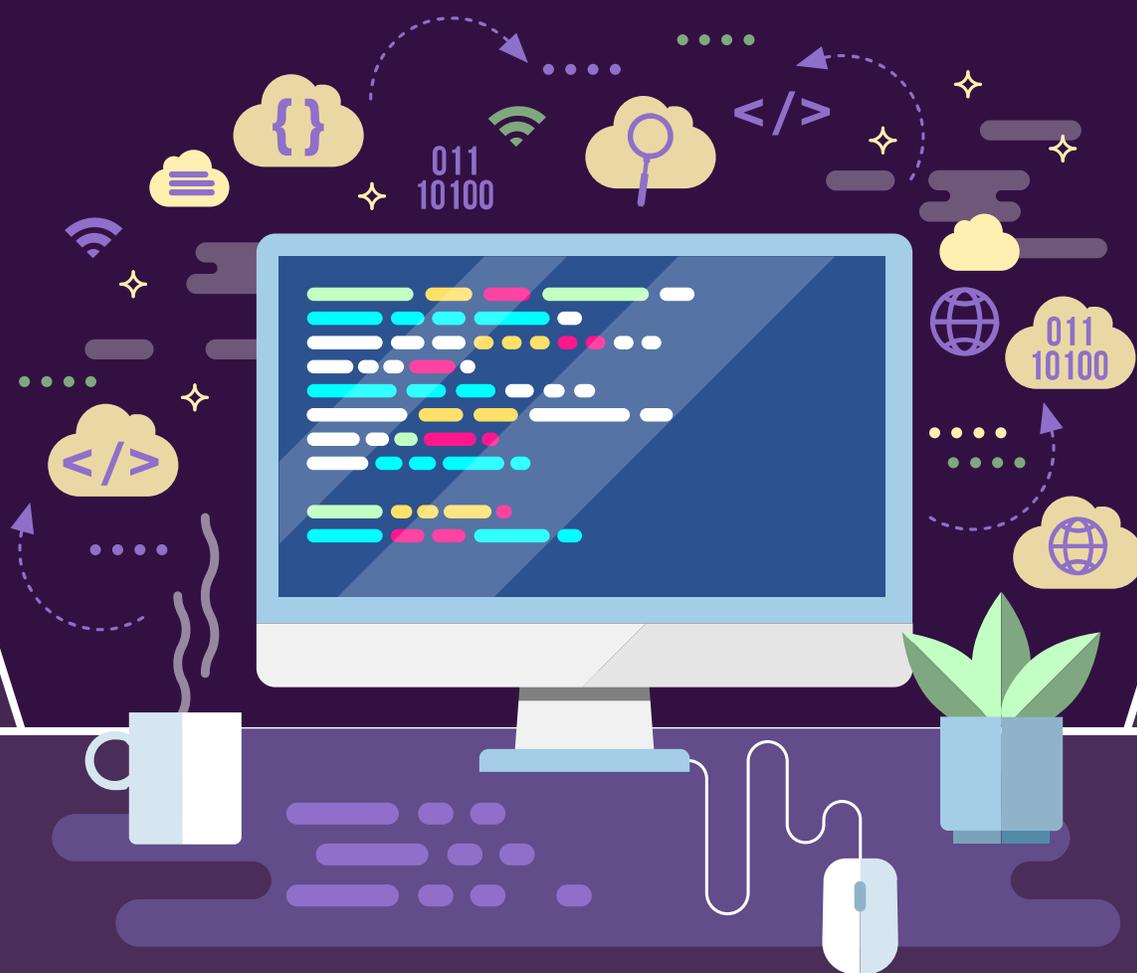


# LABORATORIO VIRTUAL PARA EL APRENDIZAJE DE FUNDAMENTOS DE PROGRAMACIÓN



 Universidad de la  
Amazonia

# LABORATORIO VIRTUAL PARA EL APRENDIZAJE DE FUNDAMENTOS DE PROGRAMACIÓN

CASO DE ESTUDIO DE LA ASIGNATURA LÓGICA Y ALGORITMOS I  
DEL PROGRAMA INGENIERÍA DE SISTEMAS  
UNIVERSIDAD DE LA AMAZONIA



Esta obra deberá ser citada de la siguiente manera:

García Quintero, J. A. Espinosa Sarmiento, D. M. Álvarez Guayara, D. L. Sánchez Trujillo, E. F. Vargas Losada, H. F. Zuluaga Ramírez, L. F. 2024. Laboratorio virtual para el aprendizaje de fundamentos de programación. Editorial. Universidad de la Amazonia. p.p 121

Tamaño (21,59 x 27,94 cm).

Incluye bibliografía.

© Editorial - Universidad de la Amazonia

Autor (es): García Quintero John Arley, Espinosa Sarmiento Diana María, Álvarez Guayara Denis Lorena, Sánchez Trujillo Elio Fabio, Vargas Losada Heriberto Fernando, Zuluaga Ramírez Luisa Fernanda

ISBN (Digital): 978-628-7693-20-3

Número y año de edición: Primera edición, 2024.

Diagramación y diseño de cubierta:

Cristian Andres Cañon Recalde

Tiraje: Online

Depósito Legal: Biblioteca Nacional de Colombia.

Universidad de la Amazonia  
Vicerrectoría de Investigaciones y Posgrados  
Editorial Universidad de la Amazonia  
Campus Porvenir: Calle 17 Diagonal 17 con Carrera 3F -  
Barrio Porvenir  
Contacto: [vrinvestigaciones@udla.edu.co](mailto:vrinvestigaciones@udla.edu.co)



Prohibida la reproducción total o parcial de este con fines comerciales. Su utilización se puede realizar con carácter académico, siempre que se cite la fuente

# Prólogo

*Con total seguridad, la emoción más asociada con la creación en la computación es el algoritmo. Para quienes amamos la Algoritmia, los programas de software son la esencia y alegría de ese infinito mundo mágico que empieza con una idea y traspasa todas las fronteras del conocimiento, hasta lograr permear diversas áreas del mundo creado, conocido y soñado.*

*Aunque el desarrollo de algoritmos es grato y sencillo para algunos, su aprendizaje es arduo y difícil para otros. El proceso de abstraer el mundo para modelar y moldear las ideas de unos creativos diseños requiere conocer los secretos de la programación y la creación de algoritmos claros, sencillos, útiles, pero sobre todo eficientes en su lenguaje de desarrollo y en su ejecución.*

*En un mundo basado y controlado por las maravillas de la computación es deseable usar sus beneficios también en la educación. Sin embargo, la pregunta crucial siempre ha sido cómo involucrar la Tecnología Informática para facilitar el aprendizaje de la algoritmia, el desarrollo de software, la apropiación de la mentalidad lógica, tan necesarios en la configuración de destrezas y habilidades de un buen programador.*

*El trabajo aquí presentado como producto del proyecto de investigación de docentes del programa de Ingeniería de Sistemas de nuestra Universidad, es un aporte desde lo local para contribuir a resolver las necesidades propias de nuestros estudiantes.*

*Este texto que hoy Usted puede hojear tiene un origen noble: cuando se trabaja en Tecnologías de la Información y las Comunicaciones desde la periferia y se convive con las dificultades educativas estructurales, existe una inquietud rectora originada en la preocupación de encontrar adecuadas alternativas que contribuyan a mejorar nuestros procesos académicos.*

*Hace unos años un grupo de docentes de la Universidad de la Amazonia seriamente preocupados por las estadísticas de repitencia y deserción, especialmente en las asignaturas iniciales de programación de computadores en la Facultad de Ingeniería, deseábamos encontrar propuestas basadas en TIC que facilitasen la comprensión de las temáticas relacionadas con el arte de aprender a programar computadores.*

*Como docentes de Ingeniería de Sistemas estamos convencidos que sólo se aprende a programar programando. Para lograr la experticia requerida en programación, que sea útil para la profesión y la vida, tenemos que resolver muchos problemas y desarrollar gran cantidad de algoritmos. Es esa la premisa básica que nos inspira a producir y presentar esta propuesta docente como herramienta que facilite la enseñanza y el aprendizaje de la algoritmia.*

*El producto resultante de aquella idea gestora es el presente texto que busca describir el proceso completo desde la identificación de la problemática específica de nuestra realidad hasta la propuesta de una herramienta basada en TIC que plantea una alternativa de solución para la línea de programación de la carrera y eventualmente a otros espacios académicos de la Facultad de Ingeniería.*

*Esperamos que nuestro sueño trascienda y que este aporte creativo contribuya al innegable impacto multidimensional que genera la adecuada construcción de software en la sociedad actual.*

**Elio Fabio Sánchez Trujillo**

# *Tabla de Contenido*

PRESENTACIÓN	16
CAPÍTULO I	
PLANTEAMIENTO DEL PROBLEMA	18
1.1 INTRODUCCIÓN	19
1.2 OBJETIVOS DEL ESTUDIO	19
1.2.1 General	19
1.2.2 Específicos	19
1.3 CONTEXTUALIZACIÓN SOBRE LOS PROBLEMAS DEL APRENDIZAJE DE LOS FUNDAMENTOS DE PROGRAMACIÓN	20
1.4 CASO PROGRAMA INGENIERÍA DE SISTEMAS UNIVERSIDAD DE LA AMAZONIA	21
1.4.1 La enseñanza y el aprendizaje de los fundamentos de programación en Ingeniería de Sistemas.	22
1.4.2 Lógica y Algoritmos I: Competencias y Resultados De Aprendizaje	24
1.5 MARCO REFERENCIAL	26
1.5.1 Antecedentes	26
1.5.1.1 El uso de entornos y herramientas virtuales de aprendizaje.	26
1.5.1.2 Diseños metodológicos.	27
1.5.1.3 Laboratorios virtuales y entrenamiento para maratones de programación.	27

1.5.2 Casos de implementación de VPL	28
1.5.2.1 Laboratorio Virtual de Programación Java.	28
1.5.2.2 VPL Universidad Nacional de Río Negro, Patagonia Argentina.	28
1.5.2.3 VPL de la Universidad de Córdoba, España.	29
1.5.2.4 VPL con Autoevaluación Habilitada por Tecnología.	29
1.5.3 Aspectos teóricos y conceptuales para el diseño y el despliegue de un VPL	29
1.5.3.1 Laboratorio Virtual de Programación (VPL).	29
1.5.3.2 Moodle (Learning Managment System).	30
1.5.3.3 Moodle en la enseñanza.	31
1.5.3.4 VPL diseño y despliegue.	31
1.5.3.5 VPL Cloud computing.	31
1.5.3.6 Docker.	32
1.5.3.7 Docker compose.	32
<b>1.6 REFERENCIAS BIBLIOGRÁFICAS</b>	<b>34</b>
<b>CAPÍTULO II</b>	
<b>DESPLIEGUE EN LA NUBE DEL LABORATORIO VIRTUAL DE PROGRAMACIÓN</b>	<b>38</b>
<b>2.1 PRESENTACIÓN</b>	<b>38</b>
<b>2.2 METODOLOGÍA PARA EL DISEÑO Y DESPLIEGUE DEL LABORATORIO VIRTUAL DE PROGRAMACIÓN</b>	<b>39</b>
2.2.1 Fase 1: Análisis de requerimientos.	40
2.2.2 Fase 2: Determinación de proveedor de servicios.	40
2.2.3 Fase 3: Diseño de la arquitectura	41

