes y con un fondo más amplio, este problema se puntualizó en el Capítulo 1, Sección 1.5.1 Limitaciones donde se menciona que el reconocimiento de los marcadores dependerá del enfoque, distancia e iluminación.

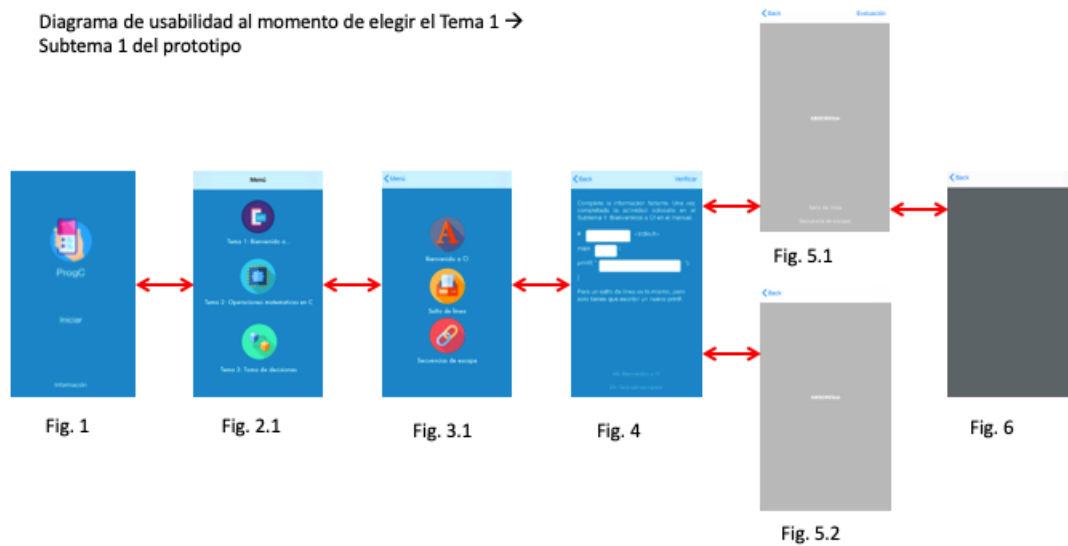


Figura 3.22: Caso de uso del Tema 1 ->Subtema 1

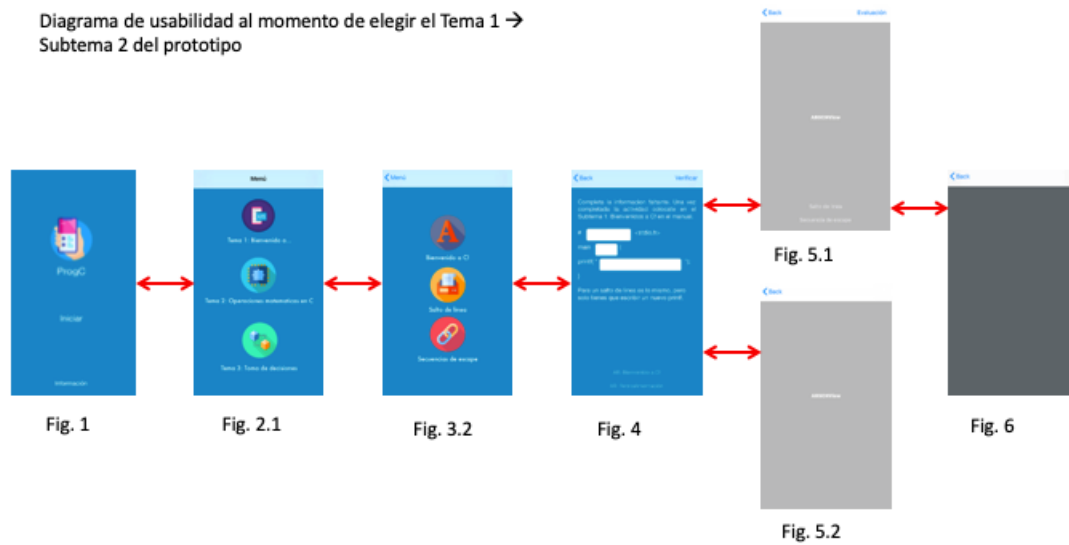


Figura 3.23: Caso de uso del Tema 1 ->Subtema 2

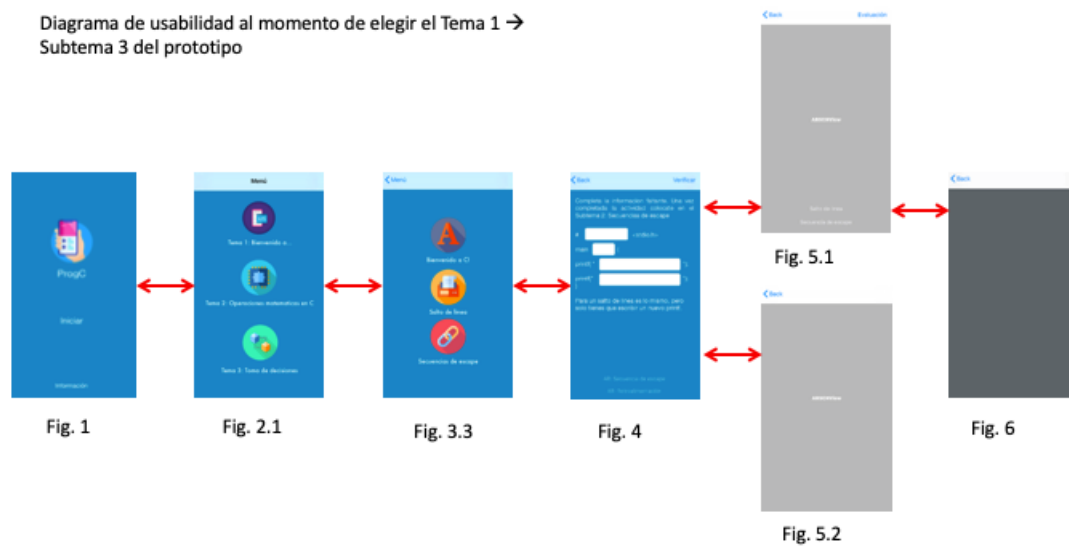


Figura 3.24: Caso de uso del Tema 1 ->Subtema 3

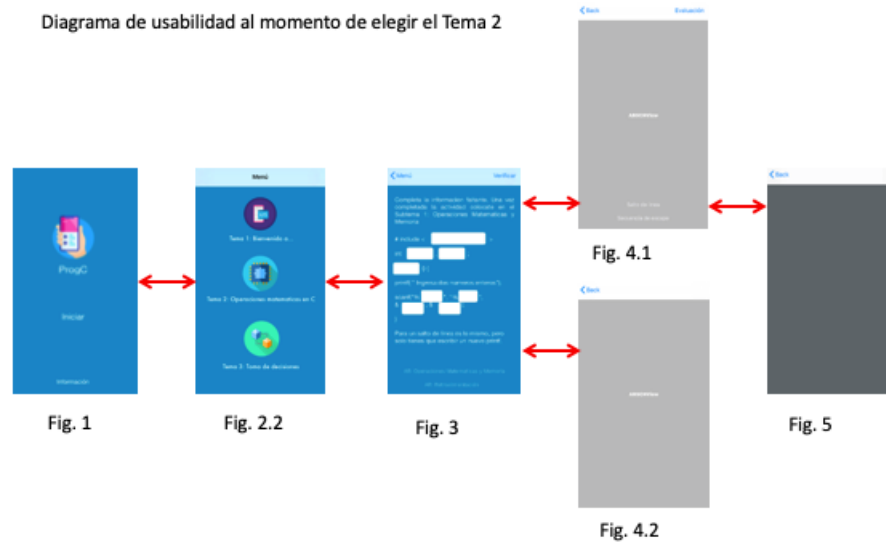


Figura 3.25: Caso de uso del Tema 2

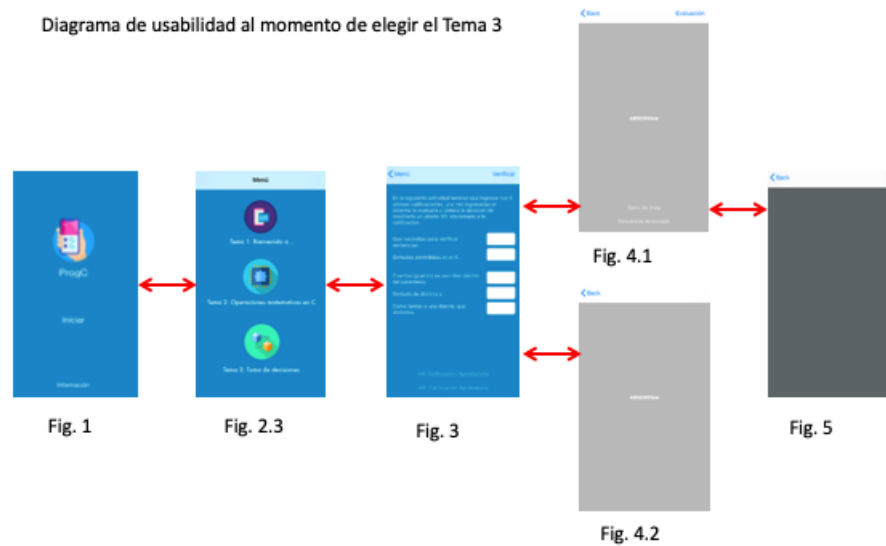


Figura 3.26: Caso de uso del Tema 3

Capítulo 4

Resultados y discusiones

En el capítulo siguiente se muestran los resultados obtenidos en la elaboración del prototipo móvil. En la primera sección se muestra el prototipo desarrollado, en la segunda sección se presentan las gráficas que se obtuvieron en las pruebas realizadas con los usuarios seleccionados para la prueba. Por último, la interpretación y discusión de los datos obtenidos.

4.1. Prototipo desarrollado

El prototipo de aplicación móvil basado en realidad aumentada para el aprendizaje del lenguaje C, se puede instalar solamente en versiones de iOS 11 o superiores. ¿Por qué se eligió esta plataforma?, como se mencionó en el Capítulo 3, el software y el hardware permitió el desarrollo del prototipo de aplicación. Para el análisis de la información se utilizó la herramienta Microsoft Power BI.

La primera visualización, al abrir el prototipo se puede encontrar el menú principal como se muestra en el Capítulo 3, Sección 3.5, de la Figura 3.2, aquí el usuario tendrá que seleccionar uno de los tres temas seleccionados para el prototipo. Una vez seleccionado, mostrará la primera escena donde tendrá que enfocar el marcador que se encontrará en el material de apoyo elaborado para este proyecto. Una vez enfocado el marcador se mostrará el objeto 3D como se muestra en el Capítulo 3 Sección 3.6 Figura 3.8.

