

UNIVERSIDAD NACIONAL DEL CALLAO

**ESCUELA DE POSGRADO
UNIDAD DE POSGRADO DE LA FACULTAD DE
INGENIERÍA INDUSTRIAL Y DE SISTEMAS**



“SIMULADORES DE ARQUITECTURA DE COMPUTADORAS Y SU
RELACIÓN CON EL RENDIMIENTO ACADÉMICO DE LOS ESTUDIANTES
EN LA ASIGNATURA DE ARQUITECTURA DE COMPUTADORAS, EN LA
UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA”

TESIS PARA OPTAR EL GRADO ACADÉMICO DE
MAESTRO EN INGENIERÍA DE SISTEMAS

EUDOLIO GREGORIO VASQUEZ PINEDO

Callao, 2021
PERÚ

HOJA DE REFERENCIA DEL JURADO

Jurado de Sustentación de Tesis:

Dr. Alejandro Danilo Amaya Chapa Presidente

Mg. Héctor Gavino Salazar Robles Secretario

Dra. Sally Karina Torres Alvarado Vocal

Asesor: Dra. Erika Juana Zevallos Vera

Libro de actas N° 01

Acta N° 001-2021-UPG-FIIS

Folio N° 45

Fecha de aprobación de la tesis: 22 de abril de 2021.

DEDICATORIA

A DIOS, que nos guía por el camino del bien; a mis familiares, por su paciencia, comprensión y apoyo para realizar este trabajo.

AGRADECIMIENTO

A los profesores de la maestría en Ingeniería de Sistemas de la Universidad Nacional del Callao, a la Dra. Érika Zevallos, por sus orientaciones en el desarrollo de este trabajo, a los profesores de la FIIS –UNAS, por su apoyo en la realización del presente trabajo.

ÍNDICE.....	Pág.
I. PLANTEAMIENTO DEL PROBLEMA	10
1.1 Descripción de la realidad problemática	10
1.2 Formulación del problema	14
1.2.1 Problema general	14
1.2.2 Problemas específicos	14
1.3 Objetivos	14
1.3.1 Objetivo general	14
1.3.2 Objetivos específicos	15
1.4 Limitantes de la investigación	15
II. MARCO TEÓRICO	17
2.1 Antecedentes	17
2.1.1 Antecedentes internacionales.....	17
2.1.2 Antecedentes nacionales	19
2.2 Bases teóricas.....	20
2.3 Conceptual.....	26
2.4 Definición de términos básicos	27
III. HIPÓTESIS Y VARIABLES.....	30
3.1 Hipótesis.....	30
3.1.1 Hipótesis general.....	30
3.1.2 Hipótesis específicas.....	30
3.2 Definición conceptual de variables.....	30
3.2.1Operacionalización de variables	31
IV. DISEÑO METODOLÓGICO	34
4.1 Tipo y diseño de investigación	34
4.2 Método de investigación	37
4.3 Población y muestra.....	39
4.4 Lugar de estudio y periodo de desarrollo.....	39
4.5 Técnicas e instrumentos para la recolección de la información.....	39
4.6 Análisis y procesamiento de datos.....	42
V. RESULTADOS	57
5.1 Resultados descriptivos.....	57
5.2 Resultados inferenciales.....	63
VI. DISCUSIÓN DE RESULTADOS	67
6.1 Contrastación y demostración de la hipótesis con los resultados	67
6.2 Contrastación de resultado con otros estudios similares	73

6.3 Responsabilidad ética de acuerdo con los reglamentos vigentes.....	75
CONCLUSIONES	76
RECOMENDACIONES	77
REFERENCIAS BIBLIOGRÁFICAS	78
Bibliografía.....	78
ANEXOS.....	82
ANEXO 01: Matriz de consistencia	83
ANEXO 02: Instrumentos validados	85
ANEXO 03: Base de datos	94
ANEXO 04: Reporte de estudiantes de ingeniería en informática y sistemas, semestre 2019-1	97
ANEXO 05: Apreciación del simulador ARMSim#.....	98
ANEXO 06: Guía de prácticas	102

TABLAS DE CONTENIDO

<i>Tabla 1 Matriz de operacionalización de variables</i>	<i>32</i>
<i>Tabla 2 Escala de calificación en la UNAS.....</i>	<i>40</i>
<i>Tabla 3 Estadísticos descriptivos de la dimensión cognitiva del postest.....</i>	<i>57</i>
<i>Tabla 4 Estadísticos descriptivos de la dimensión procedimental del postest....</i>	<i>59</i>
<i>Tabla 5 Estadísticos descriptivos del rendimiento académico en el postest</i>	<i>59</i>
<i>Tabla 6 Estadísticos descriptivos del rendimiento académico en el postest, según escala de la UNAS.....</i>	<i>60</i>
<i>Tabla 7 Estadísticos descriptivos de las dimensiones conceptual y procedimental del pre test</i>	<i>61</i>
<i>Tabla 8 Estadísticos descriptivos de las dimensiones conceptual y procedimental del postest.....</i>	<i>62</i>
<i>Tabla 9 Estadísticos inferenciales, prueba de normalidad del pre test y postest del rendimiento académico.....</i>	<i>63</i>
<i>Tabla 10 Estadísticos inferenciales, prueba de normalidad del pre test y postest de las dimensiones conceptual y procedimental</i>	<i>64</i>
<i>Tabla 11 Resultados prueba U de Mann Whitney, del pre test y postest del rendimiento académico</i>	<i>65</i>
<i>Tabla 12 Resultados de la prueba U de Mann Whitney de la dimensión conceptual en el pre test y postest</i>	<i>66</i>
<i>Tabla 13 Resultados prueba U de Mann Whitney de la dimensión procedimental en el pre test y postest.....</i>	<i>66</i>

TABLA DE GRÁFICOS

<i>Grafico 1 Histograma de la evaluación conceptual del grupo experimental</i>	<i>58</i>
<i>Grafico 2 Histograma de la evaluación conceptual del grupo de control.....</i>	<i>58</i>
<i>Grafico 3 Calificaciones en el postest en la escala de la UNAS.....</i>	<i>60</i>
<i>Grafico 4 Diagrama de cajas del Rendimiento académico de los grupos control y experimental, (a) pretest y (b) postest</i>	<i>68</i>
<i>Grafico 5 Diagrama de cajas de los resultados conceptuales pretest y postest..</i>	<i>70</i>
<i>Gráfico 6 Diagrama de cajas de la dimensión contenido procedimental pretest y postest.....</i>	<i>72</i>
<i>Gráfico 7 Facilidad de aprender usar los registros del procesador ARM</i>	<i>99</i>
<i>Gráfico 8 Satisfacción de realizar operaciones con los registros del ARMSim#..</i>	<i>99</i>
<i>Gráfico 9 Satisfaccion de haber realizado operaciones con el del ARMSim#... </i>	<i>100</i>
<i>Gráfico 10 Satisfacción de haber realizado programas en el ARMSim#.....</i>	<i>100</i>
<i>Gráfico 11 El simulador favoreció el aprendizaje de arquitectura de computadoras</i>	<i>101</i>

TABLA DE IMÁGENES Y OTROS

<i>Figura 1 Modelo por nivel del computador.....</i>	<i>21</i>
<i>Figura 2 Modelo funcional de un computador básico.....</i>	<i>22</i>
<i>Figura 3 Entorno de desarrollo del simulador ARMSim#.....</i>	<i>25</i>
<i>Figura 4. Ilustración de las formas y tipos de investigación</i>	<i>34</i>
<i>Figura 5. Relación causa efecto</i>	<i>37</i>

RESUMEN

El desarrollo de temas como la organización y arquitectura de computadoras requiere bastante esfuerzo, tanto por parte del profesor como del alumno, para su cabal comprensión. Parte de la dificultad es la carencia, para las clases prácticas, del hardware o equipamiento adecuados, necesarios para que los estudiantes relacionen la teoría con la práctica y logren un aprendizaje óptimo. En la presente investigación se planteó, como propósito, determinar si el uso de un simulador influye favorablemente en el rendimiento académico de los alumnos en los diversos temas de arquitectura de computadoras. El estudio se realizó sobre una muestra de 35 estudiantes de segundo año de la Escuela Profesional de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva, Tingo María. La muestra inicial fue dividida, de forma no probabilística, en dos grupos: 17 conformaron el grupo experimental y 18 el grupo de control. El instrumento utilizado para evaluar fue un cuestionario, aplicado tanto en un pretest como en un postest. Los resultados evidencian diferencias significativas entre los que utilizaron el simulador para los ejercicios prácticos y aquellos que no hicieron uso de este. En consecuencia, la conclusión es que el uso de un simulador, en las clases prácticas, influye positivamente en el rendimiento académico de los alumnos.

Palabras claves: Simuladores, simuladores basados en software, organización y arquitectura de computadoras, ARM.

ABSTRACT

The development of topics such as organization and architecture of computers requires a great deal of effort on the part of both the teacher and the student to understand them fully. Part of the difficulty is the lack, for practical classes, of the adequate hardware or equipment, necessary for students to relate theory to practice and achieve optimal learning. The purpose of this research was to determine whether the use of a simulator has a favorable influence on the academic performance of students in the various topics of computer architecture. The study was conducted on a sample of 35 second-year students from the Professional School of Computer and Systems Engineering at the Universidad Nacional Agraria de la Selva, Tingo María. The initial sample was divided, in a non-probabilistic way, into two groups: 17 formed the experimental group and 18 the control group. The instrument used to evaluate was a questionnaire, applied both in a pre-test and a post-test. The results show significant differences between those who used the simulator for the practical exercises and those who did not. Consequently, the conclusion is that the use of a simulator, in the practical classes, has a positive influence on the academic performance of the students.

Keywords: simulators, software-based Simulators, computer architecture and organization, ARM.

INTRODUCCIÓN

Comprender temas de organización arquitectura de computadoras es una tarea ardua debido a que los diversos temas que aborda son abstractos, complejos o requieren conocimientos avanzados; en algunos casos la enseñanza clásica en salón provoca que los estudiantes pierdan el interés y la motivación para aprender estos temas. Por ello, se requieren métodos de enseñanza que fomenten el interés y mejoren su aprendizaje.

Por lo general las instituciones que imparten estas asignaturas no cuentan con el hardware, equipamiento o módulos, para las clases prácticas. Ante esta situación, los simuladores son una alternativa para la enseñanza práctica de los temas más arduos de organización y arquitectura de computadoras.

La presente investigación, trata de comprobar si el uso de simuladores en las clases prácticas de organización y arquitectura de computadoras despierta el interés de los alumnos y mejora su rendimiento académico. El trabajo de investigación está organizado tal como se detalla a continuación.

En el capítulo 1 se presenta el planteamiento de la investigación. Se describe la realidad problemática y se formula el problema en forma de preguntas, luego se determinan los objetivos y limitantes de la investigación.

El capítulo 2 trata sobre el marco teórico, se presentan los antecedentes a nivel nacional, e internacional, sobre el uso de los simuladores en la enseñanza de arquitectura de computadoras. También se exponen las bases teóricas y conceptuales que fundamentan la investigación y la definición de términos básicos.

En el capítulo 3 se presenta la formulación de hipótesis, la general y las específicas; así mismo, la identificación de la variable independiente (VI) y la variable dependiente (VD); también, se incluye la operacionalización de variables.

El capítulo 4 trata sobre el diseño metodológico. Se sustentan el tipo y diseño de investigación, la población y muestra, lugar, periodo, técnicas e instrumentos de recolección de datos. También, se presenta el análisis y procesamiento de datos, realizado en el programa R. En el procesamiento se

emplea la prueba de normalidad Shapiro Wilk. Así mismo, se realiza el análisis de la diferencia entre los grupos experimental y el de control.

En el capítulo 5 se presentan los resultados obtenidos luego de ser procesados en el programa estadístico R.

En el capítulo 6 se realiza la discusión de los resultados y la demostración de las hipótesis. Se incluye una contrastación de los resultados obtenidos con otros resultados de estudios similares.

Finalmente, se presentan las conclusiones y recomendaciones del trabajo; así como las referencias bibliográficas consultadas y los anexos con información importante.

I. PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción de la realidad problemática

Es innegable el rol y utilidad insustituibles de las computadoras en la actualidad, tanto para los profesionales de la informática, como para los usuarios finales en prácticamente todos los ámbitos de la actividad humana. Los profesionales de la informática, en sus diversas especializaciones, deben tener un grado aceptable de conocimientos sobre la interacción entre el hardware y software.

Existen organizaciones internacionales relacionadas con la computación e informática y que constantemente promueven el desarrollo de estas disciplinas. La *Association for Computing Machinery* (ACM), la cual agrupa a profesionales relacionados con la computación, tiene como una de sus actividades la promoción de la educación en la ciencia de computación. Así mismo, el *Institute of Electrical and Electronic Engineers* (IEEE), fomenta la educación en diversas disciplinas tecnológicas.

Según la ACM, las especialidades de la computación son diversas. Esta organización ha publicado currículos de estudio para las siguientes especialidades: Computer Engineering, Computer Science, Information Systems, Information Technology y Software Engineering. Asimismo, hay un currículo de estudio para Ciberseguridad (Association for Computing Machinery, 2019).

Existe un currículo de estudio para Computer Sciences recomendado por la ACM, data del año 2013; en él aparecen cambios respecto al currículo del 2008. Para los tópicos de organización y arquitectura de computadoras (AR), se observa que la ACM reduce las horas de 36 a 16. Sin embargo, se agrega una nueva asignatura denominada System Fundamentals (SF). Esta incluye algunos conceptos previos encontrados en (AR) (Association for Computing Machinery, 2019).

Para el caso del Perú, se conocen las denominaciones de las especialidades relacionadas con la informática. Algunas de ellas son la Ingeniería de Sistemas, la Ingeniería Informática, la ingeniería de computación, entre otras denominaciones. En el Perú no se tienen las especialidades como recomienda

la ACM, salvo en algunas universidades y con solo algunas especialidades (como ciencias de computación e ingeniería de software). También, las universidades del Perú, con escuelas profesionales relacionadas con la informática, hacen esfuerzos por adaptarse a las recomendaciones de la ACM en cuanto a los currículos recomendados por ella.

Las universidades peruanas, que cuentan con escuelas profesionales relacionadas con la informática, incluyen en sus currículos de estudio diversos tópicos sobre organización y arquitectura de computadoras según se puede constatar en sus respectivas páginas web.

Existen diversas opiniones respecto a la conveniencia de que si los estudiantes de pregrado de informática deban conocer los tópicos de organización y arquitectura de computadoras. Muchos autores indican que la comprensión del sistema hardware del computador es un tema básico para los estudiantes de informática (Kurniawan & Mochammad, 2017).

Al respecto surgen algunas interrogantes como las siguientes: ¿los estudiantes de informática deberían conocer sobre el hardware del computador? o, dado que los informáticos escriben programas en lenguajes de alto nivel, ¿entonces, deberían entender como estos se ejecutan a bajo nivel? También, ¿qué mejorar cuando el programa requerido deba ser más rápido y eficiente o las aplicaciones no hacen precisamente lo que se requiere? Nuevamente, diversos autores señalan que el científico de la informática necesita entender la organización y arquitectura del computador para rectificar los problemas (Null & Lobur, 2015).

Existen definiciones de arquitectura del computador y de organización del computador, los autores consideran que están muy relacionados. La arquitectura del computador estudia los atributos del computador desde el punto de vista del programador. Por ejemplo, el conjunto de instrucciones, el uso de aritmética de coma flotante, los modos de direccionamiento, los mecanismos de entrada/salida (I/O) y otros temas. Mientras que la organización de computadoras estudia la parte asociada con las unidades operacionales del computador y las relaciones entre los componentes del computador. Por ejemplo, las señales de control, las interfaces, las diversas tecnologías de memoria, entre otros (Kurniawan &

Mochammad, 2017). Así, los temas que se abordan son fundamentales y de gran relevancia. Estrictamente hablando, según los perfiles internacionales de las especialidades de la informática, *Computer Engineering* es la especialidad que desarrolla completamente los tópicos de organización y arquitectura del computador; por su parte, *Computer Science*, desarrollar solamente algunos tópicos. En el Perú, las especialidades relacionadas con la informática consideran en sus currículos solo la asignatura de organización y arquitectura del computador. En cuanto a los temas que consideran, no hay una regla específica.

Existen diversos términos relacionados al computador y sus definiciones. Entre estos tenemos organización del computador y arquitectura del computador. “La arquitectura de computadores se refiere a los atributos de un sistema que son visibles a un programador, o para decirlo de otra manera, aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa” (Stallings, 2007, p. 8). Se consideran atributos de arquitectura, el conjunto de instrucciones, el número de bits usados para representar los tipos de datos, mecanismos de entrada salida y técnicas para direccionamiento de memoria. Respecto a la definición de organización del computador, una de ellas es: “La organización de computadores se refiere a las unidades funcionales y sus interconexiones, que dan lugar a especificaciones arquitectónicas” (Stallings, 2007, p. 8). Algunos ejemplos de atributos de organización son detalles de hardware transparentes al programador, tales como señales de control, interfaces entre el computador y los periféricos, la tecnología de memoria usada, entre otros.

Existe una estrecha relación entre el hardware del computador y muchos aspectos de programación y componentes software de un sistema computador. Para escribir buen software es importante entender el computador como un todo. Entender el hardware puede ayudar a explicar los mensajes de error que suelen aparecer cuando se ejecutan los programas. El nivel de conocimiento sobre la organización y arquitectura del computador, que un programador de alto nivel debe tener, depende de la tarea que desarrollará. Por ejemplo, para escribir compiladores se debe entender el hardware particular en el cual se ejecutará.

Por ejemplo, el hardware que implementa el pipeline. Para escribir controladores de dispositivos de video, discos, o dispositivos de entrada/salida, se necesita conocer bien las interfaces de entrada/salida y la arquitectura en general. Para trabajar con sistemas embebidos se necesita entender el espacio de memoria y parte de la arquitectura. También, antes de comprar o emplear una máquina, se necesita acceder a los benchmarking y verificar que sistema es mejor que otro. Por todo ello, el campo del científico de informática involucra entender como el hardware interactúa con el software.

Resumiendo, los temas de arquitectura y organización de computadoras son fundamentales para los estudiantes y profesionales de informática. Es necesario comprender los componentes funcionales de un sistema informático, sus características, rendimiento e interacciones, para hacer programas eficientes. Sin embargo, la enseñanza de Arquitectura de Computadoras en cualquier nivel no es una tarea fácil. Es una de las asignaturas que presenta dificultades para los estudiantes. Por ello, es necesario el apoyo con recursos y medios didácticos de tal modo que se mejore el aprendizaje y rendimiento académico a nivel de pregrado (Cecile Yehezkel, William Yurcik, Murray Pearson, Dean Armstrong, 2001).

Algunos autores mencionan que los lenguajes de programación de alto nivel no ilustran de forma clara cómo un programa es ejecutado por el computador. Ello hace decrecer el interés, por parte de los estudiantes, en aprender sobre tópicos de organización y arquitectura del computador (Atanasovski, Ristov, Gusev, & Anchev, 2013). La enseñanza de arquitectura de computadoras es un proceso difícil que requiere un gran esfuerzo por parte de los instructores y estudiantes involucrados (Stolikj, Ristov y Ackovska, citado por Atanasovski).

En la Facultad de Ingeniería en Informática y Sistemas (FIIS) de la Universidad Nacional Agraria de la Selva (UNAS), el currículo incluye temas de arquitectura de computadoras. Están ubicadas en el tercer ciclo y con un total de 60 horas. Según datos históricos, los estudiantes de la UNAS tienen dificultades con dicha asignatura, reflejadas en bajas calificaciones y con un porcentaje alto de desaprobados y alumnos que desertan (fuente OCDA). A pesar de que el

rendimiento académico está determinado por diversos factores, creemos que, con un buen material didáctico, como el uso de simuladores modernos con los cuales los estudiantes interaccionen, puede propiciar la mejora del rendimiento académico. Finalmente, se debe mencionar que, en la UNAS, para las prácticas de laboratorio, por varios años se usó el emulador EMU8086; en el año 2017 se utilizó el simulador MARS para procesadores MIPS; el año 2018 se empleó el mismo simulador para trabajar con procesadores ARM; y, en el año 2019, se utilizó el simulador ARMSim# para procesadores ARM.

1.2 Formulación del problema

1.2.1 Problema general

¿De qué manera influye el Simulador de Arquitectura de Computadoras ARMSim# en el rendimiento académico en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019?

1.2.2 Problemas específicos

- a) ¿De qué manera influye el Simulador de Arquitectura de Computadoras ARMSim# en los resultados conceptuales en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019?
- b) ¿De qué manera influye el Simulador de Arquitectura de Computadoras ARMSim# en los resultados procedimentales en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019?

1.3 Objetivos

1.3.1 Objetivo general

Determinar la influencia del Simulador de Arquitectura de Computadoras ARMSim# en el rendimiento académico en temas de arquitectura de

computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019.

1.3.2 Objetivos específicos

- a) Determinar la influencia del Simulador de Arquitectura de Computadoras ARMSim# en los resultados conceptuales en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019.
- b) Determinar la influencia del Simulador de Arquitectura de Computadoras ARMSim# en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019.

1.4 Limitantes de la investigación

1.4.1 Teórico

En la literatura existen diversas definiciones de rendimiento académico; se muestran algunas. El rendimiento académico según (Grao, citado por Coronas) debe analizarse en el contexto social, cultural, económico y político. También considerar lo familiar, académico y personal. Otro autor indica que el rendimiento académico consiste en utilizar procedimientos tradicionales como: puntaje en las asignaturas, el rendimiento en pruebas elaboradas propias para la investigación y el número de créditos aprobados (Martin, 2008).