2.2.4 Fase 4: Despliegue	41
<b>2.3 RESULTADOS</b>	<b>42</b>
2.3.1 Fase 1: requerimientos para el diseño del Laboratorio Virtual de programación	42
2.3.2 Fase 2: Hostinger como proveedor de servicios Capacidad técnica	44
Características de los planes compartidos de Hostinger:	44
2.3.3 Fase 3: Arquitectura de VPL en la nube sobre Moodle	46
2.3.4 Fase 4: Puesta en marcha del Laboratorio Virtual de Programación (VPL) en la nube	49
<b>2.4 CONCLUSIONES</b>	<b>72</b>
<b>2.5 REFERENCIAS BIBLIOGRAFICAS</b>	<b>73</b>
<b>CAPÍTULO III</b>	
<b>CONFIGURACIÓN DEL LABORATORIO VIRTUAL DE PROGRAMACIÓN</b>	<b>75</b>
<b>3.1 PRESENTACIÓN</b>	<b>76</b>
<b>3.2 CONFIGURACIÓN DEL CURSO EN MOODLE</b>	<b>77</b>
<b>3.3 ESQUEMA DE EJERCICIOS PARA EL CURSO</b>	<b>81</b>
3.3.1 Configuración de ejercicios (desde el montaje del docente-configuración)	83
3.3.2 Presentación del caso de prueba (temáticas, tiempo, grupos, ejercicios, requerimientos)	89
<b>3.4 RECOMENDACIONES PARA EL USO DEL VPL</b>	<b>92</b>
3.4.1 Perspectiva del docente	93
3.4.2 Perspectiva del estudiante	93
<b>3.5 CONCLUSIONES</b>	<b>95</b>
<b>3.6 REFERENCIAS BIBLIOGRÁFICAS</b>	<b>96</b>

CAPÍTULO IV	
CASO ESTUDIANTES INGENIERÍA DE SISTEMAS	97
4.1 PRESENTACIÓN	98
4.2 POBLACIÓN	99
4.3 PRESENTACIÓN DEL CASO DE PRUEBA	99
4.3.1 Temática	99
4.3.2 Tiempos de trabajo	101
4.3.3 Análisis de los ejercicios enviados en la práctica del Laboratorio Virtual de Programación	101
4.4 PERCEPCIÓN DE LOS ESTUDIANTES RESPECTO A LA INTERACCIÓN CON EL LABORATORIO VIRTUAL DE PROGRAMACIÓN A PARTIR DE LOS CASOS DE PRUEBA.	107
4.4.1 Interacción con la plataforma.	107
4.4.2 Apoyo al proceso de aprendizaje	113
4.4.2 Aspectos Adicionales	116
4.5 CONCLUSIONES	120
4.6 REFERENCIAS BIBLIOGRÁFICAS	121

# *Lista de Tablas*

Tabla 1. Descripción de la asignatura Lógica y Algoritmos I.	25
Tabla 2. Requerimientos	40
Tabla 3. Factores para comparar prestaciones por proveedor de servicios	41
Tabla 4. Herramientas necesarias para el VPL	42
Tabla 5. Características de los planes	45
Tabla 6. Problemas encontrados en las versiones de Moodle	48
Tabla 7. Esquema para la formulación de ejercicios	82
Tabla 8. Ejercicios introductorios	103
Tabla 9. Ejercicios secuenciales	104
Tabla 10. Ejercicios de estructuras condicionales.	105
Tabla 11. Ejercicios de estructuras cíclicas	105
Tabla 12. Número de ejercicios por unidad	106

# *Lista de Figuras*

Figura 1. Niveles de desempeño en el espacio académico Lógica y Algoritmos I entre 2013-I y 2021-I.	22
Figura 2. Estructura Docker Compose	33
Figura 3. Fases metodológicas para el diseño y despliegue del Laboratorio Virtual de Programación en la nube.	39
Figura 4. Funcionalidades del VPL	42
Figura 5. Características del proveedor elegido.	46
Figura 6. Pasos para implementar Moodle V2.6.1.	47
Figura 7. Pasos para implementar Moodle V8.4.3.	48
Figura 8. Ingreso de credenciales de Hostinger.	49
Figura 9. Inicio de sesión a AWS.	49
Figura 10. Instancias EC2 en el servidor AWS.	50
Figura 11. Visualización de los cursos en el servidor.	50
Figura 12. Iniciar sesión con las credenciales.	51
Figura 13. Llave ssh alojada en el computador.	51
Figura 14. Cargar archivo ssh.	52
Figura 15. Conectarse a la instancia.	52
Figura 16. Código de la llave ssh en la terminal.	53
Figura 17. Backups correspondientes.	54
Figura 18. Copia de la llave ssh.	54
Figura 19. Estructura del Docker.	55
Figura 20. Binding.	55

Figura 21. Estructura del volumen.	56
Figura 22. Copias de seguridad de la plataforma Moodle y la base de datos.	56
Figura 23. Servidor VPS.	57
Figura 24. Administración clave ssh.	57
Figura 25. Agregar clave ssh al servidor Hostinger.	58
Figura 26. Generar la llave ssh.	58
Figura 27. Agregar la llave ssh.	59
Figura 28. Petición al servidor.	59
Figura 29. Agregar llave ssh.	60
Figura 30. Conexión por ssh.	60
Figura 31. Conexión al VPS de Hostinger.	61
Figura 32. Actualización de referencias.	61
Figura 33. Descarga de Docker Compose.	61
Figura 34. Ejecutar Docker.	61
Figura 35. Instalación de Docker.	62
Figura 36. Levantamiento de los servicios.	62
Figura 37. Crear estructura de cero en Hostinger.	63
Figura 38. Conexión al servidor de Hostinger.	63
Figura 39. Descargar contenedores.	64
Figura 40. Error por puerto ocupado.	64
Figura 41. Consulta de los puertos ocupados.	65
Figura 42. Servicios activos.	65
Figura 43. Instalar nuevamente la arquitectura.	66
Figura 44. Visualización de la plataforma Moodle de cero.	66
Figura 45. Puerto ocupado.	66
Figura 46. Descargar nueva copia de seguridad.	67

Figura 47. Comprimir copia de seguridad.	67
Figura 48. Descarga de fichero almacenado en AWS.	67
Figura 49. Subir información a Hostinger.	68
Figura 50. Transferir información de Maria DB a Hostinger.	68
Figura 51. Error al transferir información.	68
Figura 52. Eliminar el contenido del volumen.	69
Figura 53. Copias realizadas correctamente.	69
Figura 54. Despliegue del VPL realizado correctamente.	70
Figura 55. Comandos principales.	70
Figura 56. Creación nuevo curso.	77
Figura 57. Formulario de configuración.	78
Figura 58. Restaurar curso.	78
Figura 59. Importar backup.	79
Figura 60. Detalles del backup.	79
Figura 61. Restaurar nuevo curso.	80
Figura 62. Esquema de los ejercicios.	81
Figura 63. Activar edición.	83
Figura 64. Agregar actividad o recurso.	84
Figura 65. Agregar Actividad.	84
Figura 66. Creación de un ejercicio de Laboratorio virtual de programación.	85
Figura 67. Configuración del periodo.	85
Figura 68. Restricción de envíos.	86
Figura 69. Opciones de configuración.	86
Figura 70. Opciones de configuración.	87
Figura 71. Creación de casos de prueba.	88
Figura 72. Visualización del ejercicio en el editor.	90

Figura 73. Ejemplos de Entrada – Salida.	90
Figura 74. Ejemplo caso de prueba.	91
Figura 75. Solución de salida.	91
Figura 76. Segundo caso de prueba.	92
Figura 77. Entrada -salida caso de prueba 2.	92
Figura 78. Datos de la entrega.	93
Figura 79. Verificación de la solución	94
Figura 80. Envíos realizados	102
Figura 81. Información del estudiante relacionado con los envíos	103
Figura 82. ¿Considera sencillo el ingreso a la plataforma?	108
Figura 83. ¿El acceso a los ejercicios está disponible de forma adecuada?	108
Figura 84. ¿Es clara la forma de subir el archivo del código desarrollado en el IDE?	109
Figura 85. ¿Es clara la forma de editar el código de la solución al ejercicio propuesto?	110
Figura 86. ¿Es clara la forma en cómo se debe enviar los ejercicios?	111
Figura 87. ¿Comprende el esquema entrada – salida utilizado para la presentación de los enunciados?	112
Figura 88. ¿Comprende el resultado que arroja la plataforma en el momento de enviar los ejercicios a evaluar?	113
Figura 89. ¿Considera que la plataforma le facilita el aprendizaje de un lenguaje de programación?	114
Figura 90. ¿Considera que la plataforma facilitaría el proceso de autoaprendizaje en la línea de programación de computadores?	115
Figura 91. ¿Le gustaría seguir utilizando la plataforma para mejorar las habilidades en la construcción de algoritmos?	116

Figura 92. ¿Comprende la intención que tiene la plataforma?	117
Figura 93. ¿Piensa que podrá utilizar esta plataforma sin requerir capacitación adicional?	118
Figura 94. ¿Considera adecuado el uso de la plataforma VPL en las asignaturas e la línea de programación?	119

## **PRESENTACIÓN**

Esta obra es producto del proyecto de investigación denominado “Estrategia didáctica mediada por las TIC para contribuir al fortalecimiento de las competencias en lógica algorítmica de los estudiantes de la Facultad de Ingeniería de la Universidad de la Amazonia”, avalado por el Comité de Investigaciones de la Universidad de la Amazonia en Acta No. 002 del 31 de marzo de 2020. Su formulación surge como una propuesta para dar cumplimiento de forma parcial al tercer objetivo del proyecto de investigación: “Diseñar la estrategia didáctica mediada por las TIC para el fortalecimiento de las competencias en lógica algorítmica”, teniendo en cuenta que la implementación del Laboratorio Virtual de programación integrado a Moodle, ofrece un entorno de práctica interactivo enriquecido por las diferentes posibilidades que ofrecen estas dos herramientas en cuanto a la gestión, evaluación y seguridad de un ambiente de aprendizaje con acceso a recursos y materiales ajustados a las necesidades de formación, centradas en el estudiante y con muy buenos resultados en diferentes instituciones en las que se ha implementado.

Para mostrar de forma general el proceso de análisis e implementación del Laboratorio Virtual de programación, esta obra se estructura en cuatro capítulos: en el primero se realiza una descripción de la problemática identificada con respecto al aprendizaje de los fundamentos de programación, los objetivos de la propuesta de intervención y el análisis de cómo se ha presentado la misma, para esto, se llevó a cabo un abordaje teórico que permite la introducción del lector a los conceptos y herramientas utilizadas durante el proceso de análisis, diseño e implementación en el caso de estudio con los estudiantes de la asignatura Lógica y Algoritmos I del programa Ingeniería de Sistemas Universidad de la Amazonia.