El material de apoyo es un complemento con el prototipo de aplicación elaborado, en este se encuentran ejercicios que el usuario puede elaborar con una herramienta de software libre o de licencia, en este material de apoyo se encuentran los marcadores con el cual podrán interactuar con el prototipo.

4.2. Resultados obtenidos, interpretación y discusión

En la sección de resultados, se realizó una encuesta de 11 preguntas y una sección de comentarios. La encuesta fue dividida de la siguiente manera, datos generales, preguntas de

diagnóstico y por último manejo de la herramienta. En total fueron 34 encuestas realizadas a alumnos del IIT que contaran en su mapa curricular una materia destinada a la programación. A continuación, la sección de datos generales se les preguntó lo siguiente, género, semestre que está cursando y la carrera que está cursando. En género se quiso tener el control de cuántos hombres y mujeres se encuestaron la información de la Figura 4.1. Como se ve en la Figura 4.1 el 55.9 % representa a 19 mujeres y el 44.1 % representa a 15 hombres realizaron alguna actividad del prototipo de aplicación.

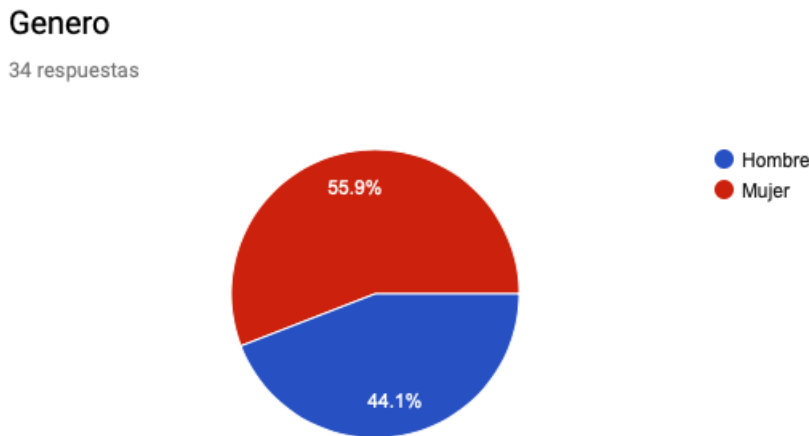


Figura 4.1: Género.

La siguiente pregunta, ¿Qué semestre estás cursando? se contestó lo siguiente: 14 alumnos contestaron que se encuentran entre los semestres 1 – 3 eso representó el 41.2 %, 8 alumnos contestaron que están cursando los semestres 4 – 6 esto representa el 23.5 %, 5 alumnos contestaron que se encuentran entre los semestres 7 – 9 lo que representa el 14.7 % y 7 alumnos contestaron que están en un semestre mayor al noveno, esto representa el 20.6 % como se muestra en la Figura 4.2.