El presente estudio tiene la limitante que se detalla a continuación. En la UNAS, los temas de arquitectura de computadores forman parte de la asignatura de Sistemas Operativos I. Por ello, los estudiantes que conformaron los grupos experimental y el de control, pertenecían al curso de Sistemas Operativos I. Según la disponibilidad de turnos, el horario del grupo experimental fue los jueves de 08:00 am a 11:00 am, los jueves de 02:00 am a 04:00 pm y los viernes de 08:00 am a 10:00 am. El horario del grupo de control fue los lunes 08:00 am a 11:00 am y los martes de 02:00 pm a 06:00 pm. Debido a ello, no se utilizó una

técnica estadística para conformar los grupos experimental y de control; el estudio es observacional, por lo cual esto fue una limitante.

1.4.2 Temporal

Una limitante fue el periodo de tiempo que se dispuso para hacer el estudio. Considerando que un semestre académico en la universidad dura entre dieciséis a diecisiete semanas durante las cuales se desarrollan las clases, evaluaciones y resultados de las asignaturas. El estudio se realizó en las primeras ocho semanas del semestre académico 2019-I, específicamente los meses de abril y mayo, debido a que los temas de arquitectura de computadoras solamente disponen de ese tiempo al formar parte de la asignatura Sistemas Operativos I en el nuevo plan de estudios de la especialidad de Ingeniería en Informática y Sistemas.

1.4.3 Espacial

El estudio se desarrolló en las instalaciones en la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva. La Universidad Nacional Agraria de la Selva está ubicada en la región Huánuco, provincia Leoncio Prado, distrito Rupa Rupa, Tingo María.

II. MARCO TEÓRICO

2.1 Antecedentes

Para la presente investigación se revisó diversas investigaciones nacionales e internacionales. Los aportes de cada uno de ellos, al presente trabajo, se describen a continuación.

2.1.1 Antecedentes internacionales

- a) (Aguilar Juárez & Heredia Alonso, 2013), en la investigación “Simuladores y laboratorios virtuales para Ingeniería en Computación”, hacen una propuesta en base al currículo de estudios de licenciatura de Ingeniería en Computación de la Universidad Autónoma del Estado de México y proponen para seis aéreas académicas, de un total de ocho que posee el programa, adoptar el uso de simuladores en los laboratorios virtuales a través de la web para la enseñanza de pregrado.
- b) (Barrachina Mir, Fabregat Llueca, Fernández Fernández, & León Navarro, 2015), en la investigación “ARMSim# y QtARMSim: simulador de ARM para la docencia”, trata sobre el desarrollo de un simulador del procesador ARM que se realizó en la Universidad de Jaume I. Así mismo, dicho procesador fue utilizado en los trabajos prácticos de laboratorios y concluyen en que mejora el porcentaje de aprobados en las asignaturas que las utilizaron.
- c) (Alasdoon, Prasad, & Chan, 2015) en la investigación “*Using Simulators for Teaching Computer Organization and Architecture*”, mencionan que un aspecto importante en la enseñanza de temas de arquitectura y organización de computadoras es enlazar conocimientos teóricos con la experiencia práctica. En esta investigación se consideran la usabilidad y la disponibilidad y cubre los contenidos del curso mediante varios simuladores. Describen la experiencia de incorporar simuladores en la enseñanza de temas de arquitectura y organización de computadoras a estudiantes de las especialidades de tecnología de la información y computación. Demostraron que el uso de simuladores ayuda a los estudiantes a

entender mejor cómo un computador está construido y cómo es su funcionamiento interno.

- d) (Yoshiro Imai, Masatoshi Imai, Yoshio Moritoh, 2013), en *“Evaluation of visual computer simulator for computer architecture education”*, hacen la evaluación de un simulador visual de computador; fue realizado durante los años 2009 a 2011 para mejorar el entendimiento de las instrucciones como herramienta de aprendizaje. Así mismo, incluye ejemplos educativos de arquitectura de computadora para estudiantes universitarios y el uso de herramientas e-Learning para programación en lenguaje ensamblador en las sesiones prácticas. Las evaluaciones de resultados se basaron en analizar las instrucciones internas con el uso de un computador Von Neumann; los estudiantes realizaron ejercicios de programación en ensamblador con el simulador. También la segunda parte describe cómo organizar múltiples sistemas servidores para aprendizaje colaborativo con el simulador visual. Finalmente se confirma, por medio de la estadística, que el uso del simulador VisuSim es efectivo para el aprendizaje de *“Computer Literacy”*.
- e) (Jorge Fernando Maxnuck Soares, Luís Tadeu M. Raunheite, Takato Kurihara, 2016), *“The use of MARIE CPU Simulator in Computer Architecture Course: A Case Study of student’s perception of learning and performance”*. El objetivo de este trabajo fue mostrar los resultados de un caso de estudio usando aprendizaje activo práctico en la disciplina de arquitectura del computador. La práctica en cuestión consistió en usar el software Marie CPU Simulator como herramienta práctica en el desarrollo del curso. El estudio tuvo como objetivo verificar si el uso de Marie CPU Simulator contribuye a mejorar el aprendizaje de la disciplina de arquitectura de computadoras, especialmente si proporciona una mejor comprensión de las partes que integran la arquitectura de un CPU determinada, con una explicación de la función de las partes y su interrelación. Este estudio muestra los primeros resultados de un estudio más completo sobre el

uso de prácticas de aprendizaje activo, utilizando software en disciplinas de alta tecnología de un curso de sistemas de información. En este estudio se obtuvieron las siguientes conclusiones: 1) el simulador es de fácil uso, 2) se logra relacionar la teoría (mundo abstracto) con la práctica (mundo real) en la disciplina de arquitectura del computador, 3) se encuentran más fáciles los temas luego de usar el simulador, 4) usando el simulador se incrementa el interés por el curso de arquitectura del computador, 5) el uso del simulador facilita entender cómo trabajan los registros, ALU, memoria principal y demás componentes del CPU.

2.1.2 Antecedentes nacionales

En el ámbito nacional se encontró investigaciones que usan simuladores en la enseñanza de temas de electricidad, electrónica, matemática y física. No se encontraron investigaciones en temas relacionados con simuladores de arquitectura de computadores. Sin embargo, se mencionará las investigaciones realizadas en Perú sobre el uso de simuladores en electricidad, electrónica, matemática y física, los cuales se mencionan a continuación.

- a) (Sinchi Yupanqui, 2018) en la tesis “La simulación computacional en el trabajo experimental para el aprendizaje de las asignaturas de circuitos eléctricos en la Facultad de Ingeniería Mecánica de la Universidad Nacional de Ingeniería (EPIME UNI)”, manifiesta que con la incorporación del uso de la simulación computacional se encuentra que existe relación significativa y que la aplicación de un programa evidencia gran ayuda a la mejora del aprendizaje procedimental de los estudiantes de ingeniería mecánica eléctrica de la EPIME UNI.
- b) (Guizado Carmona, Federico Edgard, 2015), en la tesis “Propuesta didáctica de implementación de un simulador computarizado “NI Multisim” en la enseñanza-aprendizaje de la electrónica”, la implementación del NI Multisim consiste de un conjunto de actividades didácticas que le permite a los estudiantes practicar utilizando la realidad virtual. El autor manifiesta que el estudio contribuye al

mejoramiento de la competencia laboral de los estudiantes y con esto permite desarrollar sus capacidades en tal ámbito.

- c) (De la Cruz Ganoa, 2016), “Software Geogebra y su influencia en el aprendizaje de las funciones reales en los estudiantes del primer ciclo de la Facultad de Ciencias Contables de la Universidad Nacional del Callao”, se comprobó la incidencia positiva del software Geogebra en el aprendizaje conceptual, procedimental y actitudinal de funciones reales en el grupo experimental en la que se compara antes y después de su aplicación, la hipótesis de investigación señala que el uso del software Geogebra influye positivamente en el aprendizaje de funciones reales.

2.2 Bases teóricas

En esta sección, se comenta acerca de dos organizaciones que dan las pautas sobre los currículos de estudio de computación o informática. También se plantean algunas definiciones y conceptos de organización y arquitectura de computadores. Se debe mencionar que la *Association for Computing Machinery* (ACM) y el *Institute of Electrical and Electronics Engineers* (IEEE) dan recomendaciones sobre los currículos de estudio de las especialidades de la informática. Las recomendaciones de la ACM y del IEEE, en cuanto a currículos, deberían ser consideradas por las universidades y centros superiores en las especialidades relacionadas a la informática.

En el año 2005, la ACM definió cinco subdisciplinas de la informática. Estas subdisciplinas son: Computer Engineering, Computer Science, Information Systems, Information Technology y Software Engineering. De esa fecha a la actualidad se han realizado nuevas publicaciones relacionadas a los currículos de estas disciplinas (Association for Computing Machinery, 2013). En el currículo recomendado por la ACM, en el año 2013, aparecen cambios respecto al currículo del 2008, específicamente para los tópicos de organización y arquitectura de computadoras (AR); se observa la reducción de 36 horas a 16 horas, pero aparece una nueva asignatura, denominada System Fundamentals (SF), que incluye algunos conceptos previos encontrados en (AR).

También en la literatura informática existen recomendaciones que indican que los profesionales y estudiantes de informática deben relacionar el software con el hardware. “El informático diseña algoritmos, que son implementados como programas en algún lenguaje de programación, tales como Java o C++. Pero, qué hace ejecutar el algoritmo, otro algoritmo, y otro algoritmo hace ejecutar ese algoritmo” (Null y Lobur, 2014, p. 8). Entonces, se puede deducir que las computadoras modernas son realmente implementaciones de algoritmos que ejecutan otros algoritmos. De esto se deriva el principio de equivalencia de hardware y software que dice: “si una tarea puede hacerse por software también podrá realizarse por hardware, y si una operación puede realizarse por hardware también podrá realizarse por software” (Null y Lobur, 2014, p. 3).

A nivel funcional, un computador está conformado por los siguientes bloques: un procesador para interpretar y ejecutar programas, una memoria para almacenar datos y programas, y un mecanismo para transferir datos internamente y al exterior.

Para fines didácticos, la computadora se presenta en niveles y poder, así, entender su complejidad. En la figura 1 se muestran estos niveles, un total de siete. Del nivel 0 al nivel 2, se considera el campo de la arquitectura del computador. El nivel 2, instruction set architecture (ISA) o nivel de máquina, consiste en el lenguaje de máquina reconocido por una arquitectura de computador en particular. Los programas escritos en lenguaje de máquina pueden ser ejecutados directamente por los circuitos electrónicos sin requerir intérpretes, traductores o compiladores.

Nivel 6	Usuario	Programas de aplicación
Nivel 5	Lenguaje de alto nivel	C++, Java, etc.
Nivel 4	Compiladores e interpretes	Código assembler
Nivel 3	Software del sistema	Sistemas operativos
Nivel 2	Máquina	Conjunto de instrucciones
Nivel 1	Control	Microcódigo
Nivel 0	Dispositivos físicos	Circuitos, compuertas, etc.

Figura 1 *Modelo por niveles del computador*

Fuente: Null y Lobur, 2014, p. 35.

El nivel 1, es el nivel de control; realiza la decodificación de las instrucciones para su ejecución sincronizado con transferencias de datos según requiera cada instrucción. La unidad de control interpreta las instrucciones máquina y realiza las acciones requeridas. La unidad de control puede estar diseñado de dos maneras: cableadas o microprogramadas.

El nivel 0, el nivel de lógica digital, es donde se encuentran los componentes físicos del computador, las compuertas e hilos, que implementan los diversos bloques del computador.

Modelo básico de un computador

Los computadores que se utilizan en la actualidad, a pesar de haber pasado cerca de 70 años desde la aparición de la primera computadora de programa almacenado, en realidad siguen el modelo de esta. En los libros se describe que Mauchly y Eckert propusieron la idea de la máquina de programa almacenado. Sin embargo, fue Von Neumann quien publicó dicha idea y de ahí viene la denominación de modelo o máquina de Von Neumann (Patterson & Hennessy, 2000, pp. 29-30).

La primera máquina electrónica con estas características de programa almacenado fue la denominada Electronic Discrete Variable Automatic Computer (EDVAC).

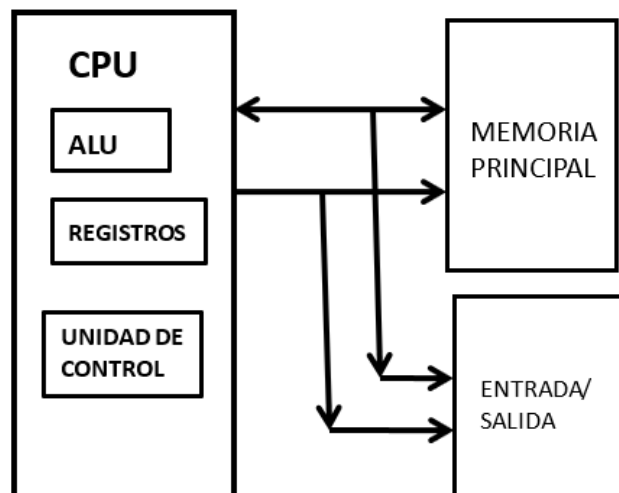


Figura 2 *Modelo funcional de un computador básico*
Fuente: Null y Lobur, 2015, p. 42.

También, se recogieron algunas definiciones de simulación. Según la Real Academia Española, el término simulador tiene el siguiente significado: “Tecnol. Aparato que reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento de quienes deben manejar dicho sistema” (Real Academia Española, 2019).

La simulación, para ciencia e ingeniería, involucra el empleo del modelado y simulación usando la computadora para resolver formulaciones matemáticas de modelos físicos de ingeniería y sistemas naturales (Glotzer, 2011, p. 1).

Sin embargo, la simulación en arquitectura y organización de computadoras tiene sus propias características y algunas se han recogido de ciertos artículos que se describen a continuación.

Según (Cecile Yehezkel, William Yurcik, Murray Pearson, Dean Armstrong, 2001), los simuladores de arquitectura de computadoras son hipotéticos o representan solo a cierta arquitectura de computadoras. Algunos fueron inventados por sus desarrolladores o son complementos de un libro de texto para cumplir con el objetivo pedagógico. Otros están orientados a un computador específico y con cierto nivel de complejidad. Un computador podría modelarse en diferentes capas jerárquicas de hardware y software. Un simulador puede modelar cualquier cosa, desde una operación de alto nivel de un CPU hasta la ejecución de microcódigo. Los desarrolladores crean el entorno de aprendizaje definiendo el rol del estudiante en la simulación, dependiendo del conjunto de habilidades que un estudiante necesita adquirir. Puede ser más apropiado para un estudiante operar la unidad de control del CPU o programar en lenguaje assembler. Los simuladores refuerzan los niveles de comprensión de los estudiantes sobre las operaciones de la CPU durante la ejecución de la instrucción en el primer caso y los procesos de ejecución del programa a nivel de máquina en el último caso.

Existen muchos simuladores de organización y arquitectura de computadoras que se utilizan según los requerimientos de los currículos de estudio. Así, la *computing curricula* señala que computer science (CS) y computer engineering (CE) deben clasificar qué tópicos cursaran los estudiantes (Kurniawan & Hanafi, 2017).

Software simulador de arquitectura de computadoras

Se mencionan algunos simuladores encontrados en investigaciones y publicaciones, estos son los siguientes:

- a. *The Visible Virtual Machine*. Simula el ciclo de lectura de instrucción de una máquina basada en el modelo Von Neumann.
- b. MARS. Un simulador de lenguaje assembler para procesador MIPS.
- c. Logisim. Una herramienta educativa para diseño y simulación de circuitos lógicos digitales. También se puede diseñar y simular una CPU.
- d. SPIM. Este simulador ejecuta lenguaje assembler para un procesador MIPS de 32 bits.
- e. CPU-OS Simulator. Este simulador puede simular un procesador tipo RISC y puede simular procesos y puede simular la administración de memoria de la capa de un sistema operativo.
- f. MARIE. Es un simulador para principiantes en organización y arquitectura de computadoras
- g. QtARMSim. Es un simulador gráfico del procesador ARM
- h. ARMSim#. Es un simulador que simula ejecución de lenguaje assembler para el procesador ARM.

Simulador ARMSim#

En esta sección se presenta el simulador que se utilizó en el presente trabajo y se mencionan algunas de sus características. Este simulador fue desarrollado por Horspool, Lyons y Serra, en la Universidad de Victoria.

El ARMSim# es una aplicación que corre en los entornos Windows, Linux y Mac OSX. Permite a los usuarios simular la ejecución de programas en assembler ARM, basados en el procesador ARM7TDMI. El ARMSim# incluye tanto un enlazador como un ensamblador. Cuando se carga un archivo, el simulador automáticamente ensambla y enlaza el programa (R.N. Horspool, W.D. Lyons, M. Serra, 2010). El ARMSim# también proporciona características que no se encuentran a menudo en aplicaciones similares. Permite a los usuarios tanto depurar los programas en assembler ARM como monitorear el estado del

sistema mientras se ejecuta un programa. La información de monitoreo incluye tanto estados de caché como ciclos de reloj consumidos.

La didáctica con el software ARMSim#

Como recurso didáctico, ARMSim# está diseñado para interactuar con el usuario; consta de barra de tareas, ventanas que muestran el código, registros, entre otros.

El rendimiento académico

Según autores, varios factores influyen en el rendimiento académico: la inteligencia y las estrategias de aprendizaje son algunos de estos (Escalante y Rivas 2002, citado por Johanna Liliana Kohler Herrera). En el presente estudio se consideró únicamente las estrategias didácticas, específicamente al usar simuladores en una asignatura.

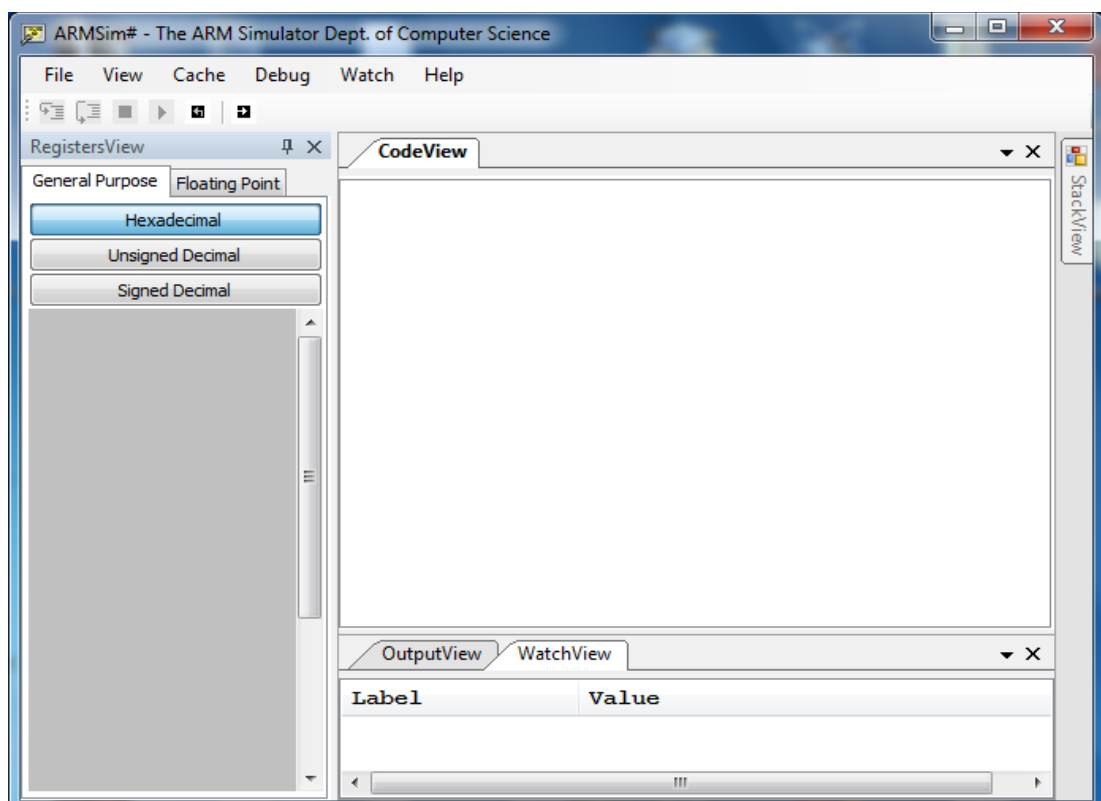


Figura 3 Entorno de desarrollo del simulador ARMSim#, autor R.N. Horspool, W.D. Lyons, M. Serra.

2.3 Conceptual

En este apartado se definen algunos conceptos relacionados con el trabajo de investigación, primero definimos los términos relativos a las variables de investigación, entre otros.

Variable

Existen ciertas definiciones al respecto. Por ejemplo, el libro “Cómo elaborar y asesorar una investigación”, menciona el término **variable de estudio** y dice que “son aquellos fenómenos, características, cualidades, atributos, rasgos o propiedades cuya variación es susceptible de medición.” (Muñoz, 2011, p. 54).

También, la siguiente definición: “Una variable es una propiedad que puede fluctuar y cuya variación es susceptible de medirse u observarse.” (Hernández, Fernández, & Baptista, 2014, p. 105).

La mayoría de las veces, las variables están relacionadas, entonces se pueden clasificar según su relación de dependencia o ubicación en una hipótesis como variable independiente, dependiente e interviniente.

Variable independiente

Determina los cambios en la variable dependiente, se utiliza para describir o medir los factores que se suponen son la causa del problema.

Variable dependiente

Es la afectada por los cambios en la variable independiente. Se denomina también de efecto, se utiliza para describir o medir el problema estudiado.

Constructos

El término constructo se refiere a la relación que existe entre dos o más variables. “Las variables adquieren valor para la investigación científica cuando llegan a relacionarse con otras variables, es decir, si forman parte de una hipótesis o una teoría. En este caso, se les suele denominar constructos o construcciones hipotéticas.” (Hernández, Fernández, & Baptista, 2014, p. 105).