En el segundo capítulo se identifican los aspectos técnicos que permitieron realizar el diseño y despliegue en la nube del Laboratorio Virtual de Programación, a partir de la ejecución de cuatro fases metodológicas, las cuales consistieron en: Fase 1: Análisis de requerimientos, Fase 2: Determinación de

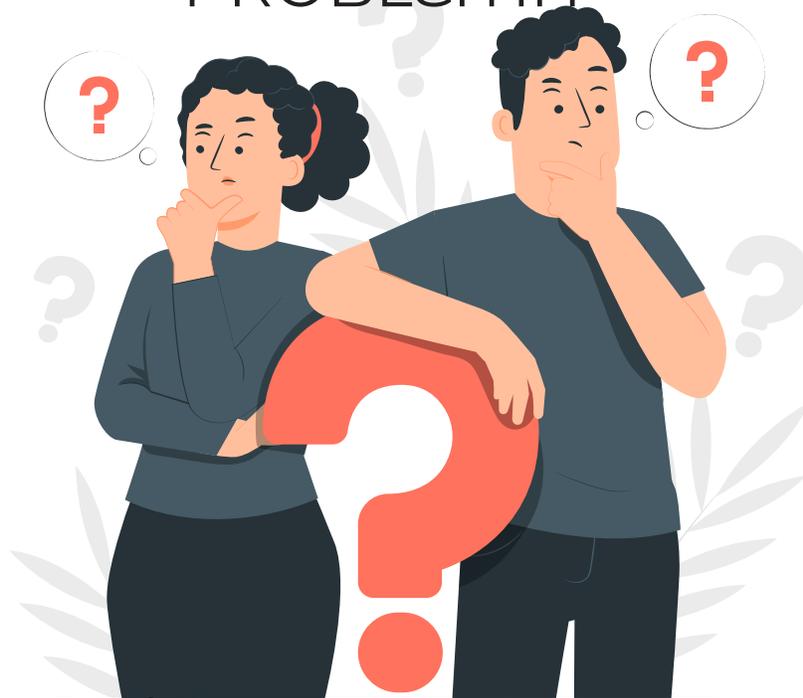
proveedor de servicios., Fase 3: Diseño de la arquitectura y Fase 4: Despliegue. Como resultado se logra contar con el laboratorio funcional para ser configurado y puesto en marcha de acuerdo con los propósitos de formación esperados, cuyo proceso se describen en los siguientes capítulos.

En el capítulo tres se aborda el proceso para la implementación del curso en Moodle y el paso a paso seguido para configurar ejercicios tipo Laboratorio Virtual, apoyados en el esquema Entada-Proceso-Salida definido por los docentes en la asignatura de Lógica y Algoritmos I. En este sentido, se estableció el modelo de cada ejercicio a partir de los elementos requeridos por el módulo Virtual Programming Lab para Moodle (VPL), así como detalles que le permitan al estudiante comprender los enunciados propuestos y plantear una solución correcta. A manera de conclusión se realizan algunas recomendaciones para el uso del VPL desde las perspectivas del docente que propone los ejercicios y de los estudiantes que hacen uso del laboratorio.

El capítulo cuatro presenta la descripción de la implementación del Laboratorio Virtual de programación con el curso de Lógica y Algoritmos I del primer semestre en el programa de Ingeniería de Sistemas de la Universidad de la Amazonia. En este apartado se describe la población intervenida y se presentan los diferentes ejercicios de prueba creados en el laboratorio, los cuales se encuentran distribuidos entre los temas de estructuras secuenciales, condicionales y cíclicas. Así mismo, se indican los tiempos de trabajo contemplados en la intervención, como también el análisis de los ejercicios enviados en la plataforma. Finalmente, se expone la percepción de los estudiantes con relación al uso del Laboratorio de Programación como herramienta tecnológica de apoyo al fortalecimiento de las habilidades de programación de computadores.

# CAPÍTULO I

## PLANTEAMIENTO DEL PROBLEMA



### **John Arley García Quintero**

Magister en E-learning y Redes Sociales. Universidad Internacional de la Rioja. Estudiante de Doctorado en Ciencias de la Educación con Énfasis en: Investigación, Evaluación y Formulación de Proyectos Educativos de la Universidad Metropolitana de Educación, Ciencia y Tecnología (UMECIT). Profesor categoría Asociado, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-5530-1746>

### **Diana María Espinosa Sarmiento**

Magister en Tecnologías de la Información y las Comunicaciones. Universidad de la Amazonia. Profesor categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-1873-0808>

### **Denis Lorena Álvarez Guayara**

Magister en proyectos mediados por TIC. Universidad de la Sabana. Estudiante del Doctorado en Educación y Cultura Ambiental, Universidad de la Amazonia. Profesora categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0001-8939-8139>

CITACIÓN SUGERIDA:

García Quintero, J, A. Espinosa Sarmiento D, M. Álvarez Guayara, D, L. (2024). Planteamiento del problema. Laboratorio virtual para el aprendizaje de fundamentos de programación. Editorial. Universidad de la Amazonia. p.p 121

## **1.1 INTRODUCCIÓN**

En este capítulo se presenta de manera general la problemática identificada respecto al aprendizaje de los fundamentos de programación, a partir de una revisión del contexto y de literatura sobre estudios y experiencias relacionadas con el tema, que sirven de fundamento para sustentar los precedentes y características que permitirán desarrollar la investigación y propuesta de implementación del Laboratorio Virtual de Programación, en el marco del proyecto de investigación “Estrategia didáctica mediada por las TIC para contribuir al fortalecimiento de las competencias en Lógica algorítmica de los estudiantes de la Facultad de Ingeniería de la Universidad de la Amazonia”, desarrollado entre los años 2020 y 2022, de igual forma se presentan los referentes que permiten la familiarización de los conceptos y herramientas utilizadas para su aplicación en el Caso de estudio de la asignatura Lógica y Algoritmos I del programa Ingeniería de Sistemas de la Universidad de la Amazonia.

## **1.2 OBJETIVOS DEL ESTUDIO**

### ***1.2.1 General***

Implementar el Laboratorio Virtual de Programación para el aprendizaje de fundamentos de programación de computadores, dirigido a estudiantes de la Facultad de Ingeniería de la Universidad de la Amazonia

### ***1.2.2 Específicos***

- Desplegar el Laboratorio Virtual en la nube para el aprendizaje de fundamentos de programación.
- Configurar el Laboratorio Virtual de Programación con el paquete de actividades y ejercicios aplicados en la asignatura de Lógica y Algoritmos.
- Identificar desde la percepción de los estudiantes, alcances y limitaciones de la implementación del Laboratorio Virtual de Programación en el caso aplicado al curso de Lógica y Algoritmos I.

### **1.3 CONTEXTUALIZACIÓN SOBRE LOS PROBLEMAS DEL APRENDIZAJE DE LOS FUNDAMENTOS DE PROGRAMACIÓN**

En la actualidad saber programar es una de las competencias más requeridas en las diferentes disciplinas de formación técnica y profesional, sin embargo, abordar desde los escenarios educativos su enseñanza, y que ésta se refleje en un aprendizaje efectivo, es un verdadero reto para las instituciones educativas, profesores y estudiantes. Bajo esta premisa, diferentes universidades nacionales e internacionales, concretamente en carreras de Ingeniería relacionadas con la programación de computadores, han llevado a cabo investigaciones orientadas a la implementación y evaluación de estrategias para el desarrollo del pensamiento computacional en los estudiantes, con el propósito de facilitar la apropiación de habilidades y capacidades para pensar de forma abstracta, analizando las problemáticas del entorno, dividiéndolas en partes más pequeñas y así poder resolverlas, lo cual exige una combinación de pensamiento matemático y pensamiento de ingeniería (Cardoso et al., 2021; Hsu et al., 2018).

Al realizar el estudio sobre los aspectos que afectan el proceso de enseñanza y de aprendizaje de los fundamentos de programación de computadores, se destaca en primera medida la complejidad de analizar problemas, resolverlos y hacerlos comprensibles para la máquina, lo cual resulta complejo para los estudiantes, ya que no se trata solamente de escribir secuencias de código y ejecutarlo, sino también implica identificar las estructuras lógicas que se deben expresar en lenguajes de programación que no solo presenten el resultado deseado, sino que también sean eficientes y optimicen los recursos de hardware para su ejecución (Villamar Coloma, 2021); sumado a esto, los estudiantes demuestran desinterés por los cursos de programación, principalmente porque consideran que los temas son complicados y requieren mucho estudio, lo cual reafirma las problemáticas asociadas a las dificultades en el aprendizaje, bajo rendimiento y reprobación del área (Claros & Sánchez, 2018; López & Gutiérrez, 2021; Zatarain Cabada, 2018); además se ha evidenciado que la deserción de estudiantes se presenta principalmente en los primeros semestres dado que ahí es donde se ubican los cursos introductorios a la programación de computadores (Teixeira et al., 2016).

De acuerdo con el Ministerio de las TIC<sup>1</sup>, para el año 2025 existirá una demanda entre 68.000 y 112.000 técnicos y profesionales con formación en desarrollo de software y programación considerando los requerimientos actuales del mercado digital. Desde esta necesidad evidente de talento humano, es importante también tener en cuenta cuáles son aquellas competencias y habilidades demandadas por las empresas desarrolladoras de software, que no se limitan a la creación y diseño de aplicaciones, sino que implican la creatividad y contar con capacidad para innovar, trabajar en equipo y actuar bajo presión (Lázaro Carrillo et al., 2016).

## **1.4 CASO PROGRAMA INGENIERÍA DE SISTEMAS UNIVERSIDAD DE LA AMAZONIA**

En la Universidad de la Amazonia, los programas de la Facultad de Ingeniería, en su núcleo común, incluyen en sus planes de estudio asignaturas denominadas Lógica y Algoritmos I y II. Estos espacios académicos tienen como propósito abordar los fundamentos de programación de computadores, necesarios para el desarrollo de las competencias y habilidades profesionales y técnicas requeridas para la formación de ingenieros (López et al., 2020).

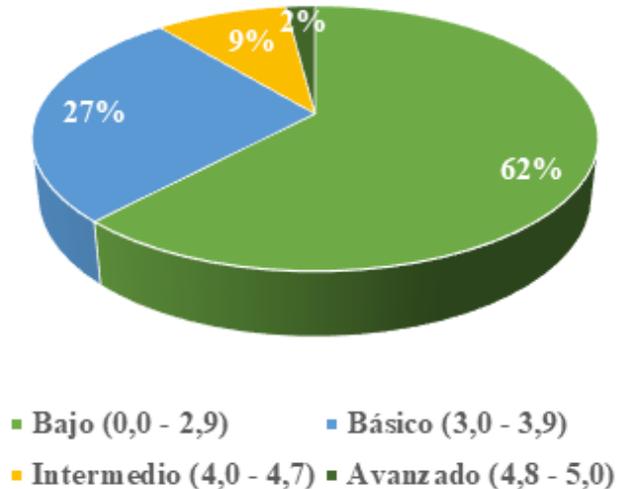
En el caso específico del programa de Ingeniería de Sistemas, estos espacios presentan los rendimientos académicos más bajos, y a pesar de ser una constante en muchas universidades del país, la situación particular en la Universidad de la Amazonia ha generado alerta, ya que los niveles de desempeño tal como se muestra en la Figura 1, entre el primer periodo de 2013 y el primer periodo de 2021, es bajo para el 62% de los estudiantes, lo cual indica que han reprobado la asignatura con notas inferiores a 3, seguidos por el 27% en nivel básico, con notas entre 3 y menos de 4, luego el 9% en intermedio con notas entre 4 y 4.7, por último, el nivel avanzado con un 2%, siendo esta última poco representativa.