La pregunta de la carrera que se está cursando contestaron lo siguiente: 20 alumnos de la carrera de Ingeniería en Sistemas Computacionales esto representó el 58.8 %, 5 alumnos de la carrera de Ingeniería en Sistemas Digitales que representó el 14.7 %, 3 alumnos de la carrera de Ingeniería en Biomédica el cual representó el 8.8 %, 2 alumnos de la carrera de Ingeniería en Manufactura que representó el 5.9 % y por último 3 alumnos de diferentes carreras (1 de Ingeniería Industrial y de sistemas, 1 alumno de Ingeniería en Aeronáutica y 1 alumno de Ingeniería Eléctrica) que representaron el 8.7 %, en las encuestas dio a entender que los alumnos de la carrera de Ingeniería en Sistemas Computacionales predominan en la programación como se puede observar en la Figura 4.3.

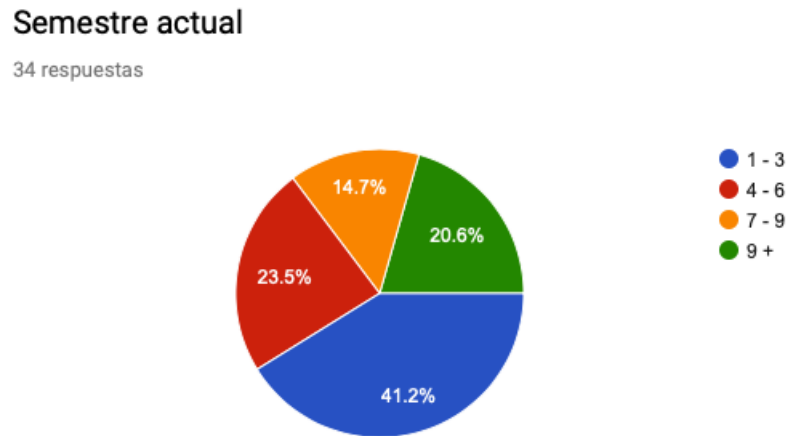


Figura 4.2: Semestres que actualmente los 34 alumnos encuestados cursan.

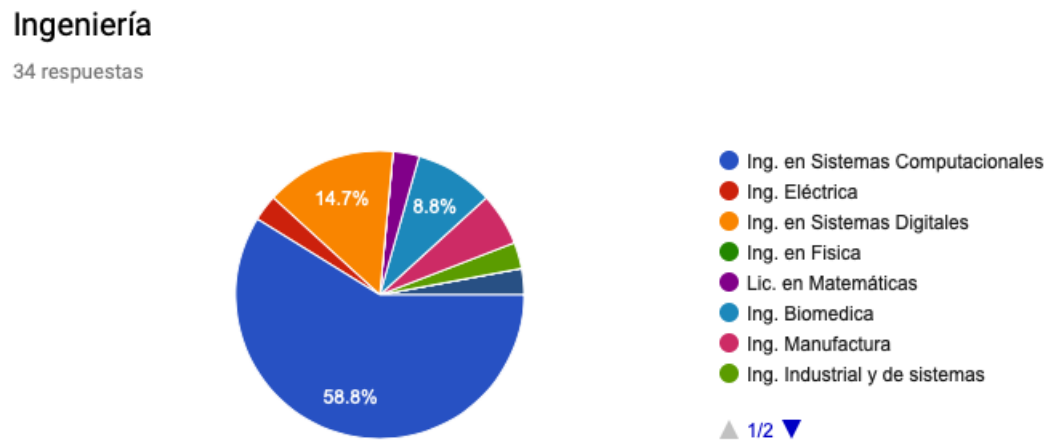


Figura 4.3: Ingenierías encuestadas.

La siguiente sección de las preguntas pertenecen al apartado de diagnóstico y se preguntaron cosas esenciales de conocimiento sobre las herramientas tecnológicas y de uso.

La pregunta que menciona si utilizarías el prototipo como complemento al material de clases se contestó lo siguiente: de las 3 opciones que disponían (Sí, No o Tal vez), los 34 alumnos contestaron Sí al uso del prototipo como complemento al material de las clases representando el 100% como se muestra en la Figura 4.4. Este resultado quiere decir que los alumnos están dispuestos a usar herramientas externas que los ayuden a complementar el aprendizaje visto en la clase, ya que como se mencionó no basta con lo que se enseña en clase para aprender y entender.

Utilizarías el prototipo como un complemento al material de las clases?

34 respuestas

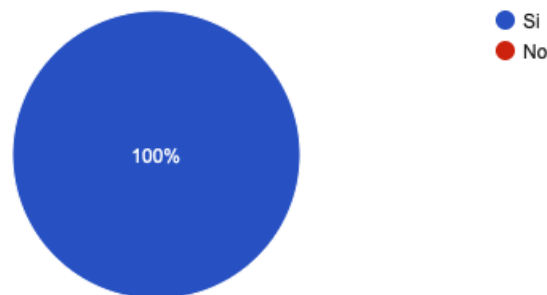


Figura 4.4: Uso de tecnología como complemento en los cursos.

La siguiente pregunta de la encuesta: consideras que la tecnología de RA sea suficiente para complementar lo visto en clase, los alumnos contestaron lo siguiente: 32 alumnos que representaron 94.1% contestaron que Sí contemplarían lo visto en clase y 2 alumnos que representó el 5.9% contestaron que No contemplarían lo visto en clase como se muestra en la Figura 4.5.

En la siguiente pregunta se le mencionó si considerarías que la institución contara con nuevas tecnologías para uso educativo. 33 alumnos contestaron Sí que representa el 97.1% y 1 alumno contestó No que representa el 2.9%, como se puede observar los alumnos desean que se usen herramientas tecnológicas para complementar su aprendizaje como se muestra en la Figura 4.6 La siguiente sección trata sobre el uso del prototipo, si fue fácil de usar, si se comprendió el tema, si la retroalimentación fue suficiente y si consideras cambios en el prototipo, el método de calificación en cada una de las preguntas fue la siguiente 1 considerando la más baja y 5 como la más alta.

La pregunta, el prototipo fue fácil de usar los alumnos contestaron lo siguiente, 20 alumnos

Consideras que dicha tecnología sea suficiente para complementar lo visto en los cursos?

34 respuestas

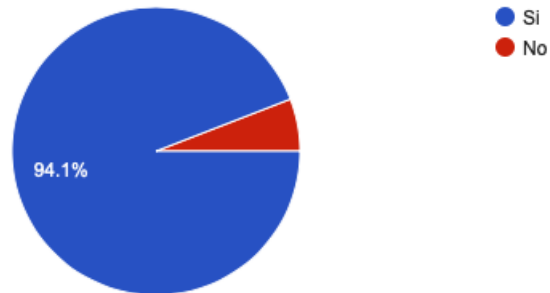


Figura 4.5: Considerar si la tecnología de RA es suficiente para complementar los cursos impartidos de programación.

Te gustaría que la institución contara con nuevas tecnologías para el uso educativo?