Rendimiento académico

Según (Gonzales 1993, citado por Jiménez Coronas), el rendimiento académico en educación superior es el éxito obtenido por los alumnos en la superación total de cada curso académico.

Según (Martin 2008, citado por Jiménez Coronas), el rendimiento académico en educación superior está determinado por múltiples factores, contextuales y personales, y propone utilizar los indicadores siguientes: notas en asignaturas, el desempeño en pruebas elaboradas ad hoc para la investigación y el número de créditos aprobados.

Simulador de arquitectura de computadoras

Es un software que simula el comportamiento de la parte funcional de una determinada arquitectura de computadora, tales como registros internos, operaciones a nivel de instrucciones, operaciones entre registros, operaciones lógicas, operaciones aritméticas y de transferencia. El simulador puede ser hipotético o puede representar a una determinada familia de procesador.

2.4 Definición de términos básicos

Computador

Aparato para hacer cálculos matemáticos. También denominada como computadora electrónica. Mediante programas, permite almacenar y tratar información, y resolver problemas de diferente índole.

Arquitectura de computadoras

De acuerdo con la RAE, se define como “estructura lógica y física de los componentes de una computadora”. También, Williams Stalling, en su libro Organización y Arquitectura de Computadores, lo define del siguiente modo: “La arquitectura de computadores se refiere a los atributos de un sistema que son visibles a un programador, o para decirlo de otra manera, aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa.”

Según Barrachina. La arquitectura de un computador especifica su modo de comportarse y funcionar de tal manera que sea posible realizar programas correctos para ese computador.

Organización de computadoras

Se refiere a las unidades funcionales y sus interconexiones que dan lugar a especificaciones arquitectónicas. Son los detalles del hardware que son transparentes al programador. Algunos de estos son las señales de control, las

interfaces de la computadora con los periféricos, la tecnología de memoria usada, entre otros.

Unidad central de procesamiento (CPU)

Es el hardware que lee y ejecuta instrucciones, y está conformada mínimamente por la Unidad aritmética-lógica (ALU), unidad de control y registros, interconectados entre ellos.

Procesador

Es el hardware que debe tener por lo menos una CPU.

Simulador

Según la Real Academia Española (RAE) y referido a la tecnología viene a ser: “Aparato que reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento de quienes deben manejar dicho sistema.”

Emulador

Software que permite ejecutar programas en una arquitectura de hardware o un sistema operativo diferente de aquella para la cual fueron escritos originalmente.

Entorno de desarrollo integrado (IDE)

Aplicación informática que proporciona servicios integrales para facilitarle al programador el desarrollo de software.

Rendimiento

Proporción entre el producto o el resultado obtenido y los medios utilizados.

Académico

Perteneciente o relativo a centros oficiales de enseñanza, especialmente a los superiores. Curso, expediente, título académico.

ALU

Unidad aritmética y lógica, parte del procesador donde se realizan las operaciones aritméticas y lógicas

Registro

Memoria de poca capacidad, del orden de bits; guarda instrucciones o datos temporalmente. Los registros están dentro del procesador.

Unidad de control

Parte del procesador que controla el procesamiento de instrucciones.

III. HIPÓTESIS Y VARIABLES

3.1 Hipótesis

3.1.1 Hipótesis general

El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en el rendimiento académico en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

3.1.2 Hipótesis específicas

- a) El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados conceptuales en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.
- b) El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

3.2 Definición conceptual de variables

Variable independiente

Simulador de Arquitectura de computadoras ARMSim#.

Definición conceptual: el simulador de arquitectura de computadoras ARMSim#, es un software emulador del procesador ARM.

Definición operacional: son las capacidades medidas a través de las dimensiones: cualidades didácticas y desarrollo del experimento.

Variable dependiente

Rendimiento académico en temas de arquitectura de computadoras.

3.2.1 Operacionalización de variables

Variable independiente. Simulador de Arquitectura de computadoras ARMSim#.

Definición conceptual: el simulador de arquitectura de computadoras ARMSim#, es un software emulador del procesador ARM.

Definición operacional: son las capacidades medidas a través de las dimensiones: facilidad de uso.

Variable dependiente

Rendimiento académico en temas de arquitectura de computadoras.

Definición conceptual: son los logros alcanzados por el estudiante en el proceso de enseñanza–aprendizaje de temas de arquitectura de computadoras utilizando las competencias cognitivas, procedimental y actitudinal después de usar un simulador.

Definición operacional: son los indicadores obtenidos mediante los exámenes y trabajos encargados en temas de arquitectura de computadoras.

Tabla 1
Matriz de operacionalización de variables

Variables	Dimensiones	Indicadores	Índices	Método	Técnicas/Instrumentos
Variable Independiente Simulador de Arquitectura de computadoras ARMSim#	<ul style="list-style-type: none"> - Simulador de registros - Simulador del procesador (CPU) y sus registros - Simulador de ejecución de instrucciones 	<ul style="list-style-type: none"> - Uso de registros - Operaciones del CPU y registros - Instrucciones del CPU 	<ul style="list-style-type: none"> - Facilidad y satisfacción en el uso de registros (%) - Satisfacción con las operaciones CPU registros (%) - Satisfacción con las instrucciones del CPU (%) 	Encuesta	Escala de opinión sobre la aplicación del simulador
Variable dependiente Rendimiento académico en temas de arquitectura de computadoras	Componente conceptual	<ul style="list-style-type: none"> - Define arquitectura del computador. - Comprende los componentes funcionales del computador. - Identifica términos del computador. - Comprende las características y los elementos del procesador. - Comprende la ejecución de instrucciones del procesador 	Preguntas: 1 2, 3, 4 5 6 y 7 8, 9 y 10	Cuestionario	Prueba objetiva con calificación de 00 a 10, que se puede transformar a la escala vigesimal. de 0 a 20. Escala: Excelente: 20, 19 y 18 Muy bueno: 17, 16 y 15 Bueno: 14 y 13 Regular: 12 y 11 Desaprobado: 10 o menos Fuente: Reglamento de estudios de la UNAS

	Componente procedimental	<ul style="list-style-type: none"> - Destreza para usar los registros del procesador - Destreza para analizar instrucciones lógicas - Destreza en el uso de instrucciones de transferencia, aritméticas, de comparación y de salto. - Destreza para interpretar el procesamiento de instrucciones de desplazamiento 	Preguntas: 11 13, 14 y 15 12, 16, 17, y 18 19 y 20	Cuestionario	Prueba objetiva con calificación de 00 a 10, que se puede transformar a la escala vigesimal de 00-20. Escala: Excelente: 20, 19 y 18 Muy bueno: 17, 16 y 15 Bueno: 14 y 13 Regular: 12 y 11 Desaprobado: 10 o menos Fuente: Reglamento de estudios de la UNAS
--	--------------------------	---	--	--------------	--

Fuente. Elaboración propia

IV. DISEÑO METODOLÓGICO

4.1 Tipo y diseño de investigación

Tipo de investigación

Al revisar la bibliografía, se aprecia que existen diferentes formas de clasificar los tipos de investigación. (Tamayo, 2003, p. 42) menciona que, según los fines que persigue el investigador, las investigaciones se clasifican en dos formas y tres tipos, de estos se desprenden los diferentes estudios; a continuación, se muestra la clasificación en la Figura 4.

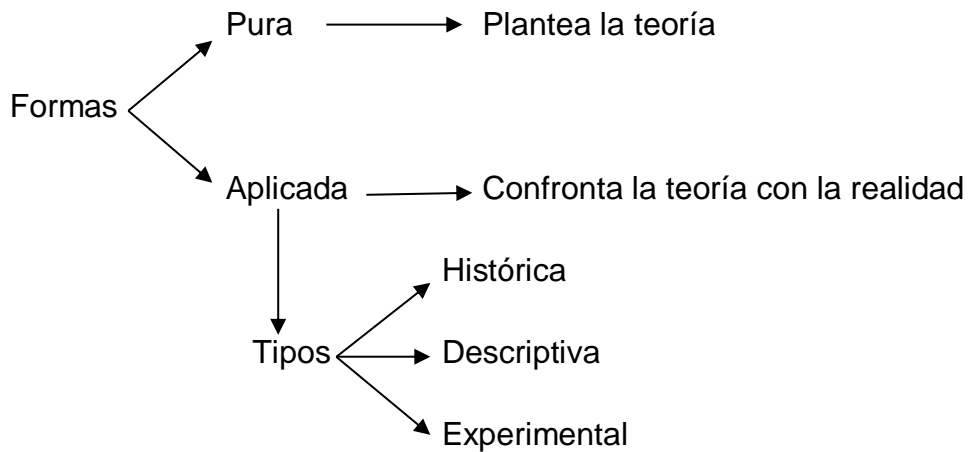


Figura 4. Ilustración de las formas y tipos de investigación
Autor Tamayo, 2003, p. 42

Luego, al clasificar las investigaciones según sus formas, estas son: 1) investigación pura y, 2) investigación aplicada.

La investigación pura, también denominada básica o fundamental, tiene como propósito desarrollar teorías al descubrir principios y sus generalizaciones.

La investigación aplicada, también denominada investigación activa o dinámica, está ligada a la investigación pura es decir depende de los descubrimientos y aportes teóricos de esta.

Ahora, de acuerdo con la clasificación de Tamayo, los tipos de investigación solo aplica a las investigaciones aplicadas y se clasifican en tres tipos: 1) investigación histórica (describe lo que era), 2) investigación descriptiva (interpreta lo que es) y 3) investigación experimental (describe lo que será). Así

mismo menciona que de estos se desprende todos los estudios que realizan los investigadores y además las investigaciones reales por lo general no pertenecen a un solo tipo, sino que son la combinación de estos.

Investigación histórica

De acuerdo con (Tamayo, 2003, p.44) se enfoca a experiencias pasadas y es válida para cualquier disciplina científica.

Investigación descriptiva

Describe, registra, analiza e interpreta la composición de los fenómenos. Ahora la pregunta sería ¿cuáles son los diversos tipos de estudios descriptivos que existen?, al respecto no hay un acuerdo entre los autores. (Tamayo, 2003, pp. 46-47) menciona los siguientes tipos: 1) estudio por encuesta, 2) estudios de casos, 3) Estudios exploratorios, 4) estudios causales, 5) estudios de desarrollo, 6) estudios de conjuntos (integración de los datos), 7) estudio de correlación (la relación de dos o más variables).

Así mismo, se presentan algunas definiciones de tesis dentro del ambiente académico. Una tesis es un trabajo de investigación y no existe una clasificación rigurosa de los tipos de tesis (Muñoz, 2013, p. 5). El autor Muñoz propone una clasificación basada en las formas y niveles en las que puede elaborarse una investigación, considera las siguientes:

- Por su método de investigación
- Por el tratamiento de su tema
- Por la forma de recopilación y por el tratamiento de su información
- Por su nivel de estudios
- De acuerdo con el enfoque de la investigación
- De acuerdo con el objetivo de estudio de la investigación
- De acuerdo con la recopilación de los datos de la investigación
- De acuerdo con el origen de la investigación

También, Hernández, en el Libro Metodología de la Investigación, clasifica los enfoques de investigación del siguiente modo: 1) enfoque cualitativo, 2) enfoque cuantitativo y 3) enfoque mixto (cualitativo y cuantitativo).

Así mismo, Hernández y Baptista mencionan los alcances que puede tener un estudio: exploratorio, descriptivo, correlacional y explicativo. Además,

mencionan que estos no deben considerarse como tipos de investigación, ya que los alcances constituyen un continuo de causalidad que tendría un estudio.

También, se tiene la clasificación según Gordon Dankhe, mencionado por (Cortés Padilla, 2012); el autor menciona cuatro tipos de estudio en las investigaciones sobre el comportamiento humano. Estos son: 1) estudios exploratorios, 2) estudios descriptivos, 3) estudios correlacionales, y 4) estudios explicativos.

Estudios exploratorios

Sirven para preparar el terreno y, por lo común, anteceden a las investigaciones descriptivas, correlacionales o explicativas, por lo general identifican relaciones entre variables.

Los estudios descriptivos

Según (Cortés padilla) buscan especificar las propiedades importantes de personas, grupos, comunidades o cualquier otro fenómeno, que sean sometidos a análisis.

Estudios correlacionales

Según (Cortés padilla) miden el grado de relación entre dos o más variables para luego hacer un análisis estadístico de correlación.

Estudios explicativos

Su propósito es explicar por qué ocurre un fenómeno y las condiciones para que se dé; también, el porqué de la relación entre dos o más variables.

La presente investigación se puede clasificar como aplicada por que trata de resolver un problema. Es experimental (cuasi experimental) porque se aplicó estímulos o tratamientos. Es descriptiva porque se basa en la medición del rendimiento académico. Es correlacional porque mide el grado de relación entre dos variables: el simulador de arquitectura de computadoras y el rendimiento académico.

Diseño de la investigación

En el libro Metodología de la Investigación, se describe el término diseño como sigue:

“El término **diseño** se refiere al plan o estrategia concebida para obtener la información que se desea con el fin de responder al planteamiento del

problema” (Hernández, Fernández, & Baptista, 2014, p. 128). Luego, cada investigador utilizará su diseño para analizar la certeza de las hipótesis formuladas en un contexto particular.

También, Hernández, Fernández, y Baptista clasifican los diseños de investigación en dos tipos: 1) no experimentales y 2) experimentales (que administran estímulos o tratamientos y/o intervenciones). A la vez cada uno de estos dos tipos está dividido en subcategorías.

Para la investigación realizada, el diseño de investigación empleado fue de tipo experimental, específicamente el cuasi experimental. Este consiste en manipular una variable, la variable independiente, para ver sus efectos en la variable dependiente. Se trabajó con dos grupos de estudiantes uno fue el experimental y el otro de control, pero en la formación de los grupos no se usó técnica estadística alguna.

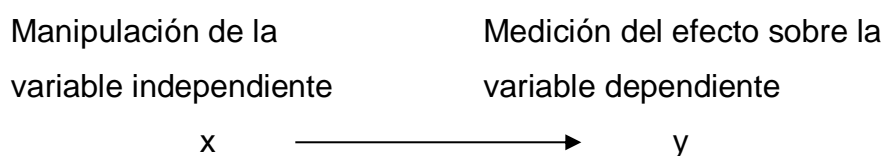


Figura 5. Relación causa efecto

4.2 Método de investigación

Se presentan, primero, algunas definiciones de método. En el diccionario de la lengua española de la Real Academia Española se dan las definiciones siguientes: “modo de decir o hacer con orden”. Una segunda definición es: “modo de obrar o proceder, hábito o costumbre que cada uno tiene y observa”. Una tercera definición dice: “obra que enseña los procedimientos de una ciencia o arte”. Finalmente, una cuarta definición señala: “procedimiento que se sigue en las ciencias para hallar la verdad y enseñarla”.

Existen otras definiciones de método. Método es un procedimiento que se siguen para obtener un fin predeterminado. También es la serie de pasos que se dan para completar una tarea determinada y alcanzar un objetivo determinado.

Ya que en este apartado se presentaron definiciones de método, también se presenta la definición de metodología. La Real Academia Española define metodología como “ciencia del método”; una segunda definición es: “conjunto de

métodos que se siguen en una investigación científica o en una exposición doctrinal”. Entonces una metodología sería colección de teorías, conceptos o ideas.

El método científico

Según (Caballero Romero, 2011), el método científico es una mega orientación o macro estrategia que abstrae una serie de actividades de investigación. El método científico es universal, pero tiene variantes y cada investigador lo adopta al trabajo que desarrollará.

Según (Muñoz, Como elaborar y asesorar una investigación de tesis, 2011), “el método científico tiene carácter general, pero existen diferentes precisiones de acuerdo con la perspectiva de diferentes autores”. Menciona y muestra varios modelos, dentro de ellos el modelo de método científico general, este consta de las siguientes etapas: 1) planteamiento del problema, 2) formulación de la hipótesis, 3) levantamiento de información, 4) análisis e interpretación de datos, 5) comprobación de la hipótesis, y 6) difusión de resultados.

A su vez, el método científico tiene métodos de aplicación específica de acuerdo con lo que se investiga. Según (Muñoz, 2013), algunos métodos específicos son los siguientes: 1) inducción–deducción, 2) analítico-sintético, 3) objetivista-subjetivista, 4) estático –dinámico.

Método inductivo

De acuerdo con (Muñoz, 2013), consiste en la observación de casos de un fenómeno, hecho o evento, para analizarlo y luego formular conclusiones de carácter general que se convertirán en leyes, teorías, postulados y otros.

Método deductivo

Este método infiere de lo general a lo específico (Muñoz, 2013).

El principal método de investigación utilizado en el presente trabajo fue el método científico, porque se realizó siguiendo las reglas de la investigación científica que a su vez utiliza el método científico.

Como método específico se empleó el método inductivo ya se analizó una muestra para generalizar a una población más grande.

4.3 Población y muestra

Población

En la presente investigación, la población lo conformaron los estudiantes de la Facultad de Ingeniería en informática y Sistemas de la Universidad Nacional Agraria de la Selva, matriculados en el semestre 2019-I, siendo en total 261 estudiantes de acuerdo con la información de la Dirección de Coordinación y Desarrollo Académico de la UNAS (Anexo 04).

Muestra

La muestra se obtuvo de los estudiantes matriculados en el semestre 2019-I, en el curso Sistemas operativos I. El estudio se realizó en la asignatura sistemas operativos I, pues el 50% de su contenido trata temas de arquitectura. Asimismo, hay que mencionar en la UNAS no existía la asignatura con el nombre arquitectura de computadoras cuando se desarrolló la presente investigación. La muestra estuvo conformada por 35 estudiantes matriculados seleccionados bajo el criterio de asistencia regular a clases durante el periodo señalado. Se formaron dos grupos, uno de 17 estudiantes para el grupo experimental; el otro grupo, conformado por 18 estudiantes, fue el grupo de control. Los grupos se formaron de acuerdo a como los alumnos se matricularon al inicio del semestre 2019-I. Así mismo, el grupo experimental fue parte de la sección con matrícula (IS030304A) y el grupo de control fue parte de la sección con matrícula (IS030304B).

4.4 Lugar de estudio y periodo de desarrollo

El estudio se desarrolló en el campus de la Universidad Nacional Agraria de la Selva específicamente en la Facultad de Ingeniería en Informática y Sistemas, ubicada en el departamento de Huánuco, provincia de Leoncio Prado, distrito de Rupa Rupa, ciudad de Tingo María. El tiempo durante el cual se desarrolló el estudio fue de abril a julio de 2019.

4.5 Técnicas e instrumentos para la recolección de la información

a) Evaluación de competencias

Para la variable independiente Simulador de Arquitectura de computadoras ARMSim#, se emplearon las siguientes:

Encuesta de opinión

Fue elaborada para conocer la opinión de los estudiantes sobre el simulador de arquitectura de computadoras ARMSim#.

Guía de práctica

Se elaboró y validó una guía de prácticas, para ser desarrollado por los alumnos.

Para la variable dependiente Rendimiento Académico, se aplicó una prueba de evaluación para **pretest** (prueba de entrada) y **postest** (prueba de salida). Consistieron en la evaluación de aspectos conceptuales, procedimentales y actitudinales de los alumnos de la asignatura de Sistemas Operativos I en temas específicos de organización y arquitectura de computadoras. La intención de estas pruebas fue medir la influencia del Simulador de Arquitectura de computadoras ARMSim# en el rendimiento académico en los temas mencionados para luego comparar los resultados de los grupos de investigación.

El instrumento empleado fue una prueba relacionado a temas de arquitectura de computadoras; estos temas están dentro de la asignatura de Sistemas Operativos I, asignatura muy importante según el perfil de egreso de los futuros profesionales en Ingeniería en Informática y Sistemas.

b) Medición de resultados de la prueba

Se evaluó en base al sistema vigesimal empleado en la universidad peruana y según la escala de calificación mostrada en la Tabla 2.

Tabla 2
Escala de calificación en la UNAS

Escala	
Excelente	20, 19 y 18
Muy bueno	17, 16 y 15
Bueno	14 y 13
Regular	12 y 11
Desaprobado	10 o menos

Fuente. Reglamento de estudios de la Universidad Nacional Agraria de la Selva.

Ficha Técnica de la prueba de pretest y postest

Nombre: Prueba de pretest y postest para medir la influencia del **Simulador de Arquitectura de computadoras ARMSim#** en el rendimiento académico en temas de arquitectura de computadoras.

Autor: Eudolio Gregorio Vásquez Pinedo

Año: 2019

Aplicación: individual

Dirigido a: Estudiantes del curso de Sistemas operativos I, de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva

Duración: 40 minutos

Edad promedio: 20 años

Procedencia: Universidad Nacional Agraria de la Selva (UNAS)

N° de Ítems: 20

Pregunta: De alternativa múltiple: a) b) c) d)

Contenido: La prueba consta de diez preguntas de contenido conceptual y diez de contenido procedimental, con un total de veinte preguntas.

Validación y confiabilidad de los instrumentos

El análisis de instrumentos se efectuó de la manera descrita a continuación. Se realizó el análisis de contenido mediante juicio de expertos, estos opinaron sobre los instrumentos indicando si se ajustan o no a lo que se quiere medir.

Validez de contenido

Para validar los instrumentos, estos se sometieron al proceso de validez de contenido por criterio de jueces. Se contó con la participación de dos profesores universitarios especialistas en el tema (ver anexo).

4.6 Análisis y procesamiento de datos

Se aplicó las pruebas de pretest y postest sobre temas de arquitectura de computadoras a los estudiantes de los grupos experimental y de control.

Se registró en hojas de Excel los resultados del pretest y postest de cada grupo incluyendo la puntuación de la parte conceptual y procedimental para cada estudiante.

Se obtuvo la nota total de cada estudiante como la suma de la prueba conceptual y procedimental, tanto para el pretest como en el postest. Para el análisis estadístico se utilizó el software R debido a que es software libre y en el que se puede observar la secuencia de pasos para determinar los estadísticos.