Históricamente ha sido evidente que el desempeño de los estudiantes en la asignatura de Lógica y Algoritmos I ha sido principalmente bajo con notas inferiores a 3 (ver Figura 1), aunque también deben considerarse aspectos como la deserción, la cual generalmente es mayor durante los tres primeros semestres y tiene una alta representatividad con relación al desempeño académico bajo.

<sup>1</sup> Cien mil colombianos serán formados en programación digital y tecnología: Ministerio de las TIC - Cien mil colombianos serán formados en programación digital y tecnología: Ministerio de las TIC. MINTIC Colombia. <https://www.mintic.gov.co/portal/inicio/Sala-de-prensa/MinTIC-en-los-medios/149214:Cien-mil-colombianos-seran-formados-en-programacion-digital-y-tecnologia-Ministerio-de-las-TIC>

### Figura 1.

Niveles de desempeño en el espacio académico Lógica y Algoritmos I entre 2013-I y 2021-I.



Fuente: Sistema Misional Chaira.

#### 1.4.1 La enseñanza y el aprendizaje de los fundamentos de programación en Ingeniería de Sistemas.

Dada la modalidad presencial en la cual se desarrolla el programa de Ingeniería de Sistemas en la Universidad de la Amazonia, las clases de la asignatura Lógica y Algoritmos I se orientan en dos encuentros presenciales en la semana con intensidad horaria de dos horas cada uno, las estrategias de enseñanza y aprendizaje realizadas incluyen: presentaciones magistrales, realizadas por el docente para la explicación de los aspectos teóricos del tema además de ejercicios, talleres y evaluaciones que permiten visualizar el avance de los estudiantes en términos de contenidos, competencias y resultados de aprendizaje esperados.

Tal vez uno de los aspectos que puede garantizar el aprendizaje efectivo de la programación es la realización constante de ejercicios, en este aspecto se hace mucho énfasis en el desarrollo del espacio académico y los talleres propuestos en las clases consisten precisamente en la ejecución de dicho componente práctico que permite afianzar el aprendizaje, preferiblemente con una retroalimentación inmediata; sin embargo, resulta demasiado complejo lograr realizar esta actividad de forma individual a todos los estudiantes en los tiempos establecidos, tanto en el desarrollo de los talleres como en las evaluaciones.

Otra estrategia que se ha venido implementando en el programa es el Semillero de Programación, creado en el año 2015, para preparar a los estudiantes con miras a participar en las maratones de programación competitiva a nivel local, nacional e internacional. A través de prácticas y sesiones de competencia, los maratonistas profundizan los fundamentos de programación, desarrollan habilidades para la resolución de problemas, en un segundo idioma y trabajo colaborativo. Los ejercicios propuestos se presentan en inglés y español, y se validan usando plataformas automatizadas como:

- BOCA: Software orientado a la web para la administración y realización de maratones de programación, utiliza las reglas del Concurso de Programación Colegiada Internacional (ICPC). Esta plataforma es empleada semestralmente para la realización de la competencia interna del programa, y suministrada por la liga RPC (Red de Programación Competitiva).
- Codeforces - <https://codeforces.com>: Este sitio es mantenido por un grupo de maratonistas de la Universidad ITMO y cuenta con más de 600.000 usuarios activos en el mundo, entre los cuales se cuenta el campeón mundial Gennady Korotkevich (Tourist).
- Virtual Judge- <https://vjudge.net/>: Aunque no es realmente un juez en línea, es muy útil al permitir la creación de competencias vinculando ejercicios de más de cincuenta plataformas, así como la conformación de grupos de trabajo.
- A2 Judge - <https://a2oj.com>: Este sitio permite organizar grupos y competencias para sus integrantes con estatus detallado del rendimiento de cada uno. Facilita la búsqueda de ejercicios, pues los agrupa por categorías según las temáticas usuales en estas competencias.
- UVA onlineJudge - <https://onlinejudge.org>: ahora Online Judge. Es una de las plataformas más populares entre los maratonistas a nivel mundial, cuenta con un amplio repositorio de ejercicios que comprende las finales mundiales desde el año 1990, los ejercicios de las 3 versiones de los libros Competitive Programming de Steven & Felix Halim y una funcionalidad para realizar depuración de los problemas.
- UFPS Training Center - <https://trainingcenter.cloud.ufps.edu.co/>: Es una plataforma para la enseñanza y entrenamiento de programación competi-

va, desarrollada por estudiantes del programa de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander en el marco de las actividades del Semillero Silux; la plataforma ofrece material de aprendizaje y un juez evaluador de código para entrenamientos en tiempo real.

### ***1.4.2 Lógica y Algoritmos I: Competencias y Resultados De Aprendizaje***

Los fundamentos de programación en el programa de Ingeniería de Sistemas se abordan en la asignatura Lógica y Algoritmos I, esta se encuentra ubicada en el primer semestre, la cual comprende tres (3) créditos académicos, equivalentes a 144 horas, distribuidas en trabajo presencial (32 h) y trabajo dirigido (32 h), estos corresponden a las horas en las que el docente acompaña de forma directa al estudiante en el aula o en la realización de actividades que requieren su acompañamiento, y el trabajo independiente (80 h), destinado para que el estudiante desarrolle otras prácticas necesarias para alcanzar sus metas de aprendizaje de forma autónoma, durante las 16 semanas del periodo académico.

En la Tabla 1, se presentan las competencias, resultados de aprendizaje y contenidos de la asignatura Lógica y Algoritmos I, la cual se encuentra incluida en el plan de estudios vigente del programa ingeniería de sistemas aprobado en el Acuerdo no. 44 de 2020, expedido por el Consejo Académico de la Universidad de la Amazonia.

**Tabla 1.***Descripción de la asignatura Lógica y Algoritmos I.*

<b>ASIGNATURA:</b>	Lógica y Algoritmos I	
<b>COMPETENCIA GENERAL</b>	Construye programas de computador mediante el uso de estructuras de control para la solución a problemas del mundo real.	
<b>COMPETENCIAS ESPECÍFICAS</b>	<b>RESULTADOS DE APRENDIZAJE</b>	<b>CONTENIDOS</b>
Identifica las partes del algoritmo haciendo uso del pensamiento sistémico para plantear soluciones a una situación problema	Hacer uso de los elementos que le permiten representar información en un algoritmo computacional utilizando el esquema Entrada - proceso - salida	<b>UNIDAD 1: Introducción a los Algoritmos</b> <ul style="list-style-type: none"> <li>• Tipos de datos: caracteres, booleanos, enteros, reales.</li> <li>• Operadores y expresiones</li> <li>• Variables - Constantes</li> <li>• Algoritmos y su representación en un lenguaje de programación</li> </ul>
Emplea estructuras condicionales a partir del análisis de situaciones problema para la construcción de soluciones algorítmicas	Implementar soluciones algorítmicas usando las estructuras condicionales simples, dobles, anidadas o múltiples.	<b>UNIDAD 2: Estructuras Condicionales</b> <ul style="list-style-type: none"> <li>• Condicional simple- doble</li> <li>• Condicional anidado</li> <li>• Condicional múltiple</li> </ul>
Implementa ciclos a partir del análisis de situaciones que requieren la iteración de conjuntos de sentencias para la elaboración de programas de computador	Crear soluciones algorítmicas haciendo uso de las diferentes estructuras que permiten implementar iteraciones	<b>UNIDAD 3: Estructuras Iterativas</b> <ul style="list-style-type: none"> <li>• Estructura Haga Para</li> <li>• Estructura Haga Mientras</li> </ul>
Plantea el uso de funciones y procedimientos como estrategia para elaborar algoritmos	Identifica los elementos de las funciones y procedimientos, para construir algoritmos estructurados, haciendo uso de las opciones provistas por el lenguaje de programación o elaborando las propias.	<b>UNIDAD 4: Funciones y procedimientos</b> <ul style="list-style-type: none"> <li>• Concepto, parámetros y retorno</li> <li>• Uso de funciones del lenguaje de programación</li> <li>• Implementación de funciones propias</li> </ul>
Construye soluciones algorítmicas con almacenamiento de datos utilizando arreglos	Implementar soluciones codificadas en un lenguaje de programación con almacenamiento temporal de datos utilizando arreglos	<b>UNIDAD 5: Arreglos Unidimensionales y Bidimensionales</b> <ul style="list-style-type: none"> <li>• Trabajando Vectores</li> <li>• Trabajando Matrices</li> </ul>

Fuente: Microcurrículo Lógica y Algoritmos I.

## **1.5 MARCO REFERENCIAL**

Para el desarrollo de la investigación se consideraron los referentes teóricos y conceptuales identificados en la revisión de literatura, estos permitieron comprender las problemáticas relacionadas con la enseñanza y aprendizaje de los fundamentos de programación, y servir de base para el diseño de la propuesta de intervención del Laboratorio Virtual de Programación y vislumbrar los retos y desafíos de su adecuada implementación.

### ***1.5.1 Antecedentes***

Diversas instituciones de educación superior y programas académicos han planteado estudios orientados al diseño de estrategias para la formación de los estudiantes en el área de programación de computadores, mediante la innovación en los procesos de enseñanza y aprendizaje. Con el propósito de identificar tendencias y características sobresalientes de las investigaciones en el tema, se realizó una revisión de literatura de la cual, enseguida, se presentan los aspectos más sobresalientes:

#### ***1.5.1.1 El uso de entornos y herramientas virtuales de aprendizaje.***

Antes de las restricciones de movilidad derivadas del Covid-19 y que tuvieron una aplicación mundial, el uso de plataformas y mediaciones tecnológicas en los programas de formación a nivel universitario, se limitaban principalmente a los programas de educación a distancia y virtual, en la actualidad, después de superada la pandemia, las condiciones para el uso de TIC en los diferentes niveles y modalidades de formación es más abierta, aprovechando significativamente los entornos educativos mediados por las TIC para enriquecer las prácticas educativas (Barráez, 2020), en este sentido se destaca la implementación de aulas virtuales, diseñadas en plataformas como Moodle, en los que se combinan contenidos, material digital, actividades de aprendizaje, foros y evaluación, que configuran ambientes de aprendizaje flexibles y con adaptaciones para favorecer la colaboración y el autoaprendizaje; también se destacan el uso de entornos gamificados, video juegos y sistemas tutores inteligentes. (López et al., 2020; López & Gutiérrez, 2021; Zatarain Cabada, 2018).

### ***1.5.1.2 Diseños metodológicos.***

Se podrían considerar como propuestas basadas en fundamentos pedagógicos que tienen como propósito abarcar diferentes perspectivas de la enseñanza y del aprendizaje, en las cuales se reconoce el papel activo del estudiante y el rol del profesor como orientador y acompañante del proceso de formación, además se suma la integración de plataformas tecnológicas como un elemento mediador, el cual provee diferentes escenarios y herramientas que transforman la práctica educativa (Checa Fernández, 2011; Lázaro Carrillo et al., 2016).

### ***1.5.1.3 Laboratorios virtuales y entrenamiento para maratones de programación.***

El uso de plataformas de entrenamiento de programación proporciona entornos colaborativos de trabajo que facilitan el monitoreo, la retroalimentación y la interactividad. Así mismo, es importante resaltar que una de las características de los laboratorios virtuales de programación más sobresaliente es la calificación automática, la cual, en una fase inicial, implica un esfuerzo adelantado por parte de los profesores en las tareas de parametrización de las tareas, entornos de prueba y validación. Sin embargo, una vez ya se ha culminado esta etapa, posteriormente el esfuerzo es mínimo una vez finalice la fecha límite de la asignación y la labor del docente se podrá enfocar hacia la retroalimentación del proceso (Castañeda et al., 2019; Guerrero et al., 2015; Lázaro Carrillo et al., 2016; Molina et al., 2016; Rojas Machado et al., 2020; Thiébaud, 2015).