34 respuestas

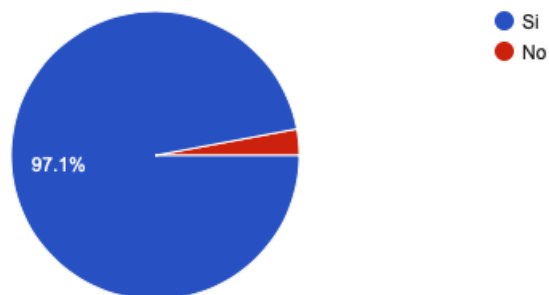


Figura 4.6: Nuevas tecnologías.

contestaron 5 como la calificación más alta esto representó el 58.8 %, 8 alumnos contestaron 4 como la segunda calificación más alta y esto representó el 23.5 %, 4 alumnos contestaron 3 como calificación intermedia y esto representó el 11.8 % y por último 2 alumnos contestaron 2 y 1 como calificaciones más bajas y representó un 4.8 % como se muestra en la Figura 4.7. En esta pregunta da a entender que el prototipo en diseño fue apto para su fácil uso.

El prototipo es facil de usar (Calificar en una escala del 1 al 5)

34 respuestas

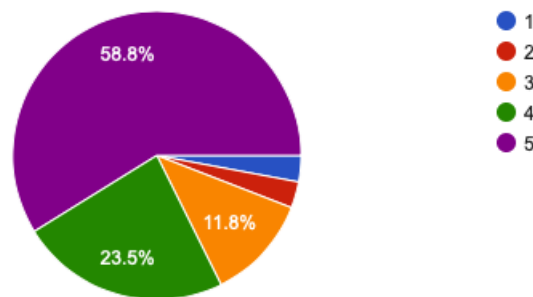


Figura 4.7: El prototipo fue facil de utilizar.

La pregunta sobre la comprensión del tema del prototipo se utilizó la misma dinámica de calificación del 1 al 5 donde 16 alumnos calificaron al prototipo con un 5 que representó el 47.1 %, 10 alumnos calificaron con 4 que representó el 20.4 %, 7 alumnos contestaron con una calificación de 3 que representó un 20.6 % y por último 1 alumno contestó con una calificación 1 la más baja y esto representa el 2.9 % como se muestra en la Figura 4.8. Los resultados de esta pregunta da a entender que la mayoría de los alumnos que interactuaron en el prototipo comprendieron el tema que seleccionaron y pudieron hacer uso de él.

La siguiente pregunta, la retroalimentación es suficiente para poder realizar la actividad donde 19 alumnos calificaron con un 5 la retroalimentación esto representó el 55.9 %, 9 alumnos contestaron con un 4 de calificación que representó el 26.5 % y 6 alumnos contestaron con un 3 de calificación que representó el 17.6 % como se muestra en la Figura 4.9.

En esta pregunta se puede interpretar que la mayoría de los alumnos encuestados entendieron la retroalimentación y pudieron completar satisfactoriamente la actividad en el prototipo.

La pregunta de cambios en el prototipo los alumnos contestaron lo siguiente: 15 alumnos contestaron Sí que representó el 44.1 % en realizar algún cambio en la aplicación, 11 alumnos contestaron No que representó el 32.4 % y 8 alumnos contestaron Tal vez en realizar un cambio en la aplicación como se muestra en la Figura 4.10.

En prototipo, la comprensión de los tema fue suficiente (Calificar en una escala del 1 al 5)

34 respuestas

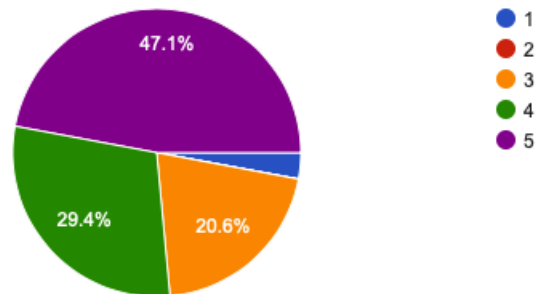


Figura 4.8: Comprensión del tema.

La retroalimentación es suficiente para poder realizar la actividad correctamente (Calificar en una escala del 1 al 5)

34 respuestas

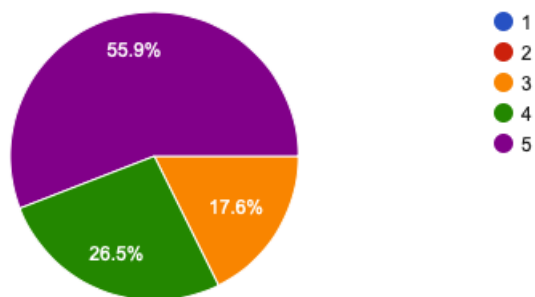


Figura 4.9: La retroalimentación ayudó a terminar las actividades.

Esta pregunta da a entender que los alumnos que utilizaron el prototipo desearían que se cambiaran algunos de los objetos relacionados en la retroalimentación ya que solo muestra texto y puede que otros alumnos pierdan el interés. Además, faltan temas de programación que los alumnos desearían que estuvieran pero no fueron agregados por el hecho del tiempo de desarrollo del prototipo los cuales se mencionaron en los alcances y limitaciones.

Crees que se deben de realizar cambios en el prototipo presentado?

34 respuestas

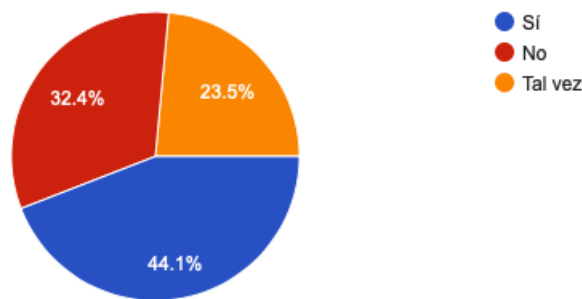


Figura 4.10: Cambios en el prototipo.

La última pregunta, al alumno se le preguntó que si el prototipo de aplicación sería un excelente complemento para cualquier curso de programación y estas fueron sus respuestas. 23 alumnos contestaron que Sí esto representó el 67.6 %, 5 alumnos contestaron No esto representó el 14.7 % y 6 alumnos contestaron con Tal vez que representó el 17.6 % como se muestra en la Figura 4.11. Esta pregunta ayuda a interpretar que el 67.6 % de los alumnos considera que debería existir nuevas herramientas de uso en las aulas para mejorar el aprendizaje en clase y que una de estas herramientas sería la realidad aumentada.

Para una mejor visualización de la información en la sección de anexos estará la tabla con concentración de información.

Después de haber utilizado el prototipo, te gustaría usarla como complemento en los cursos?

34 respuestas

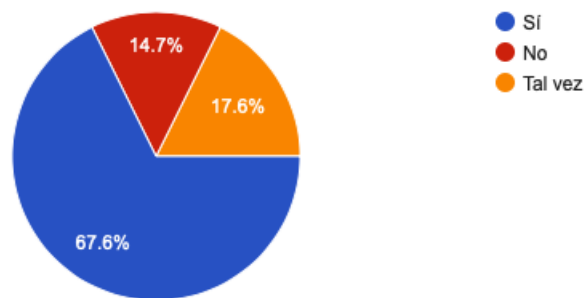


Figura 4.11: Utilizarían como complemento para el curso de programación.