4.6.1 Estadísticos descriptivos y prueba de normalidad

Para comprobar la normalidad de los datos se emplearon las pruebas estadísticas Shapiro-Wilk. Para ello se utilizó el valor de probabilidad (p-valor), con un nivel de significancia ($\alpha = 0.05$).

Si p-valor < 0.05 , los datos no siguen una distribución normal

Si p-valor > 0.05 , los datos siguen una distribución normal

Pretest conceptual del grupo experimental

El p-valor obtenido fue de 0.5027, mayor que 0.05. Entonces los datos siguen una distribución normal.

```
> notas_pretest_concep_gexp<-c(4,3,3,8,5,3,6,7,5,3,5,3,6,6,7,4,1)
> summary(notas_pretest_concep_gexp)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 1.000  3.000  5.000  4.647  6.000  8.000
> sd(notas_pretest_concep_gexp)
[1] 1.868863
> shapiro.test(notas_pretest_concep_gexp)

  Shapiro-Wilk normality test

data:  notas_pretest_concep_gexp
W = 0.95283, p-value = 0.5027
```

Pretest procedimental del grupo experimental

El p-valor obtenido fue de 0.03954, menor que 0.05. Entonces los datos no siguen una distribución normal.

```
> notas_pretest_proced_gexp<-c(2,2,3,3,2,1,1,4,3,2,3,2,4,2,4,2,3)
> summary(notas_pretest_proced_gexp)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 1.000  2.000  2.000  2.529  3.000  4.000
> sd(notas_pretest_proced_gexp)
[1] 0.9432422
> shapiro.test(notas_pretest_proced_gexp)

  Shapiro-Wilk normality test

data:  notas_pretest_proced_gexp
W = 0.88581, p-value = 0.03954
```


Rendimiento académico del grupo experimental en el pretest

El resultado de la prueba Shapiro Wilk arroja un p-valor = 0.08093, mayor que 0.05. Por lo tanto, los datos siguen una distribución normal.

```
> notaspret_total_gexp<-notas_pretest_concep_gexp+notas_pretest_p
roced_gexp
> summary(notaspret_total_gexp)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 4.000  5.000  7.000  7.176  8.000 11.000
> sd(notaspret_total_gexp)
[1] 2.404041
> shapiro.test(notaspret_total_gexp)

      Shapiro-Wilk normality test

data:  notaspret_total_gexp
W = 0.90454, p-value = 0.08093
```

Pretest conceptual del grupo de control

Como el p-valor obtenido fue de 0.5099, mayor que 0.05, entonces los datos siguen una distribución normal.

```
> notas_pretest_concep_gcontrol<-c(7,8,4,3,5,6,7,6,1,8,7,6,5,3,6,6,5,9
)
> summary(notas_pretest_concep_gcontrol)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 1.000  5.000  6.000  5.667  7.000  9.000
> sd(notas_pretest_concep_gcontrol)
[1] 2
> shapiro.test(notas_pretest_concep_gcontrol)

      Shapiro-Wilk normality test

data:  notas_pretest_concep_gcontrol
W = 0.95507, p-value = 0.5099
```

Pretest procedimental grupo de control

Como el p-valor obtenido fue de 0.0321, menor que 0.05, entonces los datos no siguen una distribución normal.

```
> notas_pretest_proced_gcontrol<-c(3,3,3,3,3,4,2,1,3,3,1,3,0,2,2,1,2,2)
> summary(notas_pretest_proced_gcontrol)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000  2.000  2.500  2.278  3.000  4.000
> sd(notas_pretest_proced_gcontrol)
[1] 1.017815
> shapiro.test(notas_pretest_proced_gcontrol)

      Shapiro-Wilk normality test

data:  notas_pretest_proced_gcontrol
W = 0.88532, p-value = 0.0321
```

Rendimiento académico del grupo de control en el pretest

Como el p-valor obtenido fue de 0.3773, mayor que 0.05, entonces los datos siguen una distribución normal.

```
> notaspret_total_gcontrol<-notas_pretest_concep_gcontrol+notas_pre
test_proced_gcontrol
> summary(notaspret_total_gcontrol)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 4.000  7.000  8.000  7.944  9.750 11.000
> sd(notaspret_total_gcontrol)
[1] 2.154946
> shapiro.test(notaspret_total_gcontrol)

      Shapiro-Wilk normality test

data:  notaspret_total_gcontrol
W = 0.94682, p-value = 0.3773
```

Postest conceptual grupo experimental

Como el p-valor obtenido fue de 0.005448, menor que 0.05, entonces los datos no siguen una distribución normal.

```
> notas_postest_concep_gexp<-c(7,7,5,7,6,7,6,8,6,7,7,7,7,6,5,7,5)
> summary(notas_postest_concep_gexp)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 5.000  6.000  7.000  6.471  7.000  8.000
> sd(notas_postest_concep_gexp)
[1] 0.8744746
> shapiro.test(notas_postest_concep_gexp)

Shapiro-Wilk normality test

data:  notas_postest_concep_gexp
W = 0.8303, p-value = 0.005448
```

Postest procedimental grupo experimental

Dado que el p-valor = 0.0307, es menor que 0.05, entonces los datos no siguen una distribución normal.

```
> notas_postest_proced_gexp<-c(10,9,5,8,1,9,7,7,8,7,9,10,3,7,8,5,9)
> summary(notas_postest_proced_gexp)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 1.000  7.000  8.000  7.176  9.000 10.000
> sd(notas_postest_proced_gexp)
[1] 2.455486
> shapiro.test(notas_postest_proced_gexp)

Shapiro-Wilk normality test

data:  notas_postest_proced_gexp
W = 0.87906, p-value = 0.0307
```

Rendimiento académico del grupo experimental en el posttest

Debido a que p-valor fue de 0.1354, mayor que 0.05, entonces los datos siguen una distribución normal.

```
> notaspostest_total_gexp <-  
notas_postest_concep_gexp + notas_postest_proced_gexp  
  
> summary(notaspostest_total_gexp)  
  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
7.00 13.00 14.00 13.65 16.00 17.00  
  
> shapiro.test(notaspostest_total_gexp)  
  
Shapiro-Wilk normality test  
  
data: notaspostest_total_gexp  
  
W = 0.91777, p-value = 0.1354
```

Postest conceptual del grupo de control

Como el p-valor obtenido fue de 0.1347, mayor que 0.05, entonces los datos siguen una distribución normal.

```
> notas_postest_concep_grucontrol <- c(9,4,8,1,7,8,7,6,5,6,9,6,8,5,4,7,6,  
,8)  
  
> summary(notas_postest_concep_grucontrol)  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
1.000 5.250 6.500 6.333 8.000 9.000  
  
> sd(notas_postest_concep_grucontrol)  
[1] 2.029199  
  
> shapiro.test(notas_postest_concep_grucontrol)  
  
Shapiro-Wilk normality test  
  
data: notas_postest_concep_grucontrol
```

Postest procedimental grupo de control

Como el p-valor obtenido fue de 0.5039, mayor que 0.05, entonces los datos siguen una distribución normal.

```
notas_postest_proced_grucontrol<-c(2,7,2,3,3,5,9,4,3,8,6,5,6,4,7,4,6,7)
> summary(notas_postest_proced_grucontrol)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 2.000  3.250  5.000  5.056  6.750  9.000
> sd(notas_postest_proced_grucontrol)
[1] 2.071437
> shapiro.test(notas_postest_proced_grucontrol)

      Shapiro-Wilk normality test

data:  notas_postest_proced_grucontrol
W = 0.95473, p-value = 0.5039
```

Rendimiento académico del grupo de control

El valor de p-valor obtenido fue de 0.2799, mayor que 0.05, entonces los datos presentan una distribución normal.

```
> notaspostest_total_grucontrol<-notas_postest_concep_grucontrol+
notas_postest_proced_grucontrol
> summary(notaspostest_total_grucontrol)
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 4.00  10.00  11.00  11.39  13.75  16.00
> sd(notaspostest_total_grucontrol)
[1] 2.892971
> var(notaspostest_total_grucontrol)
[1] 8.369281
> shapiro.test(notaspostest_total_grucontrol)

      Shapiro-Wilk normality test

data:  notaspostest_total_grucontrol
W = 0.93912, p-value = 0.2799
```

Prueba U de Mann Whitney o Wilcoxon suma de rangos

Para la prueba de hipótesis, se consideró los resultados de las pruebas de normalidad, las cuales señalan que no todos los datos siguen una distribución normal. En los casos en que no todos los datos siguen una distribución normal, no se pueden usar las pruebas paramétricas como las pruebas t student o z. Si las poblaciones fundamentales no son normales, entonces las pruebas t y F tendrán niveles de confianza que difieren de los niveles nominales o niveles prescritos por el experimentador; esto debido a que la prueba t requieren suposición de distribución normal (Devore, 2008).

Entonces, cuando no todos los datos son de distribución normal, se utilizan las pruebas no paramétricas. Así, para distribuciones fundamentales que no siguen una distribución normal existen pruebas que se denominan libres de distribución o no paramétricas (Devore, 2008).

En la presente investigación se utilizó la prueba no paramétrica U de Mann-Whitney, la cual es equivalente a la prueba Wilcoxon suma de rangos que en el programa R está representada por la función Wilcox.

La prueba U de Mann Whitney inicialmente implica el cálculo del estadístico U de cada grupo, este estadístico tiene una distribución conocida bajo la hipótesis nula identificado por Mann and Whitney en 1947 citado por (Nachar, 2008).

El estadístico U se define para cada grupo como sigue:

$$U_x = n_x n_y + \frac{n_x(n_x+1)}{2} - R_x \quad (1)$$

$$U_y = n_x n_y + \frac{n_y(n_y+1)}{2} - R_y \quad (2)$$

Donde:

n_x Es el número de observaciones o participantes del primer grupo

n_y Es el número de observaciones o participantes del segundo grupo

R_x Es la suma de rangos asignado al primer grupo

R_y Es la suma de rangos asignados al segundo grupo.

El estadístico U de la prueba se calcula como el mínimo de U_x y U_y . Se rechaza la hipótesis nula H_0 si el p del $\min(U_x, U_y) < \alpha$.

Si el número de observaciones es mayor que ocho, se puede usar la aproximación normal dada por Mann and Whitney (1947).

$$\mu_U = \frac{n_x n_y}{2} = \frac{(U_x + U_y)}{2}$$

$$\sigma_U = \sqrt{\frac{((n_x n_y)(N + 1))}{12}}$$

Donde:

$$N = n_x + n_y$$

μ_U Corresponde al promedio de la distribución U

σ_U Corresponde a la distribución estándar de U

Si cada grupo incluye más de ocho observaciones, la distribución de la muestra gradualmente se aproxima a una distribución normal, entonces:

$$z = \left(U - \left(\frac{n_x n_y}{2} \right) \right) / \sigma_U$$

Notar que la suma de rangos de ambos grupos es, siempre:

$$R_x + R_y = 1 + 2 + 3 + \dots + N = (N(N + 1))/2$$

Donde:

$$N = (n_x + n_y)$$

$$R_x + R_y = \frac{(n_x + n_y)(n_x + n_y + 1)}{2} = \frac{N(N + 1)}{2} \quad \dots \quad (3)$$

Insertando las ecuaciones (1) y (2) en la ecuación (3) se obtiene:

$$\begin{aligned} R_x + R_y &= n_x n_y + \frac{n_x(n_x + 1)}{2} - U_x + n_x n_y + \frac{n_y(n_y + 1)}{2} - U_y \\ &= \frac{(n_x + n_y)(n_x + n_y + 1)}{2} \end{aligned}$$

Realizando las operaciones algebraicas se obtiene:

$$U_x + U_y = n_x n_y$$

Esta última ecuación señala que la suma de U_x y U_y es igual al producto del número de elementos de las muestras.

Todas las muestras de la presente investigación se procesaron en el programa R. Utilizando las ecuaciones para U_x y U_y , se procesaron los datos en el programa R. A continuación, se muestran los resultados, primero para el pretest.

Prueba U de Mann Whitney en el rendimiento académico del pretest

```
> notaspret_total_gexp<-notas_pretest_concep_gexp+notas_pretest_pr
oced_gexp
> notaspret_total_gcontrol<-notas_pretest_concep_gcontrol+notas_pret
est_proced_gcontrol
> ## U Mann Whitney para el pre test
> muestraTotal <-c(notaspret_total_gexp,notaspret_total_gcontrol)
> rangosMuestra <- rank( muestraTotal )
> muestraTotal <- cbind( muestraTotal, rangosMuestra )
> muestraTotal
  muestraTotal rangosMuestra
[1,]         6         10.5
[2,]         5          6.0
[3,]         6         10.5
[4,]        11         32.5
[5,]         7         15.5
[6,]         4          2.0
[7,]         7         15.5
[8,]        11         32.5
[9,]         8         21.5
[10,]        5          6.0
[11,]        8         21.5
[12,]        5          6.0
[13,]        10         28.0
[14,]        8         21.5
[15,]        11         32.5
[16,]        6         10.5
[17,]        4          2.0
[18,]        10         28.0
[19,]        11         32.5
[20,]        7         15.5
[21,]        6         10.5
[22,]        8         21.5
[23,]        10         28.0
[24,]        9         25.5
[25,]        7         15.5
[26,]        4          2.0
[27,]        11         32.5
[28,]        8         21.5
[29,]        9         25.5
```



```

[30,]      5      6.0
[31,]      5      6.0
[32,]      8     21.5
[33,]      7     15.5
[34,]      7     15.5
[35,]     11     32.5
> sumaRangosx <- sum( rangosMuestra[ 1:17 ] )
> sumaRangosy <- sum( rangosMuestra[ 18:35 ] )
> nx<- length( notaspret_total_gexp )
> ny<-length(notaspret_total_gcontrol)
> Ux<-nx*ny+nx*(nx+1)/2-sumaRangosx
> Uy<-nx*ny+ny*(ny+1)/2-sumaRangosy
> c( Ux, Uy )
[1] 184.5 121.5
> sumaRangosx
[1] 274.5
> sumaRangosy
[1] 355.5
> rangopromediox<-sumaRangosx/nx
> rangopromedioy<-sumaRangosy/ny
> rangopromediox
[1] 16.14706
> rangopromedioy
[1] 19.75
> wilcox.exact(notaspret_total_gexp,notaspret_total_gcontrol,exact = F,
alternative = "t",conf.int = 0.95)

      Asymptotic Wilcoxon rank sum test

data: notaspret_total_gexp and notaspret_total_gcontrol

W = 121.5, p-value = 0.2936

alternative hypothesis: true mu is not equal to 0

95 percent confidence interval:

-2.9999716  0.9999491

```

En la evaluación total del pre test se obtuvieron los siguientes valores: $U = 121.5$ y $p\text{-valor} = 0.2936$, mayor que 0.05 , por lo que se puede afirmar que existe similitud entre los grupos el experimental y el de control.

Prueba U de Mann Whitney en lo conceptual del pretest

Se observa que p-valor = 0.1155, mayor que 0.05, entonces existe similitud en el aspecto conceptual entre los grupos experimental y de control.

```
> wilcox.test(notas_pretest_concep_gexp, notas_pretest_concep_gcontrol)
```

Wilcoxon rank sum test with continuity correction

data: notas_pretest_concep_gexp and notas_pretest_concep_gcontrol

W = 105.5, p-value = 0.1155

alternative hypothesis: true location shift is not equal to 0

Warning message:

In wilcox.test.default(notas_pretest_concep_gexp, notas_pretest_concep_gcontrol) :

cannot compute exact p-value with ties

Prueba U de Mann Whitney en lo procedimental del pretest .

Se observa que el p-valor es de 0.6149, mayor que 0.05, esto indica que entre los grupos hubo similitud en la dimensión procedimental, no hubo diferencia significativa entre los grupos experimental y de control al inicio del experimento.

```
> wilcox.test(notas_pretest_proced_gexp, notas_pretest_proced_gcontrol)
```

Wilcoxon rank sum test with continuity correction

data: notas_pretest_proced_gexp and notas_pretest_proced_gcontrol

W = 168, p-value = 0.6149

alternative hypothesis: true location shift is not equal to 0

Warning message:

In wilcox.test.default(notas_pretest_proced_gexp, notas_pretest_proced_gcontrol) :

cannot compute exact p-value with ties

Prueba U de Mann Whitney en el rendimiento académico del postest

```
> notaspostest_total_gexp<-notas_postest_concep_gexp+notas_p
ostest_proced_gexp
> notaspostest_total_grucontrol<-notas_postest_concep_grucontr
ol+notas_postest_proced_grucontrol
> ##U Mann Whitney para el postest
> muestraTotal <-c(notaspostest_total_gexp,notaspostest_total_g
rucontrol)
> rangosMuestra <- rank( muestraTotal )
> muestraTotal <- cbind( muestraTotal, rangosMuestra )
> muestraTotal
```

	muestraTotal	rangosMuestra
[1,]	17	34.5
[2,]	16	31.5
[3,]	10	7.0
[4,]	15	27.5
[5,]	7	2.0
[6,]	16	31.5
[7,]	13	18.5
[8,]	15	27.5
[9,]	14	23.0
[10,]	14	23.0
[11,]	16	31.5
[12,]	17	34.5
[13,]	10	7.0
[14,]	13	18.5
[15,]	13	18.5
[16,]	12	15.5
[17,]	14	23.0
[18,]	11	12.0
[19,]	11	12.0
[20,]	10	7.0
[21,]	4	1.0
[22,]	10	7.0
[23,]	13	18.5
[24,]	16	31.5
[25,]	10	7.0
[26,]	8	3.0
[27,]	14	23.0
[28,]	15	27.5
[29,]	11	12.0

```

[30,]    14    23.0
[31,]     9     4.0
[32,]    11    12.0
[33,]    11    12.0
[34,]    12    15.5
[35,]    15    27.5
> sumaRangosx <- sum( rangosMuestra[ 1:17 ] )
> sumaRangosy <- sum( rangosMuestra[ 18:35 ] )
> nx<- length(notaspostest_total_gexp)
> ny<-length(notaspostest_total_grucontrol)
> Ux<-nx*ny+nx*(nx+1)/2-sumaRangosx
> Uy<-nx*ny+ny*(ny+1)/2-sumaRangosy
> c( Ux, Uy )
[1] 84.5 221.5
> sumaRangosx
[1] 374.5
> sumaRangosy
[1] 255.5
> rangopromediox<-sumaRangosx/nx
> rangopromedioy<-sumaRangosy/ny
> rangopromediox
[1] 22.02941
> rangopromedioy
[1] 14.19444

```

```

> wilcox.test(notaspostest_total_gexp,notaspostest_total_grucontrol
)

```

Wilcoxon rank sum test with continuity correction

data: notaspostest_total_gexp and notaspostest_total_grucontrol
W = 221.5, p-value = 0.02389
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(notaspostest_total_gexp, notaspostest_total_grucontrol) :
cannot compute exact p-value with ties

La prueba revela la existencia de diferencias entre los grupos experimental y control, debido a que $p= 0.02389$ es menor a 0.05.

Prueba U de Mann Whitney en lo conceptual del postest

La prueba U revela que entre los grupos experimental y de control no hay diferencia significativa en la dimensión cognitiva, debido a que p-valor = 0.8918 es mayor que 0.05.

```
> wilcox.test(notas_postest_concep_gexp, notas_postest_concep_grucontrol)

Wilcoxon rank sum test with continuity correction

data: notas_postest_concep_gexp and notas_postest_concep_grucontrol
W = 148.5, p-value = 0.8918
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(notas_postest_concep_gexp, notas_postest_concep_grucontrol) :
cannot compute exact p-value with ties
```

Prueba U de Mann Whitney en lo procedimental del postest

El valor de $p = 0.006687$ es menor a 0.05, esto indica que existe diferencia entre los grupos el experimental y el de control.

```
> wilcox.test(notas_postest_proced_gexp, notas_postest_proced_grucontrol)

Wilcoxon rank sum test with continuity correction

data: notas_postest_proced_gexp and notas_postest_proced_grucontrol
W = 235, p-value = 0.006687
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(notas_postest_proced_gexp, notas_postest_proced_grucontrol) :
cannot compute exact p-value with ties
```

V. RESULTADOS

Para la contrastación de hipótesis, se utilizó la estadística descriptiva e inferencial.

5.1 Resultados descriptivos

Son los resultados de la prueba posttest. Se presentan los resultados de las pruebas conceptuales y procedimentales. Los datos fueron procesados en el programa estadístico R.

5.1.1 Resultados sobre el rendimiento académico

Resultados de la parte conceptual del posttest

De acuerdo a la Tabla 3, se observa que el grupo experimental posee una media de 6.470588, y el grupo de control tiene una media de 6.333. La media del grupo experimental es mayor que la del grupo de control, pero es una diferencia mínima.