Los resultados obtenidos con la implementación de estas estrategias han sido tanto positivos como negativos, pero es gracias a la fundamentación, experimentación y los resultados, que ha sido posible sentar precedentes para el desarrollo de nuevas investigaciones y mejoras a las propuestas de intervención, las que han sido consideradas para el estudio planteado.

## ***1.5.2 Casos de implementación de VPL***

A continuación, se presentan características de algunos estudios sobre la implementación del Laboratorio Virtual de Programación (VPL), en diferentes países e instituciones de educación superior:

### ***1.5.2.1 Laboratorio Virtual de Programación Java.***

En el centro de investigación de Computación ubicado en México D.F se realizó un Laboratorio Virtual de Programación que permite la elaboración de materiales didácticos, y el uso de contenidos multimedia con temática orientada a objetos. El eje central del laboratorio virtual es un compilador en línea de código java utilizado para visualizar los resultados de los ejercicios y una interfaz gráfica amigable con el usuario para mayor posibilidad de aprendizaje, y los estudiantes puedan evaluar los conocimientos comparándolos con los resultados obtenidos (Peredo Valderrama, 2009).

### ***1.5.2.2 VPL Universidad Nacional de Río Negro, Patagonia Argentina.***

La Universidad Nacional de Río Negro creó el VPL como una estrategia de aprendizaje mediante los recursos informáticos, y como introducción a la Lógica y Algoritmos. Fue elaborado con el objetivo que los estudiantes tuvieran habilidades como la programación, la capacidad de comunicación y la ganas de aprender. El VPL se creó debido a los siguientes factores: la complejidad de la resolución de problemas algorítmicos, facilitar la interacción con los participantes del plantel educativo y la formación de profesionales principalmente en ciencias de la computación. También permite realizar prácticas de programación a través del módulo Moodle facilitando el aprendizaje personalizado de cada alumno para resolver las dificultades presentadas en los ejercicios de programación. Además, brinda la posibilidad de ejecutar el código fuente y pruebas para controlar el plagio, y corroborar que los ejercicios sí cumplan con las especificaciones, limitar el tiempo para entregar actividades, edición de archivos y uso de contraseña para su ingreso (Lovos E. , 2015).

### ***1.5.2.3 VPL de la Universidad de Córdoba, España.***

La herramienta permite realizar procesos de aprendizaje de manera virtual y las veces que desee, sin estar sujeto a un horario, y también le permite al docente llevar un control completo de las actividades llevadas a cabo por el alumnado. El VPL permite subir archivos, compilarlos y descargar prácticas para que cada estudiante las realice. Además, cuenta con una verificación de tiempo de acceso para cerciorarse de que el usuario sí sea la persona que está iniciando sesión (Montijano, 2005).

### ***1.5.2.4 VPL con Autoevaluación Habilitada por Tecnología.***

Se desarrolló un laboratorio virtual de programación que proporciona a los estudiantes contenido temático, modelos, videos, simulaciones y cuestionarios para que ellos comprendan conceptos a través del autoaprendizaje. El VPL permite a los estudiantes aprender conceptos y ejercicios sin necesidad de tener un maestro presente en el aula de clase. La idea nace debido a que los recursos didácticos disponibles se han vuelto un recurso importante para que los estudiantes realicen los ejercicios las veces que quieren a través de simulaciones. Cada módulo contiene teoría y videos para que los estudiantes aprendan con el objetivo de reducir el tiempo de aprendizaje de los estudiantes y gráficos interactivos (Upadhyay, 2022).

## ***1.5.3 Aspectos teóricos y conceptuales para el diseño y el despliegue de un VPL***

A continuación, se relacionan los aspectos teóricos y conceptuales que fundamentan la realización del estudio, necesarios para realizar el diseño y despliegue del VPL:

### ***1.5.3.1 Laboratorio Virtual de Programación (VPL).***

El VPL implementado en una plataforma Moodle permite realizar la gestión automática de prácticas y pruebas de prácticas de programación. La implementación en conjunto con la plataforma Moodle es de vital importancia porque tiene la posibilidad de crear actividades que se pueden reutilizar, y crear un repositorio con ejercicios previamente usados, además de brindar al usuario un entorno amigable para personalizar las actividades y realizarlas de una manera más fácil (Rodríguez del Pino et al., 2010).

La herramienta Moodle permite realizar las prácticas de laboratorio en un ambiente colaborativo en los cursos de programación de cualquier nivel y fomentar el interés de los alumnos ya que lo ven como una propuesta novedosa de trabajo. El uso del VPL es sumamente beneficioso para los alumnos que se están iniciando en el mundo de la programación, en donde el acompañamiento inicial y las guías son de vital importancia (Lovos & González, 2014).

Se ha comprobado que la utilización de laboratorios virtuales por parte de los estudiantes contribuye a la mejora de los conocimientos de programación y lógica de estos a través de la realización de ejercicios y la posibilidad de verificarlos cuantas veces quieran y corregir errores, además de ahorrarle trabajo al docente a la hora de calificar ejercicios y pueden invertir el tiempo en dudas de alto interés (Alvarez Martínez, 2012).

Los VPL presentan diferentes características que permiten fortalecer competencias de la Lógica y Algoritmos, gracias a la posibilidad de integración con la plataforma Moodle, esto permite que las entregas se puedan editar y controlar, ejecutar y evaluar el código fuente para que el proceso se realice de forma más rápida, resolver los ejercicios planteados y permite personalizar actividades de acuerdo con las necesidades de los estudiantes (Barrios & Marín, 2014); así mismo, cuenta con múltiples ventajas como: capacidad de enviar los trabajos a la nube, el costo de la implementación va de acuerdo con el consumo que el usuario realice, permite aumentar la capacidad del servidor de acuerdo con la cantidad de usuarios, se puede hacer uso del VPL con el usuario institucional de la Universidad a través de la plataforma Moodle, y los servicios de la nube garantizan que se pueda utilizar en cualquier tiempo, además de permitirle a los estudiantes evaluar si los ejercicios planteados son correctos y corregir los errores en múltiples ocasiones (Lorandi et al., 2011).

### **1.5.3.2 Moodle (*Learning Management System*).**

La plataforma Moodle nace en el 2002 cuando Martin Dougiamas quería construir una herramienta que fuera intuitiva para facilitar el aprendizaje entre todos los integrantes de un equipo. Moodle facilita el auto aprendizaje y permite retroalimentar el trabajo entre todos e incorporar diferentes módulos, a medida que se van viendo temáticas durante la clase. Moodle tiene diversas ventajas: ahorro en papel utilizado en fotocopias del material necesario para clases, se puede implementar herramientas necesarias para que estudiantes con necesidades especiales puedan estudiar, permite estudiar de

manera virtual y llevar a cabo una metodología de manera más activa a diferencia de la usada en el aprendizaje tradicional, además de ser muy fácil de descargar (Martínez de Lahidalga, 2008).

### ***1.5.3.3 Moodle en la enseñanza.***

En el mundo pedagógico la plataforma Moodle brinda grandes ventajas para mejorar la enseñanza buscando que sea más didáctica, además de incentivar a los profesores a capacitarse en el uso de las Tic en el aula de clase. Por esto la plataforma Moodle está construida de forma que sea muy intuitiva para el uso de cualquier persona. La plataforma Moodle permite que la obtención de conocimiento sea de forma personalizable de acuerdo con las necesidades del curso y el ritmo de aprendizaje. La mayor ventaja de la plataforma Moodle en el aprendizaje es que se puede integrar con diferentes herramientas como chats, mensajería, recordatorio de tareas, actividades que se pueden resolver entre todos, seguimiento de actividades y muchos recursos de multimedia (Lázaro, 2010).

### ***1.5.3.4 VPL diseño y despliegue.***

Un Laboratorio Virtual de Programación es una herramienta que permite la realización de ejercicios en la plataforma Moodle a través de un editor de código, y tiene como ventaja que los estudiantes pueden ejecutar los ejercicios de manera remota y la realización de ejercicios desde su computador personal. El VPL es una herramienta en la que se pueden realizar ejercicios de programación una y otra vez para corregir errores cometidos por los estudiantes e identificar situaciones de eventual plagio en el código fuente (Rodríguez del Pino, 2010).

### ***1.5.3.5 VPL Cloud computing.***

La computación en la nube abarca diversos servicios que se ofrecen a través de Internet, la nube tiene a su disposición servicios que los usuarios requieran consumir. La computación en la nube ha sido de vital importancia en las empresas, debido a que les permite distribuir productos digitales sin necesidad de contar con una estructura física, sin necesidad de costos adicionales, puesto que sólo se paga por el servicio consumido (Academia pragma, 2019). La computación en la nube presenta grandes beneficios como: necesitar únicamente la implementación de terminales para su funcionamiento, evitando invertir grandes sumas de dinero para comprar un computador

robusto (servidor), debido a que el consumidor solo utiliza los recursos que necesita y los servicios de procesamiento y almacenamiento lo obtiene simplemente conectándose a la nube. Otra ventaja importante es la disminución del consumo de energía, ya que las computadoras no consumen más de 10 watts en comparación con la energía requerida por las computadoras de escritorio (Ávila Mejía, 2011).

Además, una de las desventajas que presenta la computación en la nube, es que cuando el servicio en la nube falla, son millones de usuarios los que se ven afectados y esto ocasionará la pérdida de confiabilidad en los servicios ofrecidos; una fuga en la información almacenada en la nube puede tener un impacto desastroso, y la disponibilidad de la información personal se reduce al acceso del internet que tenga el usuario.

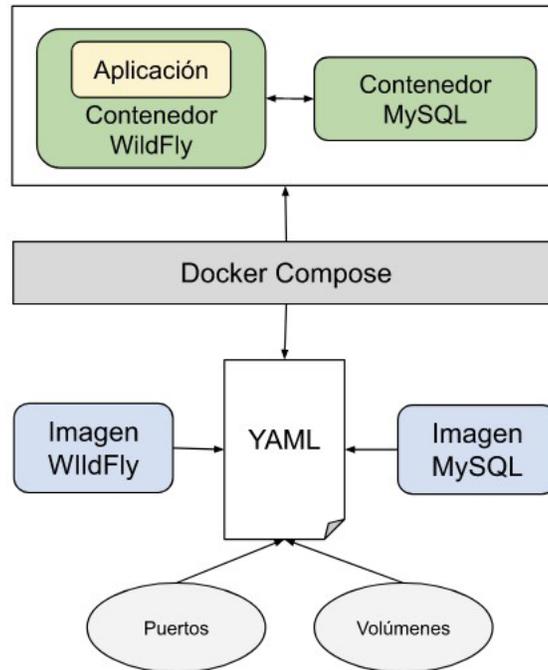
#### **1.5.3.6 Docker.**

Docker es una plataforma que permite compilar y ejecutar aplicaciones en contenedores con el objetivo de que las aplicaciones puedan estar separadas, pero alojadas en el mismo host. Los contenedores son cada vez más usados debido a que permiten que aplicaciones complejas se puedan alojar; los contenedores son más eficientes que las máquinas virtuales; se pueden implementar contenedores en la nube, son seguros y no es necesario que el usuario lo configure para su funcionamiento (Barrios, 2020).