Capítulo 5

Conclusiones

Se puede concluir que mediante el prototipo, se proporciona un aprendizaje en el aula y fuera de ella. Parte de estos aprendizajes se basan en que el usuario descubra nuevos métodos y que no se conforme con lo visto en un salón de clases, así como se mencionó en la introducción. Los prototipos y aplicaciones de RA en dispositivos móviles con ayuda de software que se encuentra en constante cambio, apoyaron la construcción del prototipo que refuerza al aprendizaje con la interacción en el mundo físico.

Después del proceso de programación y requerimientos de hardware y software se cuenta con un prototipo móvil basado en tecnología de RA que puede ser distribuido para una instalación en dispositivos de gama media y gama alta. En el desarrollo del prototipo fue necesario hacer una investigación más allá de las áreas del conocimiento de sistemas computacionales como ecuación diferencial, diseño 3D y aplicación del diseño.

Cabe resaltar, durante este paso inicialmente el prototipo interactuaría más con el usuario dejando de lado los marcadores pero por efectos de ayuda visual se decidió utilizar los marcadores y con las que se realizaron las pruebas que el Capítulo 4. Estos cambios se realizaron durante este proceso y esto se hizo con el propósito de cumplir con los objetivos generales.

5.1. Objetivos cumplidos

En un comparativo sobre los resultados que se obtuvieron en la aplicación del test a los usuarios, se observó que alumnos que contaban con el conocimiento de programación (UCC) y usuarios que no cuentan (USC) con dicho conocimiento se pudo observar que los USC al aplicar el test de conocimiento adquirido (TCA) están a la par de los UCC que se les aplicó el TCA con una discrepancia del 3 %. Al concluir con el test los usuarios manifestaron un interés sobre el prototipo y más por el hecho que está desarrollado mediante la tecnología de RA.

Por parte del prototipo se concluyó que es de ayuda para las nuevas Tecnologías de Información y Comunicación (TIC) como lo es la RA, ya que con la estadística obtenida del capítulo 4 y cumpliendo con el objetivo general que es "Desarrollar un prototipo de

aplicación para dispositivos móviles en iOS utilizando la realidad aumentada para apoyar el aprendizaje del lenguaje de programación en C para estudiantes de pregrado de las carreras de ingeniería", ya que el prototipo se desarrolló con éxito ofreciendo un apoyo para adquirir conocimiento de la programación en C.

Además, se cumplió con el desarrollo del manual de usuario con 3 temas que interactúan a través del prototipo móvil como lo son la introducción a la programación en C, sumatoria de dos números enteros, conceptos de memoria y operaciones de igualdad y relaciones, cada uno de estos temas contienen actividades para que el usuario interactúe con el manual y el prototipo. El propósito del manual es guiar al alumno para llevar un orden con los temas principales de programación y ayudar al prototipo móvil a desplegar la realidad aumentada en el lugar requerido.

5.2. Trabajos a futuro

Es necesario puntualizar que el proceso que se llevó a cabo requiere de varias áreas de conocimiento para poder presentar un producto o servicio que pueda beneficiar a los usuarios a los que se está destinando.

El prototipo tiene mejoras que se pueden realizar en un futuro, se pueden agregar más temas a los cuales al alumno se le dificulte el aprendizaje del lenguaje C, también se puede ofrecer mejores diseños en las animaciones y agregar más contenido multimedia que por cuestiones de tiempo no se agregaron. Además, como ya se mencionó, Xcode permite la realidad aumentada sin marcadores que en este prototipo no se agregó ya que para el usuario era tedioso el girar su dispositivo para encontrar las animaciones, en un proyecto futuro se puede mejorar con estas funciones ya que no se necesitaría un manual para la función del prototipo.

Finalmente, el aprendizaje de un alumno no termina al finalizar la carrera, si no que continúa en la vida diaria. La continua mejora de herramientas como es la RA da el impulso a la generación de nuevas herramientas para la ayuda del usuario en una tarea específica como se desarrolló en este prototipo.

Bibliografía

- [1] A. N. Mayer, O. G. Roldán, and S. J. Guerrero, *Modelo Educativo Para La Educación Obligatoria*, 1st ed., 2017, vol. 1.
- [2] J. Cabero Almenara, *Nuevas tecnologías aplicadas a la educación*. McGraw-Hill, 2007, no. C10-40.
- [3] H. M. Deitel, P. J. Deitel, J. Listfield, T. Nieto, C. Yaeger, and M. Zlatkina, *C: como programar*. Pearson Education, 2003.
- [4] C. P. Espinosa, “Realidad aumentada y educación: análisis de experiencias prácticas,” *Pixel-Bit. Revista de Medios y Educación*, no. 46, pp. 187–203, 2015.
- [5] F. Quint, S. Katharina, and G. Dominic, “A mixed-reality learning environment,” *Procedia Computer Science*, vol. 1, p. 48, 2015.
- [6] G. Kiss and Z. Arki, “The influence of game-based programming education on the algorithmic thinking,” *Procedia-Social and Behavioral Sciences*, vol. 237, pp. 613–617, 2017.
- [7] M. Kesim and Y. Ozarslan, “Augmented reality in education: current technologies and the potential for education,” *Procedia-Social and Behavioral Sciences*, vol. 47, pp. 297–302, 2012.
- [8] D. Radosevic and T. Orehovacki, “Verifier: Educational tool for learning programming, institute of mathematics and informatics,” *Vilnius*, vol. 1, pp. 216–280, 2012.
- [9] W. Y. Seng and M. H. Yatim, “Computer game as learning and teaching tool for object oriented programming in higher education institution,” *Procedia-Social and Behavioral Sciences*, vol. 123, pp. 215–224, 2014.
- [10] X. Basogain, M. Olabe, K. Espinosa, C. Rouèche, and J. Olabe, “Realidad aumentada en la educación: una tecnología emergente,” *Recuperado de www.anobium.es/docs/gc_fichas/doc/6CFJNSalrt.pdf*, 2007.
- [11] F. Telefónica, *Realidad Aumentada: una nueva lente para ver el mundo*. Fundación Telefónica, 2011.