Tabla 3

Estadísticos descriptivos de la dimensión cognitiva del posttest

Grupo	Estadístico	Valor
Experimental N = 17	Media	6.470588
	Mediana	7
	Varianza	0.7647059
	Desviación estándar	0.8744746
	Mínimo	5
	Máximo	8
De control N = 18	Media	6.333
	Mediana	6.500
	Varianza	4.117647
	Desviación estándar	2.029199
	Mínimo	1.000
	Máximo	9.000

Fuente: Elaboración propia, procesado de datos en R

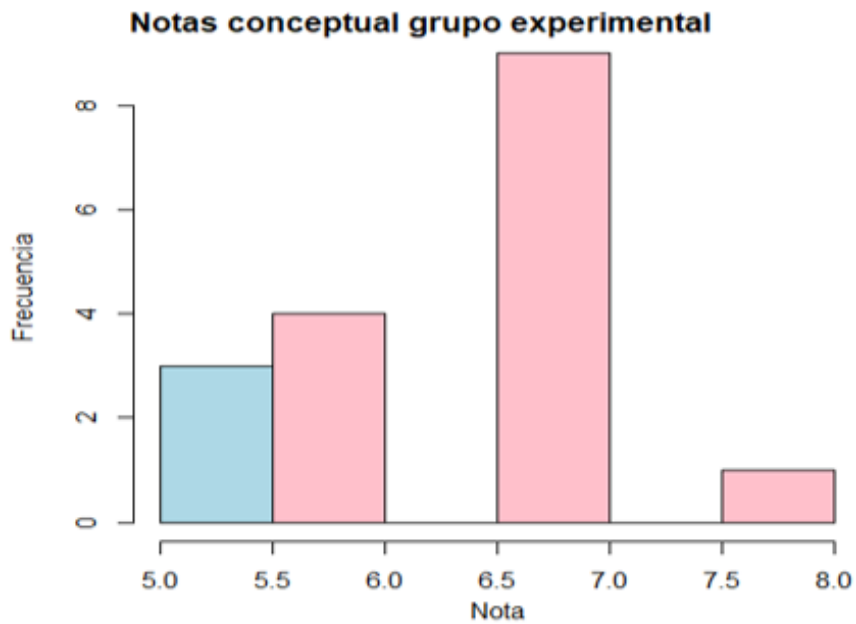


Gráfico 1 Histograma de la evaluación conceptual, grupo experimental, postest

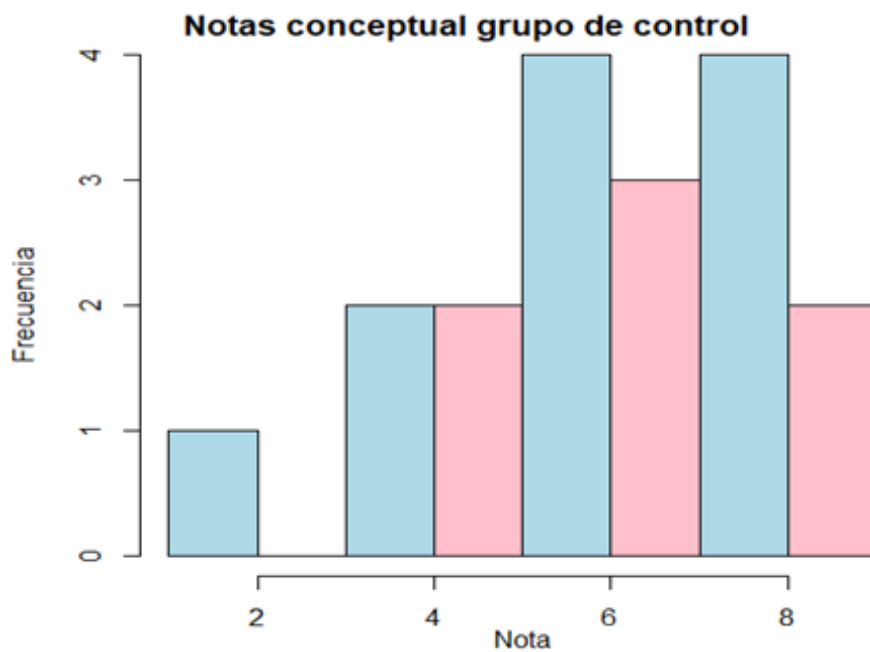


Gráfico 2 Histograma de la evaluación conceptual, grupo de control en postest

Resultados de la parte procedimental del postest

Según la información de la Tabla 4, se observa que el grupo experimental posee una media de 7.176471 y el grupo de control tiene una media de 5.056. La media del grupo experimental es mayor que la del grupo de control.

Tabla 4

Estadísticos descriptivos de la dimensión procedimental del postest

Grupo	Estadístico	Valor
Experimental N = 17	Media	7.176471
	Mediana	8
	varianza	6.029412
	Desviación estándar	2.455486
	Mínimo	1
	Máximo	10
De control N = 18	Media	5.056
	Mediana	5.000
	Varianza	4.29085
	Desviación estándar	2.071437
	Mínimo	2
	Máximo	9

*Fuente: Elaboración propia, procesado de datos en R***Resultados del rendimiento académico promedio postest**

Según los resultados mostrados en la Tabla 5, se observa que el grupo experimental posee una media de 13.65 y el grupo de control de 11.39. La media del grupo experimental es mayor que la del grupo de control.

Tabla 5

Estadísticos descriptivos del rendimiento académico en el postest

Grupo	Estadístico	Valor
Experimental N = 17	Media	13.65
	Mediana	14
	varianza	7.367647
	Desviación estándar	2.714341
	Mínimo	7
	Máximo	17
De control N = 18	Media	11.39
	Mediana	11.00
	Varianza	8.369281
	Desviación estándar	2.892971
	Mínimo	4
	Máximo	16

Fuente: Elaboración propia, procesado de datos en R

Descriptivos del rendimiento académico posttest

De la Tabla 6 y del Gráfico 3 se observa que siete estudiantes (41.1%) del grupo experimental obtienen el calificación muy bueno. Así mismo, de la Tabla 6 y del Gráfico 3, se observa que seis estudiantes (33.334%) del grupo de control obtienen el calificación de desaprobado, esto en cuanto al rendimiento académico en el posttest.

Tabla 6
Estadísticos descriptivos del rendimiento académico en el posttest, según escala de la UNAS

Grupo	Calificativos	Frecuencia	porcentaje
Experimental N = 17	Excelente [20-18]	0	0.0
	Muy bueno [17-15]	7	41.17647
	Bueno [14-13]	6	35.29412
	Regular [12-11]	1	5.882353
	Desaprobado [10-0]	3	17.64706
	Sub total	17	100
De control N = 18	Excelente [20-18]	0	0.0
	Muy bueno [17-15]	3	16.667
	Bueno [14-13]	3	16.667
	Regular [12-11]	6	33.334
	Desaprobado [10-0]	6	33.334
	Sub total	18	100

Fuente: Elaboración propia

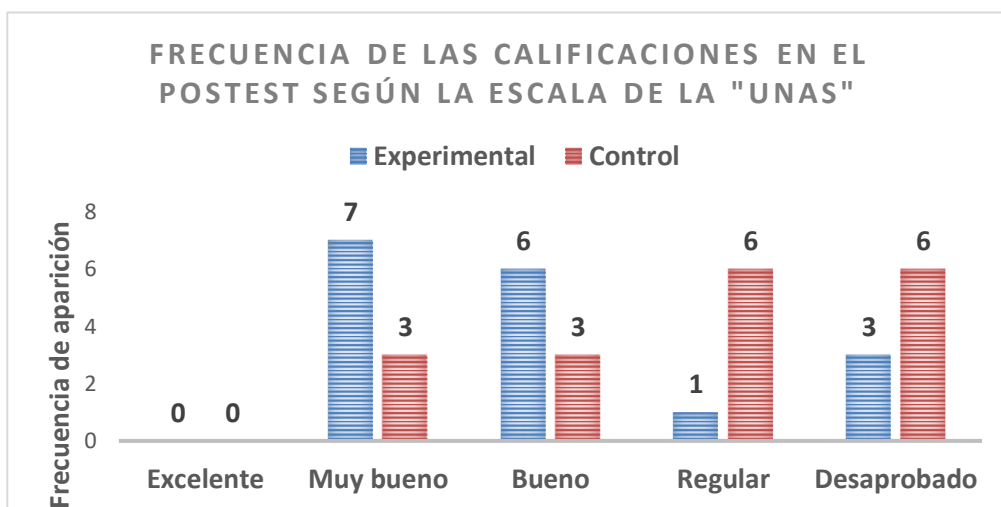


Gráfico 3 Calificaciones en el posttest en la escala de la UNAS

Resultados de pretest y postest

La Tabla 7 indica que, en pretest, el grupo experimental obtuvo una media de 4.647 en lo conceptual, mientras que el grupo de control de 5.667. Así, la media del grupo de control fue ligeramente mayor.

Tabla 7
Estadísticos descriptivos de las dimensiones conceptual y procedimental del pretest

Grupo	Dimensión	Estadístico	Valor
Experimental N = 17	Pre-conceptual	Media	4.647
		Mediana	5.000
		varianza	3.492647
		Desviación estándar	1.868863
		Mínimo	1
		Máximo	8
De control N = 18	Pre-conceptual	Media	5.667
		Mediana	6.000
		Varianza	4
		Desviación estándar	2
		Mínimo	1
		Máximo	9
Experimental N = 17	Pre procedimental	Media	2.529
		Mediana	2.000
		varianza	0.8897059
		Desviación estándar	0.9432422
		Mínimo	1
		Máximo	4
De control N = 18	Pre procedimental	Media	2.278
		Mediana	2.500
		Varianza	1.035948
		Desviación estándar	1.017815
		Mínimo	0
		Máximo	4

Fuente: Elaboración propia, procesado de datos en R

También, el grupo experimental obtuvo una media de 2.529 en lo procedimental y el grupo de control obtuvo una media de 2.278. La media del

grupo experimental es ligeramente mayor a la del grupo de control. En general, no existen diferencias estadísticas significativas entre los grupos experimental y el de control en el pretest.

La tabla 8, muestra los resultados del postest. Se observa que, para el caso de la variable conceptual, no existen diferencias estadísticas significativa. Mientras que, para el caso de la variable procedimental, si existen diferencias estadísticas significativas.

Tabla 8
Estadísticos descriptivos de las dimensiones conceptual y procedimental del postest

Grupo	Dimensión	Estadístico	Valor
Experimental N = 17	Post conceptual	Media	6.470588
		Mediana	7
		varianza	0.7647059
		Desviación estándar	0.8744746
		Mínimo	5
		Máximo	8
De control N = 18	Post conceptual	Media	6.333
		Mediana	6.500
		Varianza	4.117647
		Desviación estándar	2.029199
		Mínimo	1.000
		Máximo	9.000
Experimental N = 17	Post procedimental	Media	7.176471
		Mediana	8
		varianza	6.029412
		Desviación estándar	2.455486
		Mínimo	1
		Máximo	10
De control N = 18	Post procedimental	Media	5.056
		Mediana	5.000
		Varianza	4.29085
		Desviación estándar	2.071437
		Mínimo	2
		Máximo	9

Fuente: Elaboración propia, procesado de datos en R

5.2 Resultados inferenciales

Resultados de la prueba de normalidad de los datos

En la sección 4.6, análisis y procesamiento de datos, se realizaron las pruebas de normalidad de los datos que se muestran en las tablas siguientes.

Tabla 9

Estadísticos inferenciales, prueba de normalidad del pretest y postest del rendimiento académico

Grupo		Shapiro Wilk	
		Estadístico	p-valor
Experimental N = 17	Pre test	0.90454	0.08093
	Post test	0.91777	0.1354
De control N = 18	Pre test	0.94682	0.3773
	Post test	0.93912	0.2799

Fuente: Elaboración propia, procesado de datos en R

Según la prueba de normalidad Shapiro Wilk, en el pretest del grupo experimental se obtuvo un p-valor de 0.08093, que es mayor que 0.05, lo que indica que los datos pertenecen a una población que sigue una distribución normal. En el postest, también para el grupo experimental, se obtuvo un p-valor de 0.1354, que es mayor a 0.05, lo que indica que los datos corresponden a una población con distribución normal.

En lo que respecta al grupo de control, la prueba de normalidad Shapiro Wilk en el pretest obtuvo un p-valor de 0.3773, que es mayor que 0.05, entonces los datos pertenecen a una población que sigue una distribución normal. En el postest se obtuvo un p valor de 0.2799, que es mayor que 0.05, entonces los datos corresponden a una población con distribución normal.

Tabla 10

Estadísticos inferenciales, prueba de normalidad del pretest y postest de las dimensiones conceptual y procedimental

Grupo		Shapiro Wilk	
		Estadístico (w)	p-valor
Experimental N = 17	Pre conceptual	0.95283	0.5027
	Post conceptual	0.8303	0.005448
	Pre procedimental	0.88581	0.03954
	Post procedimental	0.87906	0.0307
De control N = 18	Pre conceptual	0.95507	0.5099
	Post conceptual	0.921	0.1347
	Pre procedimental	0.88532	0.0321
	Post procedimental	0.95473	0.5039

Fuente: Elaboración propia, procesado de datos en R

Para el grupo experimental, en la dimensión preconceptual, el p-valor es 0.5027, que es mayor a 0.05, entonces los datos siguen una distribución normal. En el postest, para la dimensión conceptual, el p-valor es 0.005448, que es menor que 0.05, esto indica que los datos no siguen una distribución normal.

Para el grupo experimental, en el pretest, la dimensión procedimental p-valor es 0.03954, que es menor que 0.05, esto indica que los datos no siguen una distribución normal. En el post test, en la dimensión procedimental, p-valor es 0.0307, que es menor que 0.05, lo que indica que los datos no siguen una distribución normal.

Para el grupo de control, en la dimensión preconceptual p valor es 0.5099 que es mayor a 0.05, entonces los datos siguen una distribución normal. En el postest para la dimensión conceptual p valor es 0.1347 que es mayor que 0.05, esto indica que los datos siguen una distribución normal.

Para el grupo de control, en el pretest, la dimensión procedimental p-valor es 0.0321, que es menor que 0.05, esto indica que los datos no siguen una distribución normal. En el post test, de la dimensión procedimental, el p-valor es 0.5039, que es mayor que 0.05, lo que indica que los datos siguen una distribución normal.

Tabla 11

Resultados prueba U de Mann Whitney, del pretest y postest del rendimiento académico

PRETEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	274.5	16.15	U =121.5
Control	18	355.5	19.75	p = 0.294 z = -1.050
POSTEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	374.5	22.03	U = 84.5
Control	18	255.5	14.19	p = 0.023 z = -2.276

Fuente: Elaboración propia, procesado de datos en R

En el pretest la prueba revela que no existe diferencia entre los grupos experimental y control debido a que ($p= 0.294 < 0.05$).

En el postest la prueba indica que si existe diferencia significativa entre los grupos el experimental y el de control debido a que ($p = 0.023 < 0.05$).

Existe diferencia en el rendimiento académico entre el grupo experimental y el grupo de control luego del tratamiento al grupo experimental.

Resultados de la dimensión conceptual

Tabla12

Resultados de la prueba U de Mann Whitney de la dimensión conceptual en el pre test y en el postest

PRETEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	258.5	15.21	U = 105.5
Control	18	371.5	20.64	p = 0.112 z = -1.591
POSTEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	301.500	17.74	U = 148.5
Control	18	328.500	18.25	p = 0.878 z = -0.153

Fuente: Elaboración propia, procesado de datos en R

Resultados de la dimensión procedimental

Tabla13

Resultados de la prueba U de Mann Whitney de la dimensión procedimental en el pre test y en el postest

PRETEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	321.000	18.88	U = 138.0
Control	18	309.000	17.17	p = 0.603 z = -0.52
POSTEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	388.000	22.82	U = 71.0
Control	18	242.000	13.44	p = 0.006 z = -2.729

Fuente: Elaboración propia, procesado de datos en R

VI. DISCUSIÓN DE RESULTADOS

6.1 Contrastación y demostración de la hipótesis con los resultados

Para la contrastación y demostración de la hipótesis, se hicieron las consideraciones siguientes:

- Se planteó la hipótesis nula (H_0) y la hipótesis alterna (H_1).
- Se consideró un nivel de significación $\alpha = 0.05$.
- Se consideró como regla de decisión, para la hipótesis general y para las hipótesis específicas, rechazar la hipótesis nula si $p\text{-valor} < 0.05$ y aceptar la hipótesis alterna.
- Se consideró la prueba U de Mann-Whitney.

Hipótesis general

H₀: El Simulador de Arquitectura de computadoras ARMSim# no influye significativamente en el rendimiento académico en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

H₁: El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en el rendimiento académico en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

En el postest, la prueba U indica que los grupos de control y experimental son diferentes ($p = 0.023 < 0.05$) y con ventaja del grupo experimental, este último con rango promedio de 22.03 por encima del grupo de control el cual obtuvo un rango promedio de 14.19.

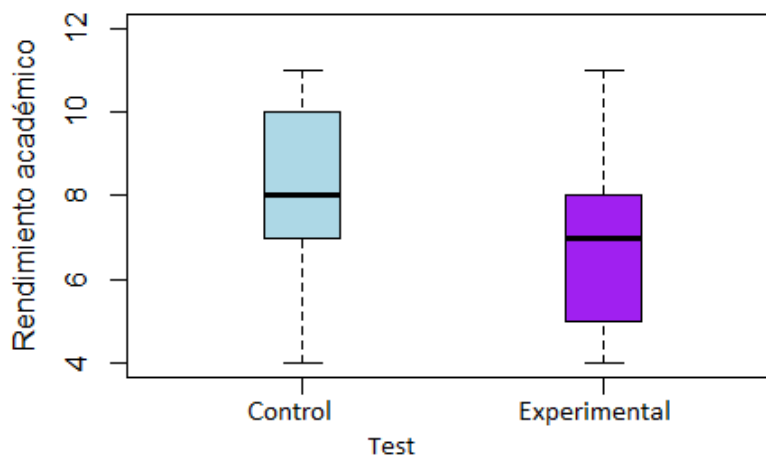
Por lo tanto, se acepta la hipótesis de que el Simulador de Arquitectura de computadoras ARMSim# influye significativamente en el rendimiento académico en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

Tabla 6.1 Estadísticos inferenciales del rendimiento académico del postest

POSTEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	374.5	22.03	U = 84.5
Control	18	255.5	14.19	p = 0.023
				z = -2.276

Fuente: Elaboración propia, procesado de datos en R

(a) Pretest



(b) Postest

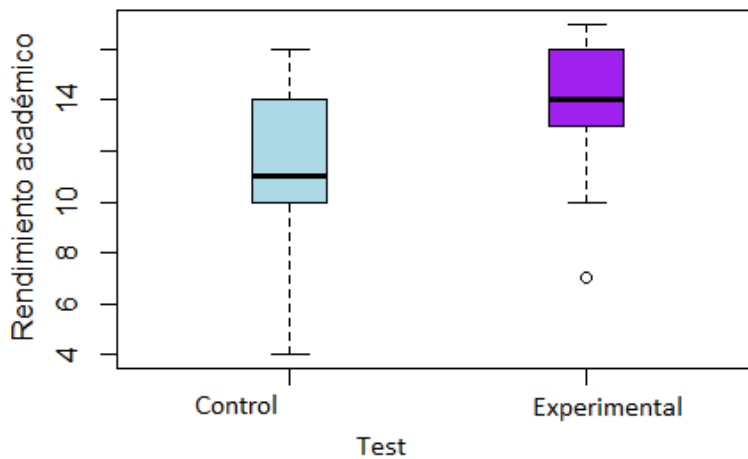


Gráfico 4 Diagrama de cajas del Rendimiento académico de los grupos de control y experimental, (a) pretest y (b) postest

En el Grafico 3, se muestran comparativamente el pretest y el postest. En el pretest, la mediana del grupo control y la del grupo experimental (líneas remarcadas que cruzan las cajas) se ubicaron casi a la misma altura. En segundo lugar, en el postest, la mediana del grupo experimental estaba muy por encima de la mediana del grupo de control. Ambos casos reflejan los resultados de las pruebas U aplicadas y comentadas previamente.

Como apunte final, en el pretest el grupo de control presentó igual dispersión de rendimientos académicos que el grupo experimental. En cambio, en el postest hubo mayor dispersión en el grupo de control (caja más extendida) que en el experimental. También, se detectó un valor anómalo en el grupo experimental (reg. 40, RA=7).

Hipótesis específica 1

H0: El Simulador de Arquitectura de computadoras ARMSim# no influye significativamente en los resultados conceptuales en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

H1: El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados conceptuales en temas de arquitectura de computadoras de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

Tabla 6.2 Estadísticos inferenciales del postest de lo conceptual

POSTEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	301.500	17.74	U = 148.5
Control	18	328.500	18.25	p = 0.878
				z = -0.153

Fuente: Elaboración propia, procesado de datos en R

Según los resultados de la prueba U de Mann-Whitney y la suma de rangos de Wilcoxon, el p-valor es 0.878, que es mayor que 0.05; entonces, de acuerdo con las consideraciones para la prueba de hipótesis, no hubo diferencia significativa entre los grupos el experimental y de control.

Por lo tanto, en este estudio se encontró que el software simulador no influye en los resultados conceptuales de los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

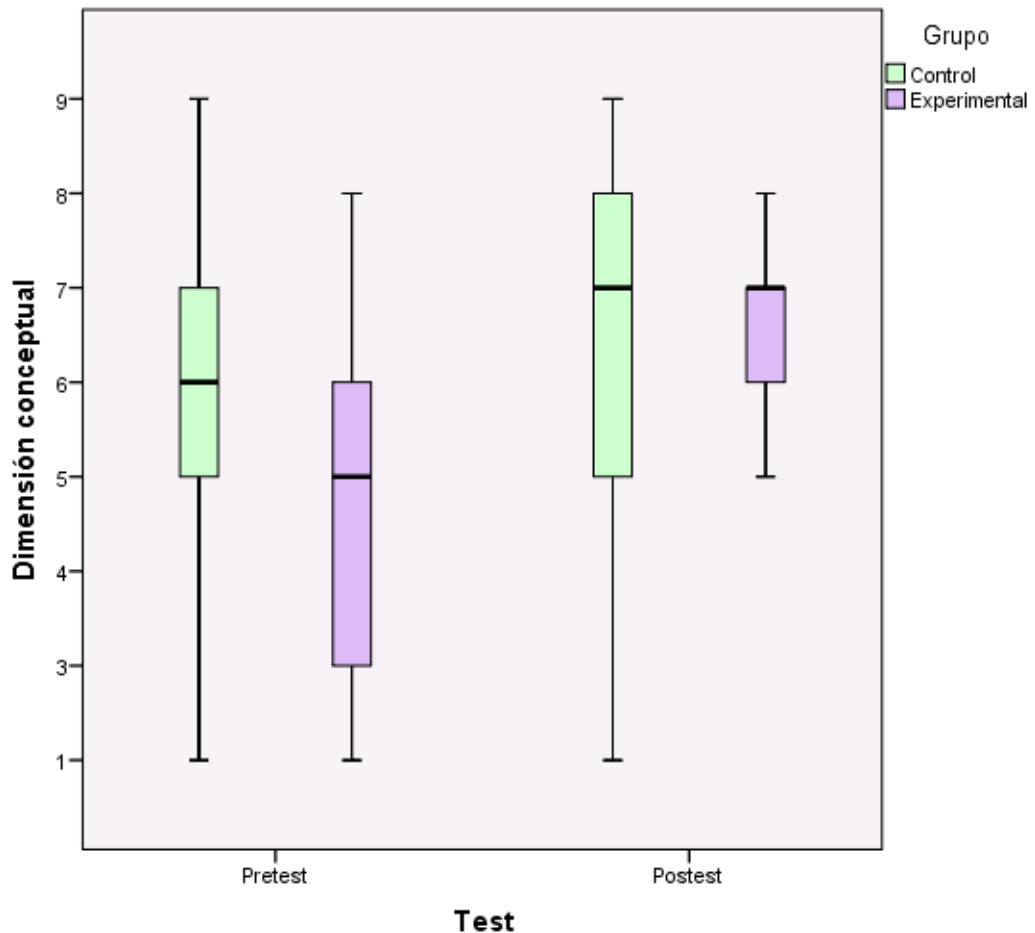


Gráfico 5 Diagrama de cajas de los resultados conceptuales pretest y postest

En el Gráfico 4, se muestran comparativamente el pretest y el postest. En el pretest, la mediana del grupo de control y la del grupo experimental (líneas remarcadas que cruzan las cajas) se ubicaron casi a la misma altura. En el postest, las medianas de ambos grupos tienen la misma altura. Ambos casos reflejan los resultados de las pruebas U comentadas previamente.