#### **1.5.3.7 Docker compose.**

Docker compose es una herramienta que permite conectar los contenedores más fácilmente, unirlos y realizar aplicaciones con ellos. Con Docker compose ya no es necesario asignar comandos y scripts, pues el Docker engine puede realizar instrucciones automáticamente en diferentes ambientes de trabajo (Corral, 2020). En la Figura 2, se puede visualizar la estructura que compone Docker Compose.

**Figura 2.**  
*Estructura Docker Compose*



Fuente: Tomada de Medina (2021).

Por otro lado, un elemento que se debe tener en cuenta es el volumen, este es un contenedor que permite conservar los datos, aunque se elimine el Docker. Además, permite el intercambio entre el host y el contenedor. Crear un volumen permite transferir, guardar e intercambiar los datos de un Docker container. El volumen es una carpeta que comparten el contenedor y el ordenador host. Un container volumen tiene su propio contenedor en el host y se comporta como una carpeta en la cual se pueden almacenar y recuperar datos (Juanola Alpiste, 2021).

## 1.6 REFERENCIAS BIBLIOGRÁFICAS

- Academia pragma. (2019). Computación en la nube: Todo lo que debe saber para una migración segura. <https://www.pragma.com.co/academia/conceptos/computacion-en-la-nube>
- Alvarez Martínez, C. E. (2012). Laboratorio Virtual Para La Enseñanza De Fundamentos De Programación De Computadores.
- Ávila Mejía, O. (2011). Computación en la nube.
- Barráez, D. P. (2020). La educación a distancia en los procesos educativos: Contribuye significativamente al aprendizaje. *Revista Internacional Tecnológica-Educativa Docentes 2.0*, 8(1), 41–49. <https://ojs.docentes20.com/index.php/revista-docentes20/article/view/91/232>
- Barrios, E. (2020). ¿Qué demonios es Docker y Docker-Compose? y cómo Dockerizar Dotnet Core WebApi y SQL Server en un ambiente de desarrollo ideal. Obtenido de <https://dev.to/ebarrioscode/que-demonios-es-docker-docker-compose-y-como-dockerizar-dotnet-core-webapi-y-sql-server-en-un-ambiente-de-desarrollo-ideal-95a>
- Barrios, T., & Marín, M. B. (2014). Aprendizaje mixto a través de laboratorios virtuales. *signos univeRsitaRios anejo*.
- Cardoso, M., Marques, R., de Castro, A. V., & Rocha, Á. (2021). Using Virtual Programming Lab to improve learning programming: The case of Algorithms and Programming. *Expert Systems*, 38(4), 1–17. <https://doi.org/10.1111/exsy.12531>
- Castañeda, J. A., Ortega Álvarez, C. I., & Galvis Ibarra, J. C. I. (2019). Dockerizando Un Laboratorio Virtual De Programación (Vpl) Y Moodle En Google Cloud. *Encuentro Internacional de Educación En Ingeniería, SE-*. <https://acofipapers.org/index.php/eiei/article/view/212>
- Checa Fernández, R. P. (2011). La innovación metodológica en la enseñanza de la programación. Una aproximación pedagógica al aprendizaje activo en la asignatura Fundamentos de Programación. *Interfases*, 4, 67–87. <https://doi.org/https://doi.org/10.26439/interfases2011.n004.154>
- Claros, F., & Sánchez, C. (2018). Repitencia de los estudiantes de pregrado de la Facultad de Ingeniería en las asignaturas de Lógica y Algoritmos I y II de la Universidad de la Amazonia. Universidad de la Amazonia.

- Corral, P. (2020). Aprendiendo a utilizar Docker Compose. Obtenido de <https://dockertips.com/utilizando-docker-compose>.
- Guerrero, S., Díaz, E., Córdoba, F., & Hernández, G. (2015). Las Maratones de Programación, un paso más al campo de la Investigación. *Revista Tecnológica ESPOL - RTE*, 28(5), 424–434. <http://rte.espol.edu.ec/index.php/tecnologica/article/view/451/316>
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, 126(June), 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Juanola Alpiste, A. (2021). Arquitectura de Servicios con Docker Swarm
- Lázaro, M. R. (2010). moodle, una plataforma formativa con gran proyección en los nuevos modelos de enseñanza. *dim: Didáctica, Innovación y Multimedia*.
- Lázaro Carrillo, G. Y., Delgado León, A. M., & Vera Rivera, F. H. (2016). Desarrollo e implementación de un marco de trabajo para el entrenamiento en programación competitiva. *Universidad Ciencia y Tecnología*, 20(79), 69–74.
- López, M. A., Duarte, E. V., & Gutiérrez, E. C. (2020). Experiencia en la Utilización de un Aula Virtual para la Enseñanza de la Programación de Computadoras: Desarrollo del Pensamiento Algorítmico y Aprendizaje de un Lenguaje de Programación. *Proceedings of the LACCEI International Multi-Conference for Engineering, Education and Technology*, July 2020, 27–31. <https://doi.org/10.18687/LACCEI2020.1.1.37>
- López, M. A., & Gutiérrez, E. C. (2021). Del B-Learning al E-Learning: Experiencia en la Enseñanza de Programación de Computadoras. *Proceedings of the LACCEI International Multi-Conference for Engineering, Education and Technology*, 2021-July, 1–10. <https://doi.org/10.18687/LACCEI2021.1.1.515>
- López, M. A., & Gutiérrez, E. C. (2021). Enseñanza de la Programación Avanzada de Computadoras aplicando un Proyecto de Desarrollo de Software. *Proceedings of the LACCEI International Multi-Conference for Engineering, Education and Technology*, 2021-July, 1–10. <https://doi.org/10.18687/LACCEI2021.1.1.82>

- Lorandi Medina, A. P., Hermida Saba, G., Hernández Silva, J., & Ladrón de Guevara Durán, E. (2011). Los Laboratorios Virtuales y Laboratorios Remotos en la Enseñanza de la Ingeniería. *Revista Internacional de educación en ingeniería*.
- Lovos, E. (2015). Ambiente de desarrollo virtual para el aprendizaje de la programación: un estudio de caso en la Lic. de Sistemas de la Universidad Nacional de Río Negro, Patagonia Argentina. *Revista internacional de aprendizaje en ciencia, matemáticas y tecnología*.
- Lovos, E., & González, A. (2014). Moodle y VPL como Soporte a las Actividades de Laboratorio de un Curso Introductorio de Programación.
- Martinez de Lahidalga, I. R. (2008). Moodle, la plataforma para la enseñanza y organización escolar. *Didáctica de la Expresión Corporal*. Obtenido de <https://addi.ehu.es/bitstream/handle/10810/6876/moodle.pdf?sequence=1&isAllowed=y>
- Medina, D. (2021). Curso Jakarta EE 9 (9). Docker (2): imágenes y contenedores. Obtenido de <https://danielme.com/2021/05/06/curso-jakarta-ee-docker-trabajando-con-imagenes-y-contenedores/>
- Molina, Y., Espinosa, D., & Trujillo, E. (2016). Uso de la programación competitiva como aporte a la enseñanza de la Lógica Algorítmica. *Experiencia Universidad de la Amazonia. Revista Científica, Esp(26)*, 112–122.
- Montijano, J. M. (2005). Laboratorio virtual para la programación de FPGAs .
- Peredo Valderrama, J. C. (2009). Laboratorio Virtual de Programación Java basado en el paradigma de Educación Basada en Web. *SISTEMAS, CIBERNÉTICA E INFORMÁTICA*.
- Rodríguez del Pino , J. C., Rubio Royo, E., & Zenón , H. F. (2010). VPL: Laboratorio Virtual de Programación para Moodle. *VPL: laboratorio virtual de programación para Moodle*.
- Rojas Machado, N., Barrero Fiu, E. E., Morales Viera, J. J., Capote-Pérez, A. A., Vigo Rodriguez, R., & Rojas Machado, J. (2020). Movimiento de programación competitiva : su aporte a los estudiantes de las ciencias médica. XVIII CONVENCIÓN Y FERIA INTERNACIONAL DE INFORMATICA, 9.
- Teixeira, A. R., Gomes, A., & Orvalho, J. (2016). E-learning multimodal system for teaching and learning programming. *Proceedings of the International Conference on Interfaces and Human Computer Interaction 2016, Game*

and Entertainment Technologies 2016 and Computer Graphics, Visualization, Computer Vision and Image Processing 2016 - Part of the Multi Conference on Compu, 332–335.

Thiébaud, D. (2015). Automatic evaluation of computer programs using Moodle's virtual programming lab (VPL) plug-in. *Journal of Computing Sciences in Colleges*, 30(6), 145–151.

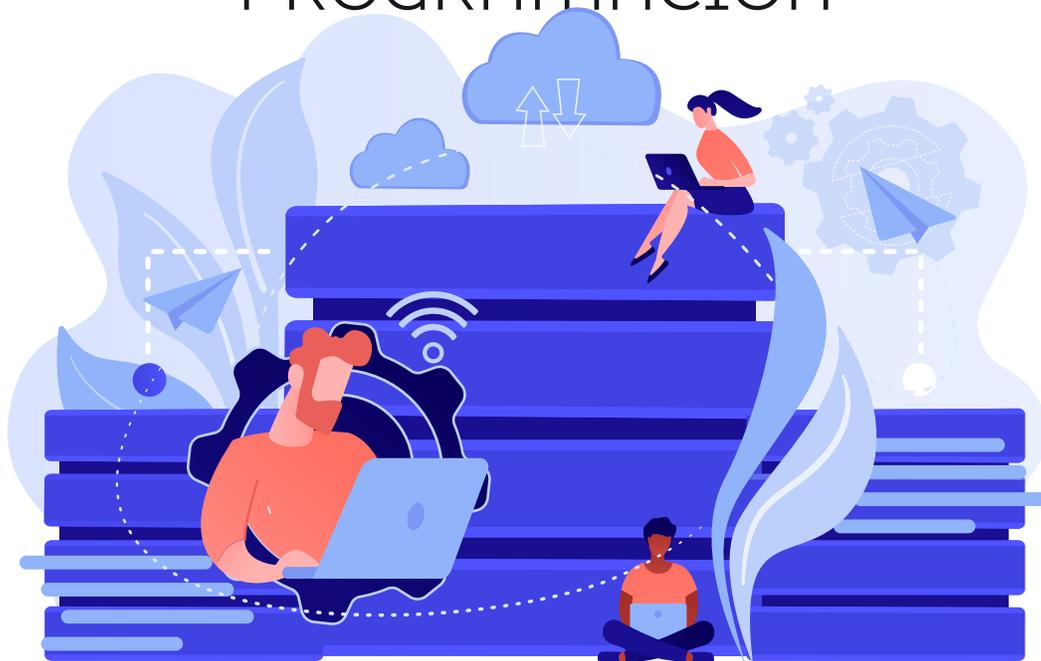
Upadhyay, A. S. (2022). *Diseño y Desarrollo de Laboratorio de Simulación Virtual con Autoevaluación Habilitada por Tecnología*.