- [12] J. Fombona Cadavieco, M. Á. Pascual Sevillano, and M. F. M. Ferreira Amador, “Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles,” *Pixel-Bit. Revista de medios y educación*, no. 41, 2012.
- [13] C. González, D. Vallejo, J. Albusac, and J. Castro, “Realidad aumentada. un enfoque práctico con artoolkit y blender,” [urlhttp://www.librorealidadaugmentada.com](http://www.librorealidadaugmentada.com), 2013.
- [14] A. B. Craig, *Understanding augmented reality: Concepts and applications*. Newnes, 2013.
- [15] M. R. Zarate Nava, C. F. Mendoza González, H. Aguilar Galicia, and J. M. Padilla Flores, “Marcadores para la realidad aumentada para fines educativos,” *ReCIBE. Revista electrónica de Computación, Informática Biomédica y Electrónica*, no. 3, 2013.
- [16] P. d. R. A. S. Marcadores, “Entornos móviles para el desarrollo de asistentes personales (rasmap). ministerio de educación y ciencia. programas nacionales del plan nacional de investigación científica, desarrollo e innovación tecnológica 2004-2007,” TIN2006-15418-C03, Tech. Rep.
- [17] *Realidad Aumentada: una nueva lente para ver el mundo*. Editorial Ariel, 2003.
- [18] M. Mathias and J. Gallagher, *Swift Programming: The Big Nerd Ranch Guide*. Pearson Technology Group, 2015.
- [19] H. Sean and S. Scott. (2017) Los primeros apps de arkit revelan cómo funciona la realidad aumentada de apple. <https://www.cnet.com/es/noticias/los-primeros-apps-de-arkit-revelan-como-funciona-la-realidad-aumentada-de-apple/>.
- [20] Android. (2017) Android developers official website. <https://developer.android.com>.
- [21] Technologies, “Unity3d,” <https://unity3d.com/es/>, 2017.
- [22] Apple, “class arsession,” <https://developer.apple.com/documentation/arkit/arsession>, 2019.
- [23] —, “class arconfiguration,” <https://developer.apple.com/documentation/arkit/arconfiguration>, 2019.
- [24] —, “class aranchor,” <https://developer.apple.com/documentation/arkit/aranchor>, 2019.
- [25] —, “class arframe,” <https://developer.apple.com/documentation/arkit/arframe>, 2019.
- [26] —, “class arcamera,” <https://developer.apple.com/documentation/arkit/arcamera>, 2019.

- [27] —, “class arscnview,” <https://developer.apple.com/documentation/arkit/arscnview>, 2019.
- [28] T. Navigation, “Sketchup,” <https://sketchup.com/learn>, 2017.
- [29] Ruby, “Ruby el mejor amigo de un desarrollador,” <https://ruby-lang.org/es/about>, 2017.
- [30] Apple, “Swift,” <https://swift.org/>, 2014.
- [31] —, “Objective-c,” <https://web.archive.org/web/20041011143829/http://developer.apple.com/docum> 2006.
- [32] I. C. S. S. E. S. Committee and I.-S. S. Board, “Ieee recommended practice for software requirements specifications.” Institute of Electrical and Electronics Engineers, 1998.

Apéndice A

Manual de usuario

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



**PROTOTIPO MOVIL BASADO EN REALIDAD AUMENTADA
PARA EL APOYO DEL APRENDIZAJE DEL LENGUAJE DE
PROGRAMACIÓN EN C (Material de apoyo)**

Material de apoyo presentado por:

Jaime Guerra Vázquez
Rodrigo Emmanuel Sánchez Benítez

Requisito para la obtención del título de
INGENIERO EN SISTEMAS COMPUTACIONALES

Asesor

Dra. Vianey Guadalupe Cruz Sánchez

Ciudad Juárez, Chihuahua

Noviembre del 2019

Introducción

Este material de apoyo está ligado a lo que se visualiza en el prototipo de aplicación basado en realidad aumentada para el aprendizaje del lenguaje de programación en C.

Instrucciones de uso

Para el uso combinado del material de apoyo y la aplicación, sigue estos sencillos pasos, en cada tema iniciará una breve introducción, contará con un marcador como se muestra en la figura 1.1. Este marcador lo utilizaremos en el prototipo de aplicación basado en realidad aumentada para el aprendizaje en el lenguaje en C (ProgC).



Fig. 1.1 Marcador para el uso en ProgC

Utilizaras los marcadores que aparecen en cada tema para interactuar con el material de apoyo.

Tema 1: Introducción a la programación en C

Objetivos

- Elaboración de programas simples.
- Capacidad para utilizar enunciados de entradas y salidas en programación

Introducción

El lenguaje de programación en C da la oportunidad de tener un sistema ordenado para el diseño de programas. Parte de este tema iniciaremos con el análisis de un programa realizado en lenguaje C.

Comencemos

```
/*Mi primera práctica en C*/  
  
main ()  
{  
    printf ("¡Bienvenido a C! \n");  
}
```

Resultado

Bienvenido a C!



Fig. 1.1 Programa para imprimir texto.

NOTA: Toma tu tiempo ingresa al prototipo ProgC, selecciona el primer tema y seguido seleccione el primer subtema “Bienvenidos a C”.

Si en la aplicación una vez realizada la actividad muestra un botón en la parte inferior con la leyenda “AR: Retroalimentación” tendrás que enfocar el siguiente marcador.



Analicemos lo visto anteriormente. En la siguiente línea de código **/*Mi primera practica en C*/** empieza con un **/*** y termina con un ***/**, esto nos indica que parte de esta línea de código es un comentario. Los comentarios los podemos utilizar para describir que es lo que ocurre en la sección del programa y como referencia para otros usuarios que hacen uso del programa.

En la línea de código **main ()** forma parte de todo programa en C. Los paréntesis que se encuentran al lado derecho del **main** nos indican que es una función general.

Parte de la línea del código **printf ("Bienvenido a C!" \n);** , le indica a la computadora que realice una acción, lo que realizara la computadora es imprimir en pantalla cierta cadena de caracteres que se encuentran dentro de las comillas. También, la función **printf** es parte de las tantas funciones que se encuentra en la biblioteca estándar del lenguaje C.

Secuencia de escape	Descripción
\n	Nueva línea. Coloca el cursor al principio de la siguiente línea.
\t	Tabulador horizontal. Mueve el cursor al siguiente tabulador.
\r	Retorno de carro. Coloca el cursor al principio de la línea actual; no avanza a la línea siguiente.
\a	Alerta. Hace sonar una campana del sistema.
\\	Diagonal invertida. Imprime un carácter de diagonal invertida en un enunciado printf.
*	Doble comilla. Imprime un carácter de doble comilla en un enunciado printf.

Fig. 1.2 Secuencias de escape comunes en C.

¡La función `printf` puede presentar de varias maneras en pantalla Bienvenido a C! Por ejemplo, el programa de la figura 1.3 obtiene el mismo resultado que en el programa realizado anteriormente mostrado en la figura 1.1.

La función **`printf`** puede tener como resultado varias líneas código como se ve en la figura 1.4. Cada vez que se encuentra con la secuencia de escape `\n`, **`printf`** se coloca al principio de la línea siguiente.

```
main ()
{
    printf ("Bienvenido");
    printf ("\n a C!");
}
```

Resultado

Bienvenido a C!

1.3 Resultado en una sola línea utilizando enunciados separados de **`printf`**.

NOTA: Toma tu tiempo ingresa al prototipo ProgC, selecciona el primer tema y seguido selecciona el segundo subtema “Salto de línea”.