Como apunte final, en el pretest, el grupo de control presentó igual dispersión de resultados conceptuales que en el caso del grupo experimental. En cambio, en el posttest hubo mayor dispersión en el grupo de control (caja más extendida) que en el experimental.

Hipótesis específica 2

H0: El Simulador de Arquitectura de computadoras ARMSim# no influye significativamente en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

H1: El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

Tabla 6.3 Estadísticos inferenciales del posttest en lo procedimental

POSTEST				
Grupo	N	Suma de rangos	Rango promedio	U de Mann-Whitney
Experimental	17	388.000	22.82	U = 71.0
Control	18	242.000	13.44	p = 0.006 z = -2.729

Fuente: Elaboración propia, procesado de datos en

La prueba U de Mann Whitney indica que los grupos de control y experimental fueron diferentes ($p=0.006<0.05$) y con ventaja del grupo experimental pues su rango promedio, 22.82, es mayor al del grupo experimental que obtuvo un rango promedio de 13.44.

Por lo tanto, se acepta la hipótesis que el simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva.

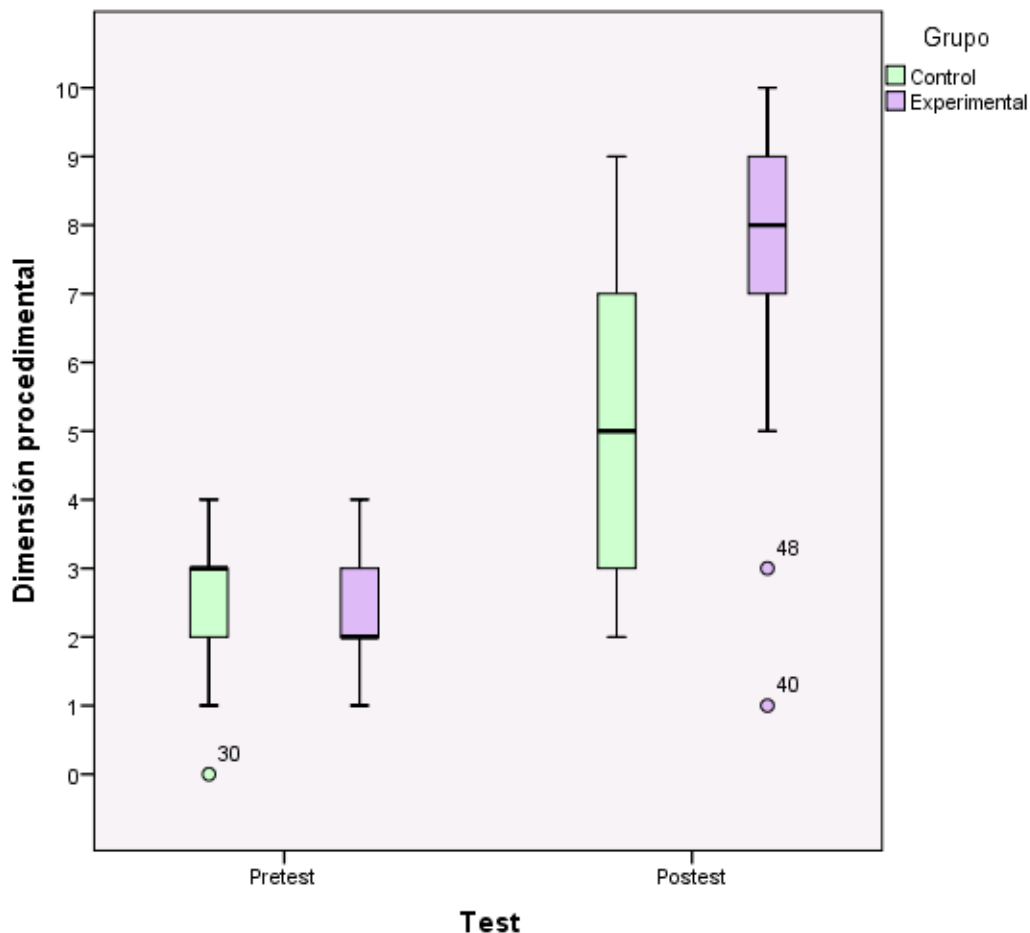


Gráfico 6 Diagrama de cajas de la dimensión contenido procedimental en pretest y posttest

En el Gráfico 5, se ilustran los dos resultados Procedimentales (pretest y posttest). En el pretest existe proximidad de medianas (en concordancia con el resultado del test U, visto anteriormente); mientras que, en el posttest, las medianas no fueron iguales (concuerta con el test U correspondiente) y se aprecia que la mediana del grupo experimental fue mayor que la del grupo de control.

Por otro lado, en el pretest, los grupos presentaron similitud en cuanto a dispersión de puntajes, con un único valor anómalo en el grupo de control (reg. 30, RP=0). En cambio, en el posttest, los resultados procedimentales del grupo de control fueron más dispersos que los del grupo experimental, en este último se tuvieron dos resultados anómalos (reg. 40, RP=1 y reg.48, RP=3).

Hay que mencionar que la dimensión actitudinal es importante en el modelo por competencias. Ella está ligada a aspectos psicológicos, pero no son parte de la presente investigación y, por ello, no se consideró dicha dimensión.

6.2 Contrastación de resultado con otros estudios similares

Para la contrastación de resultados se tomaron investigaciones, tanto del ámbito internacional, como del ámbito nacional. A continuación, se detallan.

En la investigación realizado por (Prasad, Alsadoon, Beg, & Chan, 2015), profesores e investigadores en las especialidades de Computación y Matemática y Tecnología de Información, en universidades y centros de investigación de Australia, utilizaron el simulador MARIE, acrónimo de Machine Architecture That is Really Intuitive Easy, y que consistió de tres laboratorios, en ellos los estudiantes tenían que escribir código en assembler para ejecutar diferentes ciclos de lectura, decodificación y ejecución de instrucción, y entender cómo trabaja por dentro de una máquina.

Al final de las sesiones de laboratorios, cien estudiantes fueron encuestados para dar su opinión del simulador MARIE. Casi el 95% de los estudiantes estuvieron de satisfechos con el entorno grafico del simulador para ejecutar programas en assembler y visualizar como se mueven los datos entre la memoria los registros del CPU.

Todos los estudiantes estuvieron de acuerdo en que el simulador y los archivos fueron de mucha ayuda para aprender a escribir código en assembler. El 94% de estudiantes mencionaron estar de acuerdo con la funcionalidad del simulador para entender la operación CPU/registros.

Luego de seleccionar el simulador MARIE, se comparó los resultados de las evaluaciones en las asignaturas Sistemas de computadoras e introducción a los sistemas de información para dos semestres diferentes del verano 2013 y el otoño de 2014; en el semestre de verano no se utilizó ningún simulador, en el semestre de otoño se utilizó el simulador MARIE. El profesor para ambos semestres fue el mismo; también, el contenido del curso y los materiales fueron los mismos, la diferencia fueron las tareas asignadas.

Como resultados finales, presentaron que, para los estudiantes más destacados, el utilizar ejercicios de laboratorio no les beneficia mucho; hubo tres estudiantes en ambos semestres con calificaciones de alta distinción, sus puntajes fueron entre 85 y 100. Las calificaciones promedio de los estudiantes para ambos semestres fueron similares estando estas entre 65 y 74. El utilizar laboratorios, incrementó en 23% los estudiantes con puntaje entre 50 a 60, comparado a 4% de los que no utilizaron laboratorios.

Sin embargo, los investigadores concluyen que el utilizar ejercicios de laboratorio no ayuda significativamente aprender mejor los conceptos de arquitectura de computadoras y que el beneficio fue más significativo en aquellos estudiantes que se habrían esforzado sin el uso de laboratorios para entender conceptos de arquitectura de computadoras.

También se encontró el uso del simulador denominado EduMIPS64 para la enseñanza de temas de organización y arquitectura de computadoras. Los investigadores evalúan el beneficio de utilizarlo realizando una encuesta aplicada a los que finalizaron el curso, comprueban lo apropiado del simulador y las fortalezas y debilidades de este (Patti, Spadaccini, Palesi, Fazzino, & Catania, 2012).

No se encontraron investigaciones relacionadas al uso de simuladores en la enseñanza de organización y arquitectura de computadoras que se hayan realizado en Perú. Solo se encontraron investigaciones relacionadas con el uso de programas informáticos en la enseñanza de asignaturas como matemática, física y circuitos eléctricos.

En la tesis de maestría de Francisco Sinchi Yupanqui, el experimento consistió en un único grupo de estudiantes, se les aplicó una prueba inicial, o pretest, sin que estos hayan utilizado ningún simulador. Posterior a esto los estudiantes utilizaron un simulador para las prácticas de laboratorio durante varias sesiones, después de esto los estudiantes son nuevamente evaluados mediante una prueba final o postest. Luego del tratamiento estadístico, los resultados son que el simulador influye significativamente en el aprendizaje de los estudiantes que fueron parte de dicho experimento. Los resultados de la dimensión cognitiva fueron que el simulador o programa no ayuda en el

aprendizaje cognitivo de los estudiantes que fueron parte de dicho experimento. Los resultados de la dimensión procedimental fueron que el simulador si influye en el aprendizaje procedimental.

6.3 Responsabilidad ética de acuerdo con los reglamentos vigentes

La presente investigación se desarrolló teniendo en cuenta principios de ética y valores en general. Así mismo el autor declara conocer el código de ética de investigación de la Universidad Nacional del Callao, y se ha actuado de acuerdo con el mismo. También los comentarios, opiniones escritos en la presente investigación son responsabilidad del autor.

CONCLUSIONES

En este apartado, según los objetivos de la investigación, se presentan las conclusiones del trabajo realizado.

- a) Se determinó que el simulador de arquitectura de computadoras ARMSim# influye en el rendimiento académico en temas de arquitectura de computadoras de los estudiantes de la escuela profesional de ingeniería en informática y sistemas de la Universidad Nacional Agraria de la Selva, en la muestra que se investigó en el semestre académico 2019-1.
- b) Según los resultados inferenciales de la prueba Wilcoxon, se encontró que no existe diferencia significativa entre el grupo que utilizó el simulador y el grupo que no lo usó. En la dimensión conceptual, con p-valor de 0.878 (mayor que 0.05), la aplicación del programa simulador no influye en los resultados conceptuales de los estudiantes de ingeniería en informática y sistemas de la FIIS-UNAS.
- c) Se determinó que el simulador de Arquitectura de computadoras ARMSim# influye en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes de la escuela profesional Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva. Los resultados inferenciales de prueba de hipótesis de Wilcoxon dieron un p-valor igual a 0.006 que es menor que 0.05. Por tanto, la aplicación del programa simulador influye en los resultados procedimentales de los estudiantes de ingeniería en informática y sistemas de la FIIS-UNAS.

RECOMENDACIONES

- a) Al existir escasas o nulas investigaciones en Perú sobre el uso de simuladores en temas de organización y arquitectura de computadoras, sería recomendable que se promueva nuevas investigaciones en estas áreas, de tal manera se tenga mayores evidencias acerca de su impacto positivo o no.
- b) Sería recomendable el uso de simuladores como apoyo a la enseñanza de temas de organización y arquitectura de computadoras, pues relaciona la teoría con la práctica; sin embargo, debe considerarse también que está sea acompañada de técnicas de enseñanza que favorezcan el aprendizaje por parte de los estudiantes.
- c) Sería recomendable que los profesores tengan entrenamiento y habilidades en la utilización de los simuladores para un mejor performance en la enseñanza.

REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

- ACM/IEEE-CS. (2013). *Computer Science Curricula 2013 - ACM Digital Library - Association ...*. Nueva York: ACM/IEEE.
- Aguilar Juárez, I., & Heredia Alonso, J. R. (2013). Simuladores y laboratorios virtuales para Ingeniería en Computación. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 1-19.
- Alasdoon, A., Prasad, P., & Chan, A. (2015). Simulators for Teaching Computer Organization and Architecture – A Brief Survey. *4th International Conference on Education and Information Management 2013*. Penang: IFRD.
- Association for Computing Machinery. (Diciembre de 2013). *cs2013_web_final.pdf*. Recuperado el 17 de Diciembre de 2019, de https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf
- Association for Computing Machinery. (2019). *Education*. Recuperado el 12 de 04 de 2019, de Curricula recomendatio: <https://www.acm.org/education/curricula-recommendations>
- Atanasovski, B., Ristov, S., Gusev, M., & Anchev, N. (2013). EDUCache simulator for teaching computer architecture and organization. *IEEE Xplore*.
- Barrachina Mir, S., Fabregat Llueca, G., Fernández Fernández, J. C., & León Navarro, G. (2015). ARMSim y QtARMSim: simulador de ARM para docencia. *XXI Jornadas de la Enseñanza Universitaria de la Informática* (págs. 1-8). Universitat Jaume I.
- Caballero Romero, A. E. (2011). *Metodología integral innovadora para planes y tesis*. Lima: Instituto Metodológico Alen Caro E.I.R.L.
- Carlos, M. (2011). *Como elaborar y asesorar una investigación de tesis*. México: Prentice Hall.
- Ceballos Villacrés, Y. E. (2010). *Incidencia del uso del simulador de redes de computadoras packet tracer de cisco systems, en el desarrollo de competencias procedimentales, en la asignatura redes de computadoras II, del cuarto año de la escuela de sistemas, de la facultad de ingeniería,*. Ambato: Universidad Técnica de Ambato.
- Cecile Yehezkel, William Yurcik, Murray Pearson, Dean Armstrong. (2001). Three Simulator Tools For Teaching Computer Architecture:. *ACM Digital Library*, 1-4.
- Cecile Yehezkel, William Yurcik, Murray Pearson, Dean Armstrong. (2001). Three Simulator Tools for Teaching Computer Architecture: EasyCPU, Little Man

Computer, and RTLsim. *ACM Journal of Educational Resources in Computing*, 60-80.

Cortés Padilla, M. T. (2012). *Metodología de la investigación*. México: Trillas.

De la Cruz Ganoa, E. (2016). *Software Geogebra y su influencia en el aprendizaje de las funciones reales en los estudiantes del primer ciclo de la Facultad de Ciencias Contables de la Universidad Nacional del Callao*. Lima: Univ. Nac. del Callao.

Garay, G., Tchernykh, A., Yu, A., Garichev, S., Nesmachnow, S., & Torres, M. (04 de Abril de 2017). *Using Simulators for Teaching Computer Organization and Architecture*. Recuperado el 12 de Noviembre de 2019, de <http://dx.doi.org/doi:10.1016/j.jocs.2017.04.004>

Guizado Carmona, Federico Edgard. (2015). *PROPUESTA DIDÁCTICA DE IMPLEMENTACIÓN*. Lima: Universidad San Ignacio de Loyola.

Hennesy, J., & Patterson, D. (2012). *Computer Architecture*. New york: Elsevier.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. D. (2014). *Metodología de la investigación*. México D.F.: Mc Graw-Hill/Interamericana editores.

Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*. México D.F.: McGraw Hill. Obtenido de <https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>

Huamán Monroy, G. (2006). *Influencia del método experimental didáctico y refuerzo del aprendizaje asistido por computadora en el rendimiento académico de física de los estudiantes de educación de la UNA-Puno, 2006*. Puno: Universidad Nacional Mayor de San Marcos.

Jay, D. (2008). *Probabilidad y estadística para ingeniería y ciencias*. México: Cengage Learning Editores, S.A. de C.V.

Jorge Fernando Maxnuck Soares, Luís Tadeu M. Raunheite, Takato Kurihara. (2016). The use of MARIE CPU Simulator in Computer Architecture Course: A Case Study of student's perception of learning and performance. 7, 7-13.

Kurniawan, W., & Hanafi, M. (2017). Teaching and learning support for computer architecture and organization courses design in Computer Engineering and Computer Science for undergraduate: A review. *IEEE Xplore digital*.

Kurniawan, W., & Mochammad, H. (2017). Teaching and learning support for computer architecture and organization courses design on computer engineering and computer science for undergraduate. *IEEE Xplore*, 1. Obtenido de <https://ieeexplore.ieee.org/abstract/document/8239076>

- M. Alnoukari, A. El Sheikh, Z. Alzoabi. (2009). "Data Mining and Simulation", In *Handbook of Research on Discrete Event Simulation Environments-Technologies And Applications*.
- Minaya Cañari, R. (2014). *Los simuladores en el logro de las capacidades en la unidad didáctica de proyectos electrónicos en el Instituto de Educación Superior Tecnológico Público "Carlos Cueto Fernandini" de Comas - 2014*. Lima: Universidad César Vallejo.
- Muñoz, C. (2011). *Como elaborar y asesorar una investigación de tesis*. Mexico: Pearson.
- Muñoz, C. (2011). *Como elaborar y asesorar una investigación de tesis*. México: Prentice Hall.
- Nachar, N. (2008). The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution. *Tutorials in Quantitative Methods for Psychology*, 13-20.
- Nova, B., Ferreira, J. C., & Araújo, A. (2013). Tool to Support Computer Architecture Teaching and Learning. *1st International Conference of the Portuguese Society for Engineering Education (CISPEE)* (págs. 1-8). Porto: IEEE.
- Null, L., & Lobur, J. (2015). *The Essentials of Computer Organization and Architecture*. United States of America: Jones and Bartlett Learning. Obtenido de https://books.google.com.pe/books?hl=es&lr=&id=3kQoAwAAQBAJ&oi=fnd&pg=PR1&dq=The+Essentials+of+Computer+Organization+and+Architecture&ots=senBShr2WL&sig=fIDo6CuCT_nzWG2Vf3ptJ_U9iMw&redir_esc=y#v=onepage&q=The%20Essentials%20of%20Computer%20Organization%20
- Patterson, D. A., & Hennessy, J. L. (2000). *Estructura y Diseño de Computadores*. Barcelona: Reverté, S.A. Recuperado el 18 de 12 de 2019, de https://books.google.com.pe/books?id=z1ukg428VjcC&pg=PR20&lpg=PR20&dq=download,+arquitectura+de+computadores+patterson+y+hennessey&source=bl&ots=VJYs0rfFsd&sig=ACfU3U1pX6egs0k1tFkS_14ugxA-Ug-7Zg&hl=es&sa=X&ved=2ahUKEwj3hrnRztvnAhVNHbkGHXCxDo8Q6AEwB3oECAoQA
- Patti, D., Spadaccini, A., Palesi, M., Fazzino, F., & Catania, V. (2012). Supporting Undergraduate Computer Architecture Students Using a Visual MIPS64 CPU Simulator. *IEEE Explore*, 406 - 411.
- Prasad, P., Alsadoon, A., Beg, A., & Chan, A. (01 de octubre de 2015). *Wiley Online Library*. Recuperado el 22 de 05 de 2019, de <https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.21699>
- R.N. Horspool, W.D. Lyons, M. Serra. (julio de 2010). *The ARMSim# User Guide*.

Razo, C. M. (2011). *Como elaborar y asesorar una investigacion de tesis*. México: Pearson Educación.

Sinchi Yupanqui, F. E. (2018). *La simulación computacional en el trabajo experimental para el aprendizaje de las asignaturas de circuitos eléctricos en la Facultad de Ingeniería Mecánica de la Universidad Nacional de Ingeniería*. Lima: Repositoria Universidad Peruana Cayetano Heredia.

Stallings, W. (2007). *Organización y arquitectura de computadoras*. México: Pearson Prentice Hall.

Tamayo, M. (2003). *El proceso de la investigación científica*. México: Editorial Limusa.

Yoshiro Imai, Masatoshi Imai, Yoshio Moritoh. (2013). Evaluation of visual computer simulator for computer architecture education. *IADIS International Conference e-Learning 2013*.