Villamar Coloma, M. A. (2021). Metodología ABP para mejorar el aprendizaje en la asignatura Fundamentos de Programación del Instituto Superior Tecnológico Babahoyo, Ecuador – 2020. [https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/68151/Villamar\\_CMA-SD.pdf?sequence=1&isAllowed=y](https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/68151/Villamar_CMA-SD.pdf?sequence=1&isAllowed=y)

Zatarain Cabada, R. (2018). Reconocimiento afectivo y gamificación aplicados al aprendizaje de Lógica algorítmica y programación Affective. *Revista Electronica de Investigacion Educativa*, 20(3), 115–125. [https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054654725&doi=10.24320%-2Friede.2018.20.3.1636&partnerID=40&md5=681c7d13df61ab3af3cda-bea4cd6a878](https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054654725&doi=10.24320%2Friede.2018.20.3.1636&partnerID=40&md5=681c7d13df61ab3af3cda-bea4cd6a878)

# CAPÍTULO II

## DESPLIEGUE EN LA NUBE DEL LABORATORIO VIRTUAL DE PROGRAMACIÓN



### **Denis Lorena Alvarez Guayara**

Magister en proyectos mediados por TIC. Universidad de la Sabana. Estudiante del Doctorado en Educación y Cultura Ambiental, Universidad de la Amazonia. Profesora categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0001-8939-8139>

### **Luisa Fernanda Zuluaga Ramírez**

Estudiante del Programa de Ingeniería de Sistemas Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-1943-9027>

### **Diana María Espinosa Sarmiento**

Magister en Tecnologías de la Información y las Comunicaciones. Universidad de la Amazonia. Profesor categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-1873-0808>

#### CITACIÓN SUGERIDA:

Álvarez Guayara, D, L. Zuluaga Ramírez L, F. Espinosa Sarmiento D, M. (2024). Despliegue en la nube del laboratorio virtual de programación. Planteamiento del problema. Laboratorio virtual para el aprendizaje de fundamentos de programación. Editorial. Universidad de la Amazonia. p.p 121

## 2.1 PRESENTACIÓN

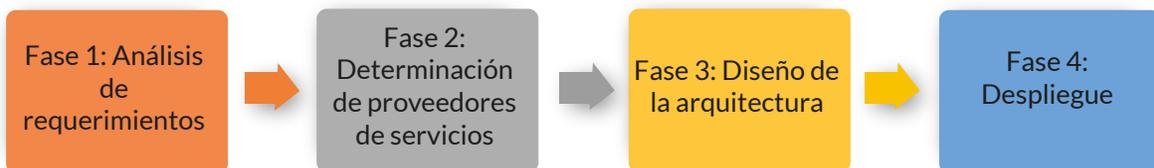
En este capítulo se presenta la descripción de la metodología implementada para el diseño y despliegue del Laboratorio Virtual de Programación (VPL), con base en la determinación de las cuatro fases metodológicas y la identificación de los requerimientos técnicos y tecnológicos propios de la implementación de servicios en la nube, particularmente para el VPL y su implementación en Moodle. Así mismo se muestran los resultados obtenidos a partir de la realización de las actividades planteadas en cada una de las fases y las conclusiones con respecto al análisis sobre el proceso desarrollado para poner en marcha el VPL, para así posteriormente configurarlo a las necesidades particulares que permitan abordar las temáticas relacionadas con los fundamentos de programación, dirigido a estudiantes de ingeniería.

## 2.2 METODOLOGÍA PARA EL DISEÑO Y DESPLIEGUE DEL LABORATORIO VIRTUAL DE PROGRAMACIÓN

Para el diseño y despliegue del Laboratorio Virtual de Programación en la nube se plantearon cuatro fases metodológicas, las cuales se presentan claramente definidas en la Figura 3:

### Figura 3.

*Fases metodológicas para el diseño y despliegue del Laboratorio Virtual de Programación en la nube.*



Fuente: Elaboración propia.

A continuación, se describen cada una de las fases de acuerdo con su propósito y el producto esperado:

### **2.2.1 Fase 1: Análisis de requerimientos.**

En esta fase se determinaron los requerimientos básicos necesarios para el diseño y despliegue del VPL en la plataforma Moodle, para lo cual se tuvo cuenta la comunicación en red y la persistencia de la información. En la Tabla 2, se presentan los requerimientos considerando los aspectos anteriormente descritos.

**Tabla 2.** *Requerimientos*

<b>Requerimiento</b>	<b>Objetivo</b>
Gestor BD	Establecer la base datos necesaria para la gestión de la información
Versión compatible Moodle	Determinar la versión de Moodle Compatible con el VPL Revisar las pruebas realizadas con cada una de las versiones de Moodle.
Jail Server	Implementar el Jail Server en la plataforma Moodle.

Fuente: Elaboración propia.

### **2.2.2 Fase 2: Determinación de proveedor de servicios.**

Para la elección del proveedor de servicios en la nube adecuado a los requerimientos de implementación del VPL, se tuvieron en cuenta algunos factores, sobre los cuales se realizan comparaciones, principalmente, de tipo técnico de tal forma que permitieran visualizar la mejor opción en cuanto a las prestaciones y servicios ofrecidos, en correspondencia con los planes ofertados. En Tabla 3 se hace el comparativo.

**Tabla 3.** Factores para comparar prestaciones por proveedor de servicios

<b>Factor</b>	<b>Hosting sencillo</b>	<b>Hosting Premium</b>	<b>Hosting Empresarial</b>
Cantidad de sitios web	1	100	100
Almacenamiento SSD	30 GB	100 GB	200 GB
Cantidad de Bases de Datos	2	Ilimitada	Ilimitada
Acceso SSH	No	Si	Si
Acceso a GIT	Si	Si	Si
Dominio gratis	No	Si	Si
Ancho de banda	100 GB	Ilimitado	Ilimitado
SSL Gratis	Si	Si	Si
Mail	1 cuenta	Correo gratuito	Correo gratuito

Fuente: Tomada de (Hostinger, 2022)

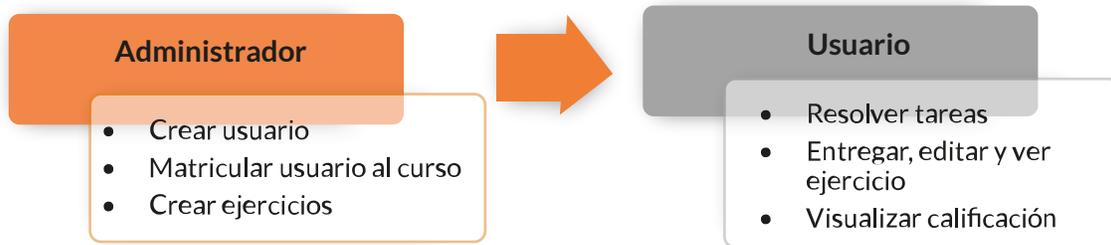
### **2.2.3 Fase 3: Diseño de la arquitectura**

En esta fase se documentan las actividades realizadas para integrar la plataforma Moodle con el VPL-Jail en la Cloud del proveedor de servicios seleccionado. De acuerdo con la documentación, se encontró que las versiones compatibles con el plugin VPL son la versión v2.6.1 y la versión v3.8.4.

### **2.2.4 Fase 4: Despliegue**

En esta etapa se pone en marcha la implementación del VPL en la plataforma Moodle en la nube, desde las funcionalidades del usuario y el administrador, dividido en sus opciones principales como: crear usuario manual, matricular usuario al curso, actividades del curso, realizar entrega y ver calificación. Esto implicó la verificación de las opciones para los usuarios de acuerdo con su rol según se muestra en la Figura 4:

**Figura 4.**  
Funcionalidades del VPL



Fuente: Elaboración propia.

## 2.3 RESULTADOS

De acuerdo con las cuatro fases de desarrollo propuestas en la metodología, se presentan los siguientes resultados:

### 2.3.1 Fase 1: requerimientos para el diseño del Laboratorio Virtual de programación

En el análisis de requerimientos se identificaron las herramientas necesarias en el diseño del VPL:

- Sistema gestor de base de datos
- Plataforma Moodle
- Jail server.

En la Tabla 4, se describen las características analizadas a partir del requerimiento.

**Tabla 4.** Herramientas necesarias para el VPL

Requerimiento	Necesidades de los usuarios	Características funcionales
<b>MariaDB</b>	MariaDB es el gestor recomendado para la plataforma Moodle y ofrece un mejor rendimiento de manera gratuita.	<ul style="list-style-type: none"> <li>-Es más rápida en el momento de realizar consultas complejas.</li> <li>-Puede manejar hasta 32 segmentos clave.</li> <li>-Extenso soporte de ayuda para solucionar errores.</li> <li>-Software libre, se puede utilizar durante todo el proyecto.</li> </ul>

<b>Moodle</b>	En el proyecto se implementaron 2 versiones Moodle. De estas, la que tenía más características y la versión más actualizada de las que eran compatibles con el proyecto fue la versión Moodle v3.8.4.	<ul style="list-style-type: none"> <li>-Interfaz moderna</li> <li>-Tablero personalizado</li> <li>-Actividades y herramientas colaborativas.</li> <li>-Calendario</li> <li>-Modificación de archivos.</li> <li>-Editor de texto</li> <li>-Monitoreo de progreso</li> </ul>
<b>Jail Server</b>	Es un complemento utilizado para que la plataforma Moodle y el VPL sean compatibles entre sí. Ejecuta los scripts sin que el servidor Moodle se vea afectado.	Permite editar código fuente de los programas en el navegador, ejecutar programas interactivos y pruebas de software, buscar similitud entre archivos, establecer restricciones de edición y evitar el copiar y pegar texto desde otras páginas.

Fuente: Tomada de Moodle (2020).

Cabe destacar que la selección del sistema gestor de base de datos MariaDB en MySQL, obedece a su versatilidad, considerando que es un software libre, este permite la incorporación de consultas avanzadas con la posibilidad de implementar clúster para trabajar en la nube y manejar estructuras complejas, además, cuenta con mejoras en seguridad y rendimiento (Hostinplus, 2020), también, tiene comunicación http mediante el puerto 3306 para el consumo de la base de datos. El respaldo de los datos se hace mediante el comando mysqldump.

Por otro lado, la plataforma Moodle sobre la cual se ejecuta el VPL, permite crear y gestionar ambientes de aprendizaje centrada en las necesidades de los estudiantes, que permite su uso 24/7 dado que se aloja en un servidor web (Moodle, 2015), mediante el uso de los puertos 80 y 443 para los protocolos http y https respectivamente, donde el protocolo https es de uso exclusivo para una configuración con certificado SSL (Niño Monje, 2022). Se debe agregar que la carpeta de archivos Moodledata almacena todos los ficheros necesarios para el funcionamiento de Moodle sin necesidad de que sean almacenados en la base de datos como documentos, imágenes y videos. Esta funcionalidad mejora el rendimiento y la gestión de archivos dentro de los cursos (Castañeda et al., 2019).

En cuanto al uso del Jail Server, su principal función es permitir la ejecución de los scripts de prueba en los programas que utilizan los estudiantes. Si el programa se bloquea el Jail Server permite que el servidor Moodle no se vea afectado. Cuando el estudiante envía el programa, el servidor Moodle empaqueta la prueba del instructor junto con el código del instructor y el Jail Server ejecuta los scripts de prueba en un entorno aislado, y la salida va directamente al servidor Moodle con calificación o comentarios (Thiébaud, 2015).