Si en la aplicación una vez realizada la actividad muestra un botón en la parte inferior con la leyenda “AR: Retroalimentación” tendrás que enfocar el siguiente marcador.



En la siguiente actividad se mostrará las secuencias de escape y como las podemos utilizar en una línea de código.

```
main ()  
{  
    printf("Bienvenido\n\nC!\n");  
}
```



Resultado

Bienvenido
a
C!

Fig. 1.4 Líneas múltiples

NOTA: Toma tu tiempo ingresa al prototipo ProgC, selecciona el primer tema y seguido seleccione el tercer subtema “Secuencias de escape”.

Si en la aplicación una vez realizada la actividad muestra un botón en la parte inferior con la leyenda “AR: Retroalimentación” tendrás que enfocar el siguiente marcador.



Tema 2: Sumatoria de dos números enteros y conceptos de memoria.

Objetivos

- Comprender los conceptos de la memoria de la computadora.
- Comprender la precedencia de los operadores aritméticos.

Iniciemos

En el siguiente programa, se utilizará una función que se encuentra en la biblioteca estándar del lenguaje C **scanf**. Esta función obtiene los datos ingresados desde el teclado por el usuario y los almacena en la memoria para ser utilizados en el tiempo de ejecución del programa.

#include <stdio.h>

es una librería que es utilizada en el preprocesador de C. Las líneas que se inician con el signo de # son procesadas por el preprocesador, antes de la compilación del programa. El archivo de cabecera (stdio.h) es un archivo que contiene todas las referencias, información y declaraciones para que el compilador reconozca todas las funciones estándar de la biblioteca en C

```
#include<stdio.h>
main ()
{
    int interger1, interger2, sum; /*Line del código donde se declaran las variables globales*/

    printf ("Ingresa el primer número entero"); /*Mensaje que se imprime en la pantalla del usuario*/
    scanf ("%d", &interger1); /*Scanf interactúa lee la información que se almacenara*/
    printf ("Ingresa el primer número entero"); /
    scanf ("%d", &interger2);

    sum = interger1 + interger2;
```

```
printf ("La suma es:", sum);  
  
return 0; /*Indica que el programa termino satisfactoriamente*/  
}
```

NOTA: Toma tu tiempo ingresa al prototipo ProgC, selecciona el segundo tema y seguido seleccione el primer subtema “Operaciones aritméticas”.



Resultado

Ingresa el primer numero
45
Ingresa el segundo numero
72
La suma es: 117

Fig. 2.1 Resultado de la suma dos números enteros.

Si en la aplicación una vez realizada la actividad muestra un botón en la parte inferior con la leyenda “AR: Retroalimentación” tendrás que enfocar el siguiente marcador.



Las variables **ingerger1**, **ingerger2**, y **sum** se les puede asignar cualquier tipo de datos, como lo puede ser Flotante (*float*), Entero (*Interger*), *Char* (carácter), etc. Estas variables son una posición de memoria donde se puede almacenar un valor para uso de un programa en tiempo de ejecución. Esta declaración especifica que las variables **interger1**, **interger2**, y **sum** son tipo **int** lo que significa que estas variables contendrán valores enteros, es decir, tipos de datos numéricos que no contienen números con punto decimal.

En la línea de código `printf ("Ingresa el primer número");` imprime la literal **Ingresa el primer número** en la pantalla y **se colocó** al principio de la línea siguiente.

La función `scanf` interactúa con lo que el usuario ingresa directamente desde el teclado. La función `scanf` contiene dos argumentos muy importantes, `"%d"` y `&interger1`. Parte del primer argumento, es una cadena que da el control necesario del formato del valor, en otras palabras, el tipo de datos a ingresar. El ampersand agregado a la variable parte del estándar le indica al `scanf` que posición de la memoria se estará almacenando dicha información ingresada desde el teclado.

En la línea de código de **asignación** `sum = interger1 + interger2;` calcula la suma de las variables `interger1` e `interger2` y asigna el resultado a la variable `sum` utilizando el operador de asignación `=`.

Parte de la línea de código printf ("La suma es: %d\n", sum); utilizando la función printf para mostrar en pantalla lo siguiente → La suma es: seguida del **valor numérico** de la variable **sum**.

Integer1	45
Integer2	72
sum	117

Fig. 2.3 Localizaciones de memoria.

Especimen

Tema 3: Operaciones de igualdad y relaciones

Objetivos

- El usuario podrá ser capaz de escribir líneas de código simples de toma de decisiones.

Iniciemos

Para determinar si la calificación de una persona en un examen mayor que o igual a 60, y si es así, imprimir en pantalla el mensaje “Felicidades! Usted paso”. Las condiciones en las estructuradas de if se forman utilizando los operadores de igualdad y los operadores relacionales se encuentran resumidos en la figura 3.1. Los operadores de igualdad tienen un nivel de precedencia menor que los operadores relacionales estos se asocian en una relación de izquierda a derecha.

Operador de igualdad estándar algebraico u operador racional	Operador de igualdad o relacional de C	Ejemplo de condición de C	Significado de la condición en C
Operadores de igualdad			
=	==	X == Y	X es igual a Y
≠	!=	X != Y	X no es igual a Y
Operadores relacionales			
>	>	X > Y	X es mayor que Y
<	<	X < Y	X es menor que Y
≥	>=	X >= Y	X es mayor o igual que Y
≤	<=	X <= Y	X es menor o igual que Y

Fig. 3.1 Operadores de igualdad y relacionales (1 de 2).

El ejemplo de la figura 3.2 utiliza diferentes tipos de líneas de código para la función if para comparar dos números escritos por el usuario. Esto si cumple la condición de estas líneas de código se ejecutará otra línea de código printf asociado con dicho if.

```
main ()
{
    int num1, num2;
    printf ("Ingresa dos números enteros y te diré el resultado:");
    scanf ("%d%d", num1, num2);
    if (num1 == num2)
        printf ("%d es igual que %d", num1, num2);
    if (num1 != num2)
        printf ("%d no es igual que %d", num1, num2);
    if (num1 < num2)
        printf ("%d es menor que %d", num1, num2);
    if (num1 > num2)
        printf ("%d es mayor que %d", num1, num2);
    if (num1 <= num2)
        printf ("%d es menor o igual que %d", num1, num2);
    if (num1 >= num2)
        printf ("%d es mayor o igual que %d", num1, num2);
    return 0;
}
```

Realizar la actividad del material de apoyo y luego enfoca este marcador.



Si en la aplicación una vez realizada la actividad muestra un botón en la parte inferior con la leyenda “AR: Retroalimentación” tendrás que enfocar el siguiente marcador.



La figura 3.3 muestra de donde provienen dichos operadores presentados en este tema.

Operadores	Asociatividad
()	De izquierda a derecha
* / %	De izquierda a derecha
+ -	De izquierda a derecha
< <= > >=	De izquierda a derecha
== !=	De izquierda a derecha
=	De derecha a izquierda

Fig. 3.3 asociatividad de los operadores.