ANEXOS

1. Matriz de consistencia
2. Instrumentos validados
3. Base de datos
4. Reporte de estudiantes de ingeniería en informática y sistemas, semestre 2019-1
5. Apreciación del simulador ARMSim#
6. Guía de prácticas

ANEXO 01: Matriz de consistencia

Título: Simuladores de arquitectura de computadoras y su relación con en el rendimiento académico de los estudiantes en temas de arquitectura de computadoras, en la Universidad Nacional Agraria de la Selva

PROBLEMA	OBJETIVO	HIPÓTESIS	VARIABLES	DIMENSIÓN	INDICADORES	METODOLOGÍA
¿De qué manera influye el Simulador de Arquitectura de computadoras ARMSim# en el rendimiento académico en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019?	General Determinar la influencia del Simulador de Arquitectura de computadoras ARMSim# en el rendimiento académico en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019	General El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en el rendimiento académico en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019	VI: Simulador de Arquitectura de computadoras ARMSim#	-Simulador del procesador (CPU) y sus registros. -Simulador de ejecución de instrucciones.	<ul style="list-style-type: none"> ▪ Uso de registros. ▪ Operaciones del CPU y registros. ▪ Instrucciones del CPU 	<p>Tipo de investigación. Aplicada, Descriptiva, correlacional</p> <p>Diseño. Cuasi experimental</p> <p>Población. 265 estudiantes matriculados el 2019-1 en la FIIS-UNAS.</p> <p>Muestra. No probabilística, 35 estudiantes del curso sistemas operativos I, 2019-1</p> <p>Instrumentos. - Prueba de pretest y post test - Reglamento de estudios UNAS.</p>
Problemas específicos a) ¿De qué manera influye el Simulador de Arquitectura de computadoras ARMSim# en los resultados conceptuales en temas de arquitectura de	Objetivos específicos a) Determinar la influencia del Simulador de Arquitectura de computadoras ARMSim# en los resultados conceptuales en temas de arquitectura de	Hipótesis específicas a) El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados conceptuales en temas de arquitectura de computadoras en los	VD: Rendimiento académico en temas de arquitectura de computadoras	Conceptual	<ul style="list-style-type: none"> - Define arquitectura del computador. - Comprende los componentes funcionales del computador. - Identifica términos del computador. 	

computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019?	computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019	estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019		<ul style="list-style-type: none"> - Comprende las características y los elementos del procesador. - Comprende la ejecución de instrucciones del procesador.
b) ¿De qué manera influye el Simulador de Arquitectura de computadoras ARMSim# en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019?	b) Determinar la influencia del Simulador de Arquitectura de computadoras ARMSim# en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019	b) El Simulador de Arquitectura de computadoras ARMSim# influye significativamente en los resultados procedimentales en temas de arquitectura de computadoras en los estudiantes del tercer ciclo de la Facultad de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva año 2019	Procedimental	<ul style="list-style-type: none"> • Destreza para usar los registros del procesador • Destreza para analizar instrucciones de transferencia y suma • Destreza para analizar instrucciones lógicas • Habilidad en el uso de instrucciones de comparación y salto • habilidad para hacer programas en assembler

ANEXO 02: Instrumentos validados

PRUEBA PRETEST Y POSTTEST

(Conceptual y Procedimental)

Conceptual

1. Es una posible definición de arquitectura de computadoras,
 - a) Hardware de la computadora
 - b) Los componentes visibles de la computadora
 - c) Atributos de un sistema que son visibles a un programador en lenguaje de bajo nivel
 - d) Componentes hardware

2. Los componentes funcionales principales de una computadora son
 - a) ALU, memoria, entrada/salida
 - b) Unidad de control, procesador, memoria
 - c) CPU, teclado, monitor
 - d) Procesador, memoria, entrada/salida

3. Es uno de los componentes del modelo de Von Neumann que determina las operaciones de la memoria, de la ALU y del subsistema de entrada/salida.
 - a) Unidad de control
 - b) Memoria
 - c) Datos de entrada/salida
 - d) Unidad Aritmético-lógica

4. En el modelo de Von Neumann el área de almacenamiento, donde los programas y los datos se almacenan durante el procesamiento, es.
 - a) Unidad Aritmético-lógica
 - b) Memoria
 - c) Unidad de control
 - d) Datos de entrada/salida

5. En la informática y tecnologías, el término bus se refiere a:
 - a) Datos de bits entre dispositivos
 - b) Información de bits en dispositivos
 - c) Camino de comunicación entre dos o más dispositivos
 - d) Caminos entre datos e información

6. Señalar Verdadero (V) o Falso (F) acerca del procesador
- I. El procesador establece las características propias diferenciales de cada computador
 - II. El conjunto de todas las órdenes que es capaz de ejecutar, se denomina conjunto de instrucciones
 - III. El tamaño de la memoria, está determinadas por el procesador
- a) VVF b) VFV c) VFF d) VVV
7. Los elementos estructurales más importantes que se encuentran en todos los procesadores son
- a) Registros, buses y entrada/salida
 - b) Registros, unidades de transformación, circuitos digitales/secuenciadores y buses
 - c) ALU, ancho del bus y datos binarios
 - d) Registros, bits y bytes

Ejecución de instrucciones

8. Las fases para ejecutar una instrucción se denomina ciclo de instrucción. Las fases del ciclo de instrucción son:
- a) Escribe la instrucción, codificación de la instrucción, incremento del contador de programa y ejecución de la instrucción.
 - b) Lectura de la instrucción, codificación de la instrucción, incremento del contador de programa y ejecución de la instrucción
 - c) Lectura de la instrucción, decodificación de la instrucción, incremento del contador de programa y ejecución de la instrucción.
 - d) Escritura de la instrucción, decodificación de la instrucción, incremento del contador de programa y ejecución de la instrucción.
9. Señalar verdadero (V) o Falso (F) con respecto a los tipos de instrucción que puede ejecutar un procesador
- I. De transferencia de datos
 - II. De transformación de datos
 - III. De control de flujo de datos
 - IV. De programación de datos
- a) VVFFV b) VVVV c) VVFF d) VFVF

10. Para ejecutar un programa en una computadora, el programa está conformado por _____ que están en _____
- a) Instrucciones, memoria
 - b) Sentencias, memoria
 - c) Instrucciones, entrada/salida
 - d) saltos, memoria

Procedimental

11. Señala Verdadero (V) o Falso (F) los siguiente:

- I. Los registros contienen información
- II. Los registros son memorias pequeñas
- III. Los registros contienen bits

- a) VVV b) VVF c) FFV d) VFV

12. Para las líneas de código siguiente:

```
.text
mov r0,#0
mov r1,#0
mov r2,#28
```

UNAS:

```
cmp r0,r2
bge fin
add r0,r0,#7
add r1,r1,r0
b UNAS
```

fin:

```
.end
```

¿Qué valor contiene el registro r1 cuando finaliza el programa?

- a) 70 b) 42 c) 105 d) 35

13. En el siguiente programa

```
.text
main:
ldr r0,=0x87654321
ldr r1,=0x00F0F0F1
and r1,r0,r1
.end
```

Al terminar el programa, ¿Qué valor se almacena en el registro r1?

- a) 87654321h b) 604020h c) 004020h d) 00604021h

14. Dadas las líneas de código

```
.text
mov r7, #0xA9
mov r8, #0xD3
eor r7,r7, r8
```

Al finalizar el programa, ¿Qué valor se almacena en r7?

- a) 7bh b) 6ah c) 7ah d) 8ah

15. En el siguiente programa, ¿Qué valor se almacena en r9 tras su ejecución?

```
.text
mov r8, #0XF0
mvn r9, r8
```

- a) F0h b) 0Fh c) 0000000Fh d) FFFFFFF0Fh

16. En el siguiente programa, ¿Qué valor se almacena en r1 tras su ejecución?

```
.text
mov r0,#0
mov r1,#0
mov r2,#0xC
Lazo:
add r0,r0,#3
add r1,r1,r0
cmp r0,r2
bne Lazo
```

- a) 12h b) 1dh c) 1eh d) 2eh

17. En el siguiente programa

```
.text
mov r0,#12
mov r1,#16
cmp r0,#20
bge fin
add r0,r0,#4
add r1,r0,#2
```

fin:

```
.end
```

Al finalizar, ¿Qué valor almacena el registro r1?

- a) 34 b) 32 c) 18 d) 16

18. En el siguiente programa

```
.data          @zona de datos
/*****
*****/
.text          @zona de codigo
mov r0,#80
mov r1,#100
cmp r0,r1
blt else
add r0,r0,#20
add r1,r1,#20
b fin
else:
sub r0,r0,#20
sub r1,r1,#20
fin:
.end
```

Al finalizar el programa, ¿Qué valor contiene el registro r0?

- a) 100 b) 60 c) 80 d) 120

19. En el siguiente programa

```
.data
@los datos
.text          @la zona de codigo
ldr r0,=0xf4f2f6
mov r1,#4
lsl r0,r1
.end
```

Al finalizar el programa, ¿Qué valor contiene el registro r0?

- a) f4f2fh b) 4f2f60h c) 7a797bh d) f4f2eh

20. En el siguiente programa,

```
.text
ldr r0,=0x221122
mov r1,#2
lsl r0,r1
.end
```

Al finalizar el programa ¿Qué valor contiene el registro r0?

- a) 112200h b) 882244h c) 882288h d) 884488h

VALIDACIÓN DE INSTRUMENTO

NOMBRE DEL EXPERTO: Pedro C. Fajardo Matute


1 = No cumple con el criterio, 2 = Bajo nivel, 3 = Moderado nivel, 4 = Alto nivel

Dimensión	Ítem	suficiencia	Coherencia	Relevancia	Claridad
Conceptual	1. Es una definición de arquitectura de computadoras	4	4	3	4
	2. ¿Cuáles son los tres principales componentes funcionales de un computador?	4	4	3	4
	3. Es uno de los componentes del modelo de Von Neumann que determina las operaciones de la memoria, de la ALU y del subsistema de entrada/salida.	4	4	4	4
	4. En el modelo de Von Neumann el área de almacenamiento, donde los programas y datos se almacenan durante el procesamiento, es:	4	4	4	4
	5. Existen denominaciones técnicas en las computadoras, una de ellas el término bus, se dice bus de datos; bus de direcciones. ¿Cuál es la alternativa que define al término bus?	4	4	3	4
	6. Son conceptos relacionados con el procesador.	3	4	4	4
	7. ¿Cuáles son los elementos estructurales más importantes que se encuentran en todos los procesadores?	4	3	4	4
	8. ¿Cuáles son las fases de ejecución de una instrucción o ciclo de instrucción?	4	4	4	4
	9. ¿Cuáles son tipos de instrucciones que puede ejecutar un procesador?	4	4	4	3
	10. En forma más básica, un programa informático está conformado por:	4	3	4	4

Dimensión	Ítem	suficiencia	Coherencia	Relevancia	Claridad
Procedimental	11. ¿Cuáles son las características de los registros	4	4	3	4
	12. Operaciones de transferencia y suma	4	4	4	3
	13. Operación lógica AND	4	4	4	4
	14. Operación lógica OR Exclusiva	4	4	4	4
	15. Operación de complemento a uno	4	4	4	4
	16. Instrucción de comparación y salto	4	4	4	4
	17. ¿Qué es una instrucción en código de máquina?	3	4	4	4
	18. ¿Qué es una directiva?	4	4	4	4
	19. En un programa, ¿Cómo se especifica la zona de código en memoria?	4	4	4	3
	20. Instrucción de desplazamiento a la izquierda	4	3	4	4

Opinión de aplicabilidad: *Si es posible su aplicación*

Tingo María, 2019



 DNI 09648055

VALIDACIÓN DE INSTRUMENTO

NOMBRE DEL EXPERTO: Bruno Pardo Soto

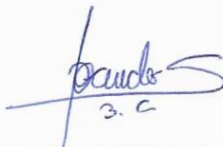
1 = No cumple con el criterio, 2 = Bajo nivel, 3 = Moderado nivel, 4 = Alto nivel

Dimensión	Ítem	suficiencia	Coherencia	Relevancia	Claridad
Conceptual	1. Es una definición de arquitectura de computadoras	4	4	4	3
	2. ¿Cuáles son los tres principales componentes funcionales de un computador?	3	4	4	3
	3. Es uno de los componentes del modelo de Von Neumann que determina las operaciones de la memoria, de la ALU y del subsistema de entrada/salida.	4	4	4	4
	4. En el modelo de Von Neumann el área de almacenamiento, donde los programas y datos se almacenan durante el procesamiento, es:	4	4	4	4
	5. Existen denominaciones técnicas en las computadoras, una de ellas el término bus, se dice bus de datos; bus de direcciones. ¿Cuál es la alternativa que define al término bus?	4	4	4	4
	6. Son conceptos relacionados con el procesador.	4	4	4	3
	7. ¿Cuáles son los elementos estructurales más importantes que se encuentran en todos los procesadores?	4	4	4	4
	8. ¿Cuáles son las fases de ejecución de una instrucción o ciclo de instrucción?	4	4	4	4
	9. ¿Cuáles son tipos de instrucciones que puede ejecutar un procesador?	4	4	4	4
	10. En forma más básica, un programa informático está conformado por:	4	4	4	3

Dimensión	Ítem	suficiencia	Coherencia	Relevancia	Claridad
Procedimental	11. ¿Cuáles son las características de los registros	4	4	4	3
	12. Operaciones de transferencia y suma	4	4	4	3
	13. Operación lógica AND	4	4	4	4
	14. Operación lógica OR Exclusiva	4	4	4	4
	15. Operación de complemento a uno	4	4	4	4
	16. Instrucción de comparación y salto	4	4	4	4
	17. ¿Qué es una instrucción en código de máquina?	3	4	4	4
	18. ¿Qué es una directiva?	4	4	4	3
	19. En un programa, ¿Cómo se especifica la zona de código en memoria?	4	4	4	4
	20. Instrucción de desplazamiento a la izquierda	4	4	4	4

Opinión de aplicabilidad: ES FACTIBLE SU APLICACIÓN.

Tingo María, 2019


Pauls
S.C.

DNI 43766096

ANEXO 03: Base de datos

Resultados de las pruebas pretest y postest del Grupo experimental, Est i=estudiante i.

	Pretest			Postest		
	Cognitivo	Procedimental	Total	Cognitivo	Procedimental	Total
Est1	4	2	6	7	10	17
Est2	3	2	5	7	9	16
Est3	3	3	6	5	5	10
Est4	8	3	11	7	8	15
Est5	5	2	7	6	1	7
Est6	3	1	4	7	9	16
Est7	6	1	7	6	7	13
Est8	7	4	11	8	7	15
Est9	5	3	8	6	8	14
Est10	3	2	5	7	7	14
Est11	5	3	8	7	9	16
Est12	3	2	5	7	10	17
Est13	6	4	10	7	3	10
Est14	6	2	8	6	7	13
Est15	7	4	11	5	8	13
Est16	4	2	6	7	5	12
Est17	1	3	4	5	9	14

Continuación de resultados de pretest y postest Grupo de control

	Pretest			Postest		
	Cognitivo	Procedimental	Total	Cognitivo	Procedimental	Total
Est18	7	3	10	9	2	11
Est19	8	3	11	4	7	11
Est20	4	3	7	8	2	10
Est21	3	3	6	1	3	4
Est22	5	3	8	7	3	10
Est23	6	4	10	8	5	13
Est24	7	2	9	7	9	16
Est25	6	1	7	6	4	10
Est26	1	3	4	5	3	8
Est27	8	3	11	6	8	14
Est28	7	1	8	9	6	15
Est29	6	3	9	6	5	11
Est30	5	0	5	8	6	14
Est31	3	2	5	5	4	9
Est32	6	2	8	4	7	11
Est33	6	1	7	7	4	11
Est34	5	2	7	6	6	12
Est35	9	2	11	8	7	15

Base de datos sobre facilidad de uso y satisfacción de usar el simulador ARMSim#, Pi= pregunta i, En = estudiante n.

	E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E1 0	E1 1	E1 2	E1 3	E1 4	E1 5	E1 6	E1 7
P 1	4	4	3	3	5	5	4	4	4	3	4	4	4	4	4	4	5
P 2	5	4	3	5	5	5	4	4	4	4	4	5	3	3	3	4	4
P 3	5	5	4	4	5	5	4	4	5	4	4	5	4	3	4	4	4
P 4	4	5	3	4	4	4	4	4	4	4	4	5	4	4	3	4	3
P 5	4	5	4	4	5	4	4	4	3	4	4	5	3	4	4	5	4

Resultados de encuesta del curso

ITEMS					
	Tdes	Des	Ind	Ac	Tac
1. Me resultó sencillo aprender usar los registros del procesador en el ARMSim#.	0	0	3	11	3
2. Estoy satisfecho de haber realizado operaciones con los registros en el ARMSim#	0	0	4	8	5
3. Estoy satisfecho de haber realizado operaciones de CPU en el ARMSim#	0	0	1	10	6
4. Estoy satisfecho haber realizado programas en assembler en el ARMSim#	0	0	3	12	2
5. El simulador favoreció mi aprendizaje en arquitectura del computador	0	0	2	11	4

0% 0% 15% 61% 24%

ANEXO 04: Reporte de estudiantes de ingeniería en informática y sistemas, semestre 2019-1

UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA		
TINGO MARIA		
DIRECCION DE COORDINACIÓN Y DESARROLLO ACADEMICO		Fecha: 02/12/2019
TOTAL DE MATRICULADOS 2019-1		Hora : 12:41:45
		P g. : 1
Fecha: 02/12/2019 12:41:46 p.m.		

ESPECIALIDAD	SUB TOTAL	

ECONOMIA	306	
INGENIERIA EN INDUSTRIAS ALIMENTARIAS	214	
INGENIERIA MECANICA ELECTRICA	119	
ZOOTECNIA	227	
INGENIERIA AMBIENTAL	291	
INGENIERIA EN INFORMATICA Y SISTEMAS	261	
AGRONOMIA	349	
INGENIERIA FORESTAL	276	
CONTABILIDAD	332	
ADMINISTRACION	328	
INGENIERIA EN RECURSOS NATURALES RENOVAB	235	
INGENIERIA EN CONSERVACION DE SUELOS Y A	287	

		3225



ANEXO 05: Apreciación del simulador ARMSim#

Cuestionario de apreciación sobre el simulador ARMSim#

El presente cuestionario pretende obtener su opinión sobre el simulador **ARMSim#**, en temas de arquitectura de computadoras.

Se le solicita marcar con una (X) dentro del paréntesis que corresponda a su respuesta, en cada ítem, según la escala siguiente

1 = Totalmente en desacuerdo (Tdes)

2 = en Desacuerdo (Des)

3 = indiferente (Ind)

4 = de Acuerdo (Ac)

5 = Totalmente de acuerdo (Tac)

ITEMS	Alternativas				
	1	2	3	4	5
1. Me resultó sencillo aprender usar los registros del procesador en el ARMSim#.					
2. Estoy satisfecho de haber realizado operaciones con los registros en el ARMSim#					
3. Estoy satisfecho de haber realizado operaciones de CPU en el ARMSim#					
4. Estoy satisfecho haber realizado programas en assembler en el ARMSim#					
5. El simulador favoreció mi aprendizaje en arquitectura del computador					

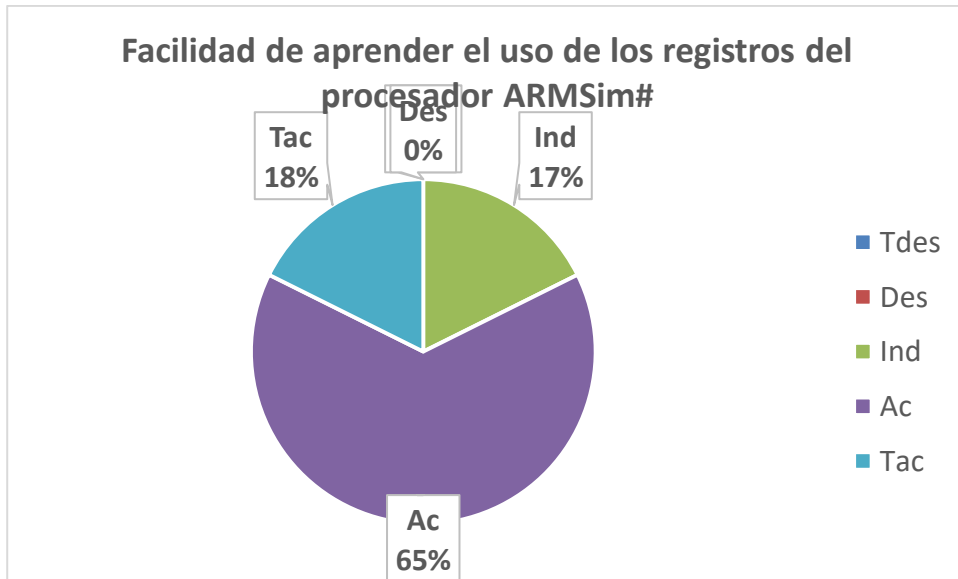


Gráfico 7 Facilidad de aprender usar los registros del procesador ARM

Respecto a la facilidad de uso del simulador ARMSim# los resultados son los siguientes 18% estuvieron totalmente de acuerdo, 65% mencionaron estar de acuerdo, 17% no están de acuerdo ni en desacuerdo, 0% en desacuerdo y 0% totalmente de acuerdo. Un 83% opinaron a favor sobre la facilidad de uso del simulador frente a un 17% que opinaron ser indiferentes.

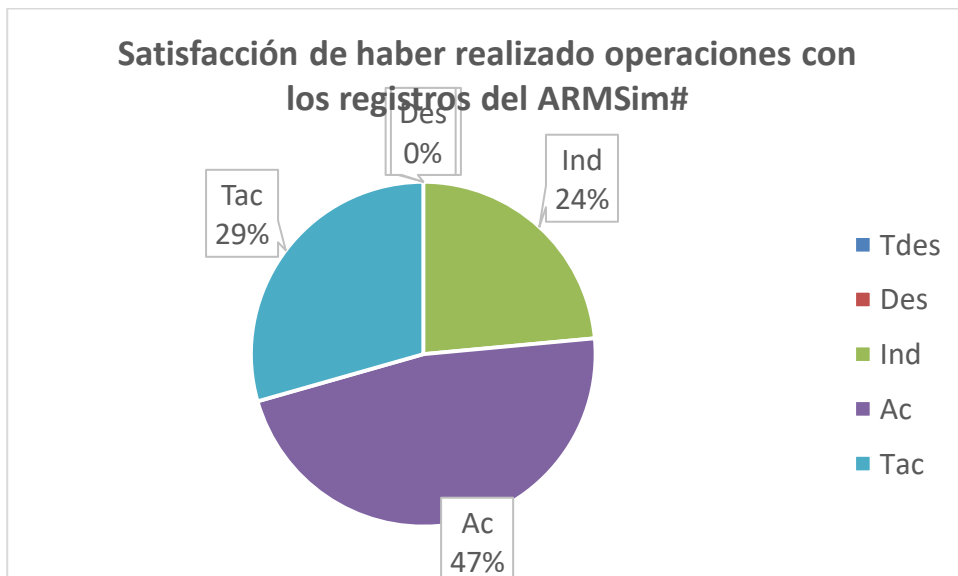


Gráfico 8 Satisfacción de realizar operaciones con registros del ARMSim#

En cuanto a la satisfacción de realizar operaciones con los registros del simulador ARMSim#, un 76 % menciona estar satisfecho de haber realizado operaciones con los registros en el simulador ARMSim#, frente a un 24 % que fueron indiferentes al respecto.

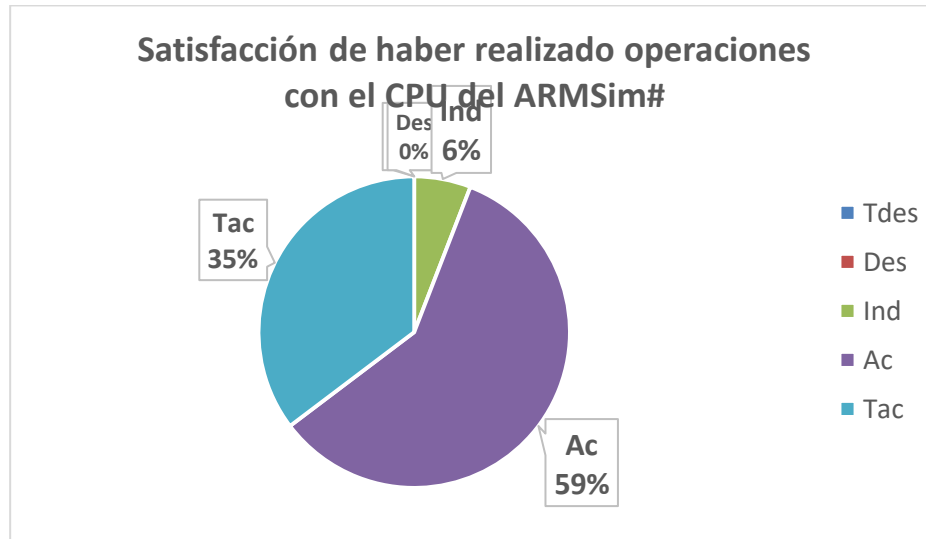


Gráfico 9 Satisfacción de haber realizado operaciones con el ARMSim#

Satisfacción de haber realizado operaciones con el CPU del ARMSim#, el 94% mencionó su satisfacción de haber realizado operaciones con el CPU del ARMSim#. Solamente el 6 % fue indiferente.

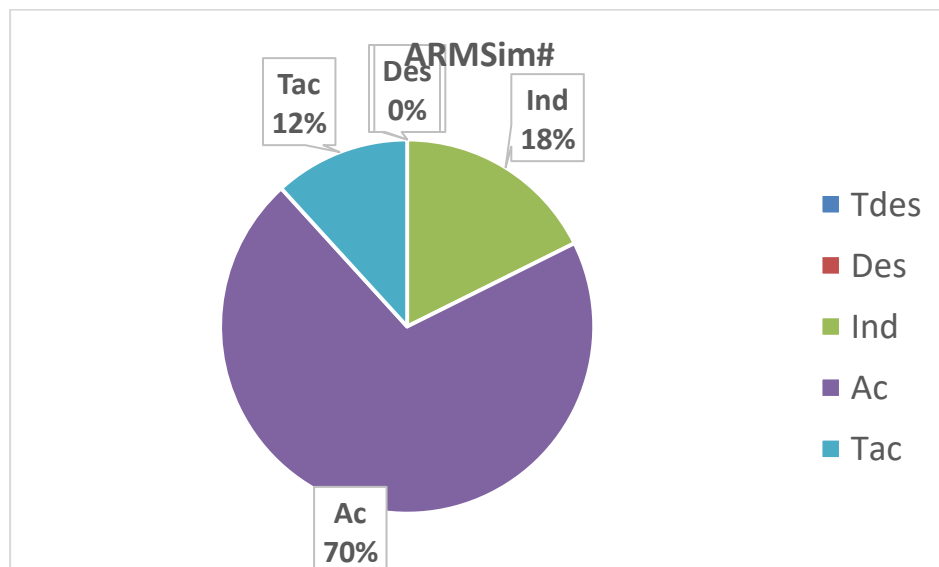


Gráfico 10 Satisfacción de haber realizado programas en el ARMSim#.

Satisfacción de haber realizado programas en assembler en el ARMSim#, 82% mencionó estar de acuerdo y 18 % fue indiferente.

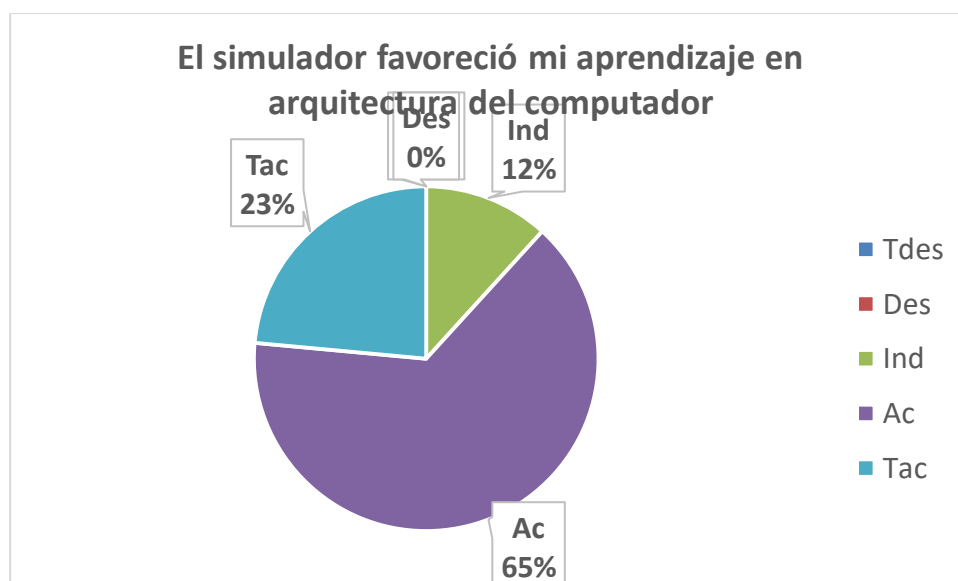


Gráfico 11 El simulador favoreció el aprendizaje de arquitectura de computadoras

Acerca de si el simulador favoreció el aprendizaje de los estudiantes, el 88% está de acuerdo con esta afirmación y el 12% fue indiferente.

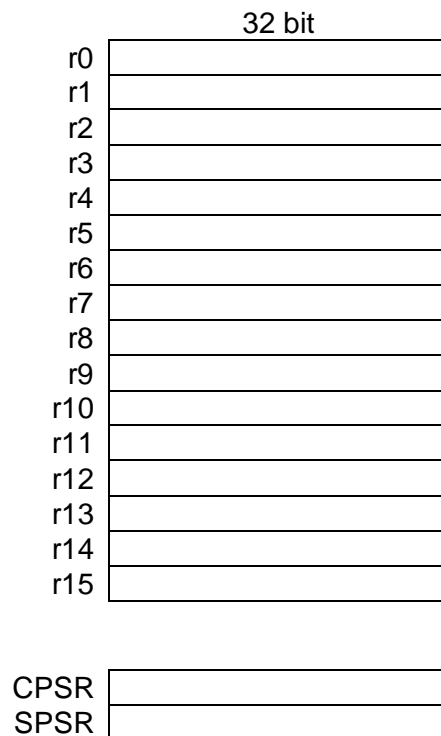
ANEXO 06: Guía de prácticas

El procesador ARM (Advance RISC Machine), fue desarrollado en Inglaterra por la compañía Acorn Computer Limited entre los años 1982 y 1985. Es un procesador RISC, proporciona dos juegos de instrucciones que son diferentes. Uno el estándar con instrucciones de 32 bits, y el denominado Thumb con su mayoría de instrucciones de 16 bits. Las instrucciones aritméticas se realizan sobre los registros no sobre memoria lo que se conoce como load store.

El procesador ARM está diseñado para trabajar según el modelo Von Neumann, es decir las instrucciones y datos están en la misma memoria, la memoria es direccionable por byte y las palabras en memoria son de cuatro bytes.

Son 37 registros, de los cuales 18 son visibles los demás no pueden ser invocados en el código del programador en assembler.

Registros del procesador ARM



Donde: r0, r1, r2,, r12, son registros de propósito general.

r13 o (SP): Registro puntero de pila.

r14 o (LR): Registro de enlace que guarda la dirección de retorno

r15 o (PC): Registro contador de programa.

Registro de estado (CPSR; Current Program Status Register).

Registro CPSR

Bits de bandera				reservados				Bits de control							
31	30	29	28	27	26						4	3	2	1	0
N	Z	C	V	I	F	T		M4	M3	M2	M1	M0

N (Negativo): Es “1” si el resultado de una operación fue negativo

Z (Zero): Es “1” si el resultado de una operación fue “0”

C (Carry): Es “1” si una adición, sustracción o complemento causa un resultado mayor de 32 bits, también en una instrucción de desplazamiento.

V (Overflow): En suma o resta, es “1” si hubo desbordamiento.

Programación en assembler

El lenguaje ensamblador aparte del juego de instrucciones propias de la arquitectura, proporciona recursos adicionales para facilitar la programación en el mismo. Algunos recursos adicionales son los siguientes

Directivas. Indican al compilador (ensamblador) que realice determinadas tareas en el proceso de compilación. Una directiva va precedida por un punto, ejemplo: .txt, .end, etc.

Etiquetas. Hacen referencia, a la dirección de memoria del elemento contenido en la línea en la que se encuentra. Por lo general van al inicio de una línea, están conformadas por letras y números y terminan en dos puntos.

Comentarios. Los comentarios van precedidos por el carácter “@”, también se puede usar el carácter “#”.

Pseudoinstrucciones. No forman parte del repertorio de instrucciones de la arquitectura, facilitan la labor del programador, el ensamblador traducirá la pseudoinstrucción en instrucciones máquina.

Estructura de un programa en assembler para el ARM

.txt

main:

mov r1, #7 @ r1 <- 7

mov r2, #8 @ r2<- 8

add r3, r1, r2 @ r3 <- r1 + r2

...

.end

Instrucciones aritméticas

Instrucción	rd, m1, m2	@ rd ← m1 operación m2
add	rd, rs1, rs2	@ rd ← rs1 + rs2
sub	rd, rs1, rs2	@ rd ← rs1 - rs2
subs	rd, rs1, rs2	@ rd ← rs1 - rs2 y actualiza N, Z, C y V según caso
adc	rd, rs1, rs2	@ rd ← rs1 + rs2 + Carry Flag
cmp	rs, rn	@ rs - rn, activa indicadores N, Z, C o V según caso
neg	rd, rs	@ rd ← -rs
mov	rd, rs	@ rd ← rs

Instrucciones de multiplicación

mul	rd, rs1, rs2	@ rd ← rs1 * rs2
muls	rd, rs1, rs2	@ rd ← rs1 * rs2, modifica los flags según resultado

Instrucciones lógicas

and	rd, rs1, rs2	@ rd ← rs1 AND rs2
orr	rd, rs1, rs2	@ rd ← rs1 OR rs2
eor	rd, rs1, rs2	@ rd ← rs1 XOR rs2
mvn	rd, rs1	@ rd ← NOT(rs1)

Instrucciones de desplazamiento

asr rd, rs1, #n @ rd ← desplazamiento aritmético a la derecha.

lsl rd, rs1, #n @ rd ← desplazamiento lógico a la derecha



lsl rd, rs1, #n @ rd ← desplazamiento lógico a la izquierda

Ejemplo.

```
.txt
main: mov r0, #0xF      @ r0<- 00001111
      mov r1, #1        @ r1<-00000001
      lsl r0, r1        @ r0<-00000111
      .end
```

Transferencia de datos

mov rd, #Inm8 @ carga un dato inmediato de 8 bits

ldr rd, =Inm32 @ cargar un dato inmediato de 32 bits

Instrucciones de acceso a memoria

ldr rd, [rb] @ rd ← memoria(rb)

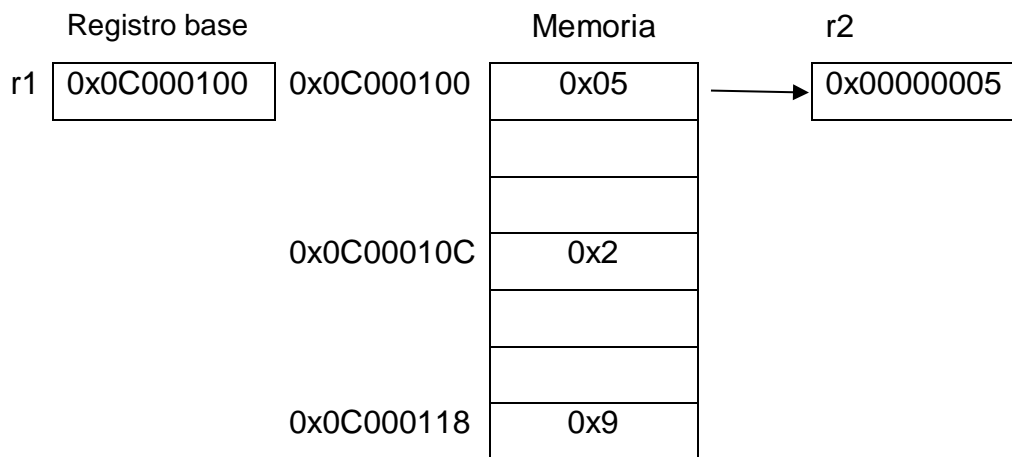
str rf, [rb] @ rf → memoria(rb)

`ldr rd, [rb, #desp] @ rd ← memoria(rb + desp)`

`str rf, [rb, #desp] @ rf → memoria(rb + desp)`

Ejemplo.

`Ldr r2, [r1] @ r2 ← 0x05`



Ejemplo.

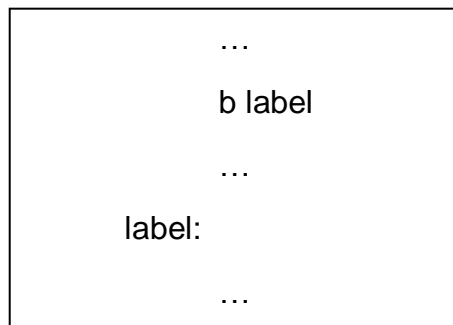
`ldr r2, [r1, #12] @ r2 ← 0x2`

Instrucciones de salto

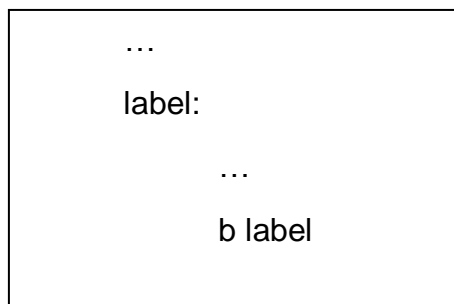
Cuando se programa en assembler, las estructuras de control se implementan mediante las instrucciones de salto. Los saltos pueden ser incondicionales o condicionales. La instrucción de salto branch “b” se considera explícitamente.

Estructura de saltos incondicionales

Salto hacia adelante



Salto hacia atrás



Instrucciones condicionales

La instrucción anterior a la de salto es comparar CMP, que activa las condiciones. Las instrucciones de salto son:

Instrucción	Descripción	Flag
beq (branch if equal)	Salta si son iguales	Z = 1
bne (Branch if not equal)	Salta si son diferentes	Z = 0
bcs (branch if carry set)	Salta si C = 1	C = 1
bcc (branch if carry clear)	Salta si C = 0	C = 0
bls (Branch if lower or same)	Salta si es menor o igual	C = 0 or Z = 1
bge (Branch if greater or equal)	Salta si es mayor o igual	N = V
blt (Branch if less than)	Salta si es menor que	N! = V
bgt (Branch if greater than)	Salta si es mayor que	Z = 0 & N = V
Ble (branch if less than or equal)	Salta si es menor o igual.	Z = 1 or N! = V

Antes de una instrucción de salto se utiliza la instrucción de comparación cmp.

Ejemplo. Uso una instrucción condicional.

```
...  
    cmp r7, 10  
    beq label  
    add r0, r0, r1  
    sub r0, r0, r2  
label:
```

Ejemplo. Implementación de la instrucción do-while

```
mov r0, #10
```

Loop:

```
...
```

```
subs r1, r1, #1
```

```
bne loop
```

```
....
```

Ejercicios resueltos

1. Inicializar registros con valor numérico

```
.txt
```

```
Start: mov r0, #25           @ 25 en decimal  
      mov r1, #0x19        @ 25 en hexadecimal  
      mov r2, #031         @ 25 en octal  
      mov r3, #0b00011001 @ 25 en binario  
      .end
```

2. Especificar carácter de forma literal

```
.text
```

```
Mov r0, #'H'  
mov R1, #'O'  
mov R2, #'L'  
mov R3, #'A'  
.end
```

3. Declarar constantes

```
.data
.equ  lunes, 1
.equ  martes, 2

.text
mov r0, #lunes
mov r1, #martes
.end
```

4. Operaciones aritméticas

```
.text          @este es el código
mov r0, #8     @ r0 ← 8
mov r1, #7     @ r1 ← 7
add r2, r1, r0 @ r2 ← r1 + r0
sub r3, r1, r0 @ r3 ← r1 - r0
.end
```

5. Operaciones de multiplicación

```
.text

main:

mov r0, #10    @ r0 ← 10
mov r1, #4     @ r1 ← 4
mul r2, r0, r1 @ r2 ← r0 * r1
.end
```

6. Operaciones lógicas

```
.text
Main:
mov r0, #0x07
mov r1, #0x0E
and r2, r1, r0
.end
```

7. Operación de desplazamiento

```
.text

main:

mov r0, #0x0F
mov r1, #1
lsl r0, r1          @ desplazamiento lógico a la derecha
.end
```

8. Uso de instrucción de salto condicional y formación de lazo

```
.text
mov r0, #0
mov r1, #0
mov r2, #4
Lazo:
add r1, r1, #1
mov r0, r1
sub r2, r2, #1
cmp r2, #0
bne lazo
.end
```

9. Suma de los 5 primeros números naturales

```
.text
mov r0, #0
mov r1, #0
Lazo:
add r1, r1, #1
add r0, r0, r1
cmp r1, 5
bne lazo
.end
```

10. Hallar el factorial de los 4 primeros números naturales

```
.text
main:
mov r0, #0      @ r0 = n = 0
mov r1, #1      @ r1 = fact = 1
while:
add r0, r0, #1  @ r0 = r0+1
mul r2, r2, r1  @ r2 = r2*r1
mov r1, r2      @ r1 = r2
cmp r1, 5
bne while
.end
```

Ejercicios propuestos

1. Analizar el siguiente programa

```
.text  
  
main: mov r0, #10  
      mov r1, #20  
      add r2, r0, r1  
      .end
```

Al finalizar, como quedan los registros r2 = ____, r1 = ____ y r0 = ____.

2. En el siguiente programa

```
.data                @ inicio de la zona de datos  
Varx1: .word         17    @ número decimal  
Varx2: .word         0x18 @ número hexadecimal  
Varx3: .word         016   @ número octal  
Varx4: .word         0b101 @ número binario  
  
.text
```

Ubica en que direcciones de memoria se ubican los números, 17, 0x18, 016 y 0b101.

3. Realizar lo siguiente, cargar los registros r5 y r6 con los valores 0xF0 y 0xAA, luego realizar las operación AND y el resultado debe quedar en el registro r0, también la operación OR y el resultado en el registro r1,
4. El contenido de un registro es el número 27, multiplicar por 4 al contenido del registro, usar la instrucción desplazamiento lógico a la izquierda *lsl*

5. En el siguiente programa, el que se utiliza la instrucción “beq”

```
.text
main: mov r0,#5
      mov r1,#10
      mov r2,#15
      mov r3,#0
      cmp r0,r2
      beq salto
          add r3,r0,r1
      b fin
salto: add r3,r3,r1
fin:
      .end
```

¿Qué valor contiene el registro r3 al finalizar el programa?

6. El siguiente programa, muestra un ejemplo de salto condicional

```
.text
main: mov r0,#40
      mov r1,#70
      mov r2,#200
      mov r3,#0
      b salto
          add r3,r1,r0
salto: add r3,r3,r2
      .end
```

¿Qué valor contiene el registro r3 al finalizar el programa?

7. En assembler ARM implementar el equivalente de la sentencia siguiente

```
if (x<50)
{
    x=x+5;
    y=x+2;
}
```

8. En assembler ARM implementar el equivalente de la sentencia siguiente:

```
if (a>10)
{
    a= a-2;
}
else
{
    a=a+5;
}
```

9. En assembler ARM implementar el equivalente de la sentencia siguiente:

```
sum=0;
term=0;
while (num<=100)
{
    term=term+1;
    sum=sum+term;
}
```

10. En assembler ARM implementar el equivalente de la sentencia siguiente:

```
x=10;
y= 0;
do
{
    y=y+x;
    x=x-1;
}
```