Apéndice B

Código del prototipo móvil

Código del Tema 1

```
import Foundation
import UIKit

class menuT1ViewController: UIViewController {
    override func viewDidLoad() {
        super.viewDidLoad()
        bienvenida()
    }

    func bienvenida(){
        let alert = UIAlertController(title:"Tema 1: Benvenido a C", message: "Utiliza tu
material de apoyo para visualizar el contenido de los marcadores.", preferredStyle:
.alert)
        alert.addAction(UIAlertAction(title:"OK", style: .default, handler: nil))
        self.present(alert, animated: true, completion: nil)
    }
}
```

```

import Foundation
import UIKit
import ARKit

class arHolaViewControllar: UIViewController, ARSCNViewDelegate{

    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {

        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 0, green: 0, blue: 0, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema1/AR_Tema1_1BienvenidoaC.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)
            node.addChildNode(planeNode)
        }
        return node
    }
}

```

```

import Foundation
import UIKit
import ARKit

class arSaltoViewController: UIViewController, ARSCNViewDelegate{

    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {

        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 0, green: 0, blue: 0, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema1/AR_Tema1_2_CambioLinea.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)

            node.addChildNode(planeNode)
        }

        return node
    }
}

import Foundation

```

```

import UIKit
import ARKit

class arSecuenciaViewController: UIViewController, ARSCNViewDelegate {

    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {

        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 0, green: 0, blue: 0, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema1/AR_Tema1_3_CambioLinea.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)

            node.addChildNode(planeNode)
        }

        return node
    }
}

```

Código del tema 2

```
import Foundation
import UIKit

class manuT2ViewController: UIViewController{
    override func viewDidLoad() {
        super.viewDidLoad()
        bienvenido()
    }

    func bienvenido(){
        let alert = UIAlertController(title:"Tema 2: Operaciones matemáticas en C",
        message: "Utiliza tu material de apoyo para visualizar el contenido de los marcadores.",
        preferredStyle: .alert)
        alert.addAction(UIAlertAction(title:"OK", style: .default, handler: nil))
        self.present(alert, animated: true, completion: nil)
    }
}
```

```

import Foundation
import ARKit

class arMemoriaViewController: UIViewController, ARSCNViewDelegate {

    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {

        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 1, green: 1, blue: 1, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema2/AR_Tema2_2_Memoria.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)
            node.addChildNode(planeNode)
        }
        return node
    }
}

```

```

import Foundation
import ARKit

class arOperacionesViewController: UIViewController, ARSCNViewDelegate{

    @IBOutlet weak var sceneView: ARSCNView!
    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {

        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 1, green: 1, blue: 1, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema2/AR_Tema2_1_Suma.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)
            node.addChildNode(planeNode)
        }
        return node
    }
}

```

```

import Foundation
import ARKit

class arResultadoViewController: UIViewController, ARSCNViewDelegate {
    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {
        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 1, green: 1, blue: 1, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema2/AR_Tema2_3_Resultado.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)
            node.addChildNode(planeNode)
        }
        return node
    }
}

```

Código del tema 3

```
import Foundation
import UIKit

class menuT3ViewController: UIViewController{
    override func viewDidLoad() {
        super.viewDidLoad()
        bienvenido()
    }

    func bienvenido(){
        let alert = UIAlertController(title:"Tema 3: Condiciones en C", message: "Utiliza
tu material de apoyo para visualizar el contenido de los marcadores.", preferredStyle:
.alert)
        alert.addAction(UIAlertAction(title:"OK", style: .default, handler: nil))
        self.present(alert, animated: true, completion: nil)
    }
}
```

```

import Foundation
import ARKit

class arCondicionesViewController: UIViewController, ARSCNViewDelegate{
    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {
        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 1, green: 1, blue: 1, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema2/AR_Tema2_3_Resultado.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)
            node.addChildNode(planeNode)
        }
        return node
    }
}

```

```

import Foundation
import ARKit

class arMemoriaCViewController: UIViewController, ARSCNViewDelegate{
    @IBOutlet weak var sceneView: ARSCNView!

    override func viewDidLoad() {
        super.viewDidLoad()
        sceneView.delegate = self
        let scene = SCNScene(named: "art.scnassets/GameScene.scn")!
        sceneView.scene = scene
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        let configuration = ARImageTrackingConfiguration()
        guard let trackedImages = ARReferenceImage.referenceImages(inGroupNamed: "Photos",
bundle: Bundle.main) else {
            print("No images available")
            return
        }
        configuration.trackingImages = trackedImages
        configuration.maximumNumberOfTrackedImages = 1
        sceneView.session.run(configuration)
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        sceneView.session.pause()
    }

    func renderer(_ renderer: SCNSceneRenderer, nodeFor anchor: ARAnchor) -> SCNNode? {
        let node = SCNNode()

        if let imageAnchor = anchor as? ARImageAnchor {
            let plane = SCNPlane(width: imageAnchor.referenceImage.physicalSize.width, height:
imageAnchor.referenceImage.physicalSize.height)

            plane.firstMaterial?.diffuse.contents = UIColor(red: 1, green: 1, blue: 1, alpha:
1)

            let planeNode = SCNNode(geometry: plane)
            planeNode.eulerAngles.x = -.pi / 2

            let shipScene = SCNScene(named: "art.scnassets/Tema2/AR_Tema2_3_Resultado.scn")!

            let shipNode = shipScene.rootNode.childNodes.first!
            shipNode.position = SCNVector3Zero
            shipNode.position.x = -2.5
            shipNode.position.z = 10

            planeNode.addChildNode(shipNode)
            node.addChildNode(planeNode)
        }
        return node
    }
}

```

AppDelegate, código que siempre debe de existir para el funcionamiento sin problemas de todos los ViewControllers.

```
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        return true
    }

    func applicationWillResignActive(_ application: UIApplication) {
        // Sent when the application is about to move from active to inactive state. This can
        occur for certain types of temporary interruptions (such as an incoming phone call or SMS
        message) or when the user quits the application and it begins the transition to the background
        state.
        // Use this method to pause ongoing tasks, disable timers, and invalidate graphics
        rendering callbacks. Games should use this method to pause the game.
    }

    func applicationDidEnterBackground(_ application: UIApplication) {
        // Use this method to release shared resources, save user data, invalidate timers, and
        store enough application state information to restore your application to its current state in
        case it is terminated later.
        // If your application supports background execution, this method is called instead of
        applicationWillTerminate: when the user quits.
    }

    func applicationWillEnterForeground(_ application: UIApplication) {
        // Called as part of the transition from the background to the active state; here you
        can undo many of the changes made on entering the background.
    }

    func applicationDidBecomeActive(_ application: UIApplication) {
        // Restart any tasks that were paused (or not yet started) while the application was
        inactive. If the application was previously in the background, optionally refresh the user
        interface.
    }

    func applicationWillTerminate(_ application: UIApplication) {
        // Called when the application is about to terminate. Save data if appropriate. See
        also applicationDidEnterBackground:.
    }
}
```