### **2.3.2 Fase 2: *Hostinger como proveedor de servicios***

El proveedor de servicios en la nube elegido para la gestión y almacenamiento de los recursos de computación de la nube fue Hostinger debido principalmente a los siguientes factores:

#### ***Capacidad técnica***

Hostinger es uno de los proveedores de alojamiento más baratos que ofrecen calidad y precio. Es el sitio de almacenamiento más valioso para propietarios de sitios web que no tienen mucha experiencia en el tema. Con el plan de Hostinger Single Shared Hosting de menos de dos dólares, el usuario puede obtener un almacenamiento SSD de 30 GB, transferencia de datos de 100 GB y creador de sitios web. Sin embargo, un usuario que necesite alojar más sitios web, lo puede hacer a través de un dominio gratuito con bases de datos ilimitadas.

#### ***Características de los planes compartidos de Hostinger:***

Los planes compartidos cuentan con mayor velocidad del sitio web, creador de sitios web con ayuda para diseñar plantillas integradas, optimización para mejor rendimiento de WordPress y SSL gratuito para realizar la conexión HTTPS.

#### ***Características de los planes Premium:***

Las características del plan Premium tiene incluidos funciones como: integración GitHub, dominio gratuito, base de datos y cronjobs, acceso a SSH y CDN gratuito (ver Tabla 5).

**Tabla 5.** *Características de los planes*

<b>Proveedor</b>	<b>Características de los planes</b>	<b>Características de servicios empresariales</b>
<b>Google Cloud</b>	Se basa en grupos de instancia gestionados y maneja una capacidad de equilibrio de carga.	Google permite a los desarrolladores tener un descuento por uso continuo.
<b>Amazon Web Services</b>	Permite escalar inferior o superiormente los planes mediante el uso de instancias de Amazon EC2	AWS ofrece descuento a usuarios de bajo nivel, pequeñas y medianas empresas, con opciones de precio de menor costo de implementación flexible.
<b>Microsoft Azure</b>	Utiliza máquinas virtuales para escalar los servicios de la nube, aplicaciones web y gestión API.	Azure solo ofrece descuentos a grandes empresas.
<b>Hostinger</b>	El plan de hosting compartido solo cuenta con 30 GB de almacenamiento en los discos SSD. Su mayor inconveniente son las funciones limitadas en los proyectos de gran escala.	Precios muy competitivos en relación con otro hosting de características similares.

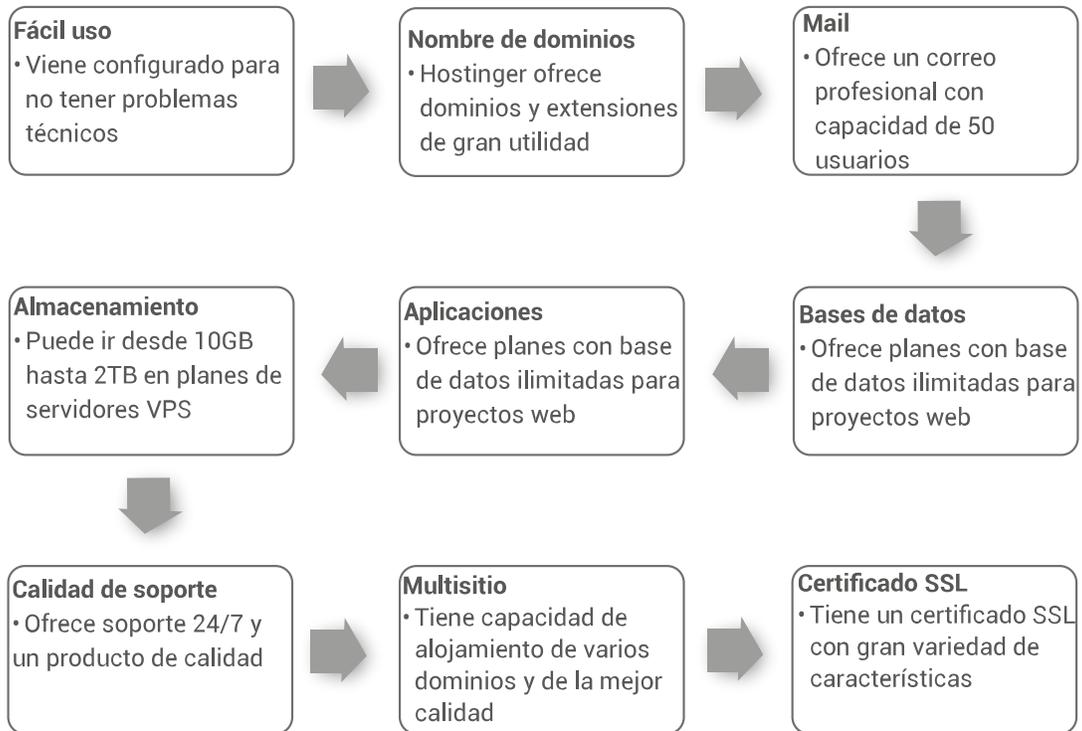
Fuente: Tomada de Hostinger (2020).

### **La capa gratuita es más extendida**

Lo más atractivo de los servicios de Hostinger es que cuenta con servicio de hosting compartido, hosting cloud y servicios de VPS a precios muy asequibles para el usuario y servicios de alta calidad. Tiene el distintivo frente a sus competidores debido a su bajo costo en mantenimiento a largo plazo ya que ofrece planes a precios muy cómodos que se adaptan a las necesidades del cliente (MuyComputer, 2018). El ser una de las Cloud más baratas del mercado con múltiples características importantes, la ubican como una opción eficiente para las condiciones de trabajo descritas en la Figura 5.

**Figura 5.**

*Características del proveedor elegido.*



Fuente: Tomada de (DigitalHerramienta, 2022)

### **2.3.3 Fase 3: Arquitectura de VPL en la nube sobre Moodle**

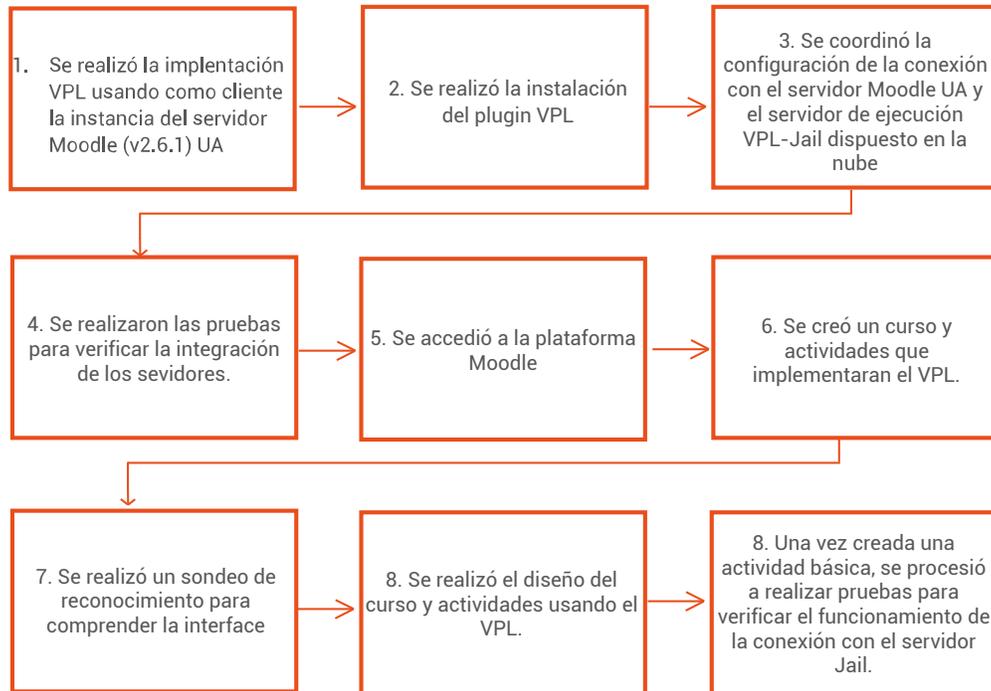
Para realizar el proceso de verificación de la versión de Moodle compatible con VPL, se realizaron pruebas que permitieron definir la estabilidad de funcionamiento, identificado los pasos para implementar las versiones Moodle (v2.6.1) y Moodle (v3.8.4) y realizar un análisis de los problemas encontrados. A continuación, se presenta el análisis realizado a las versiones:

#### **Moodle (v2.6.1)**

Se procedió a implementar el VPL usando la instancia del servidor Moodle (V2.6.1) implementado en la Universidad de la Amazonia actualmente, con los pasos presentados en la Figura 6:

## Figura 6.

### Pasos para implementar Moodle V2.6.1.



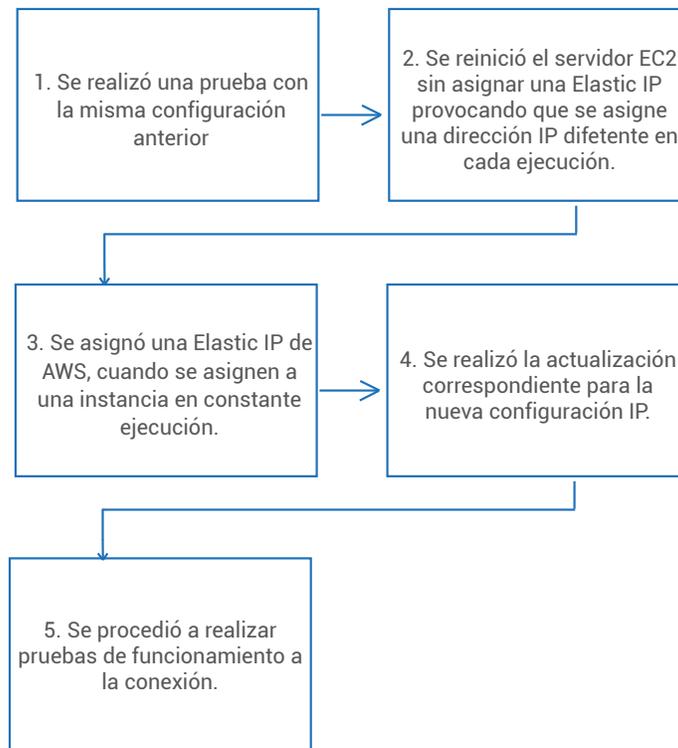
Fuente: Elaboración propia.

## Moodle (v3.8.4)

Se realizó el proceso de la implementación del VPL usando una instancia en fase de desarrollo en la versión Moodle (v3.8.4) siguiendo los pasos mostrados en la Figura 7.

### Figura 7.

#### Pasos para implementar Moodle V8.4.3.



Fuente: Elaboración propia.

En la tabla 6 se pueden visualizar los problemas presentados en las dos versiones de la plataforma Moodle.

**Tabla 6.** Problemas encontrados en las versiones de Moodle

Moodle (v2.6.1)	Moodle (v3.8.4)
1.El procedimiento de implementación del VPL en producción fue necesario realizarlo en un entorno controlado debido a que de esta manera se mitigaban los fallos en la plataforma Moodle en tiempo real.	1.Esta versión ofrece poca información acerca de los fallos que se generan con la integración de la Moodle al VPL.
2.Debido a la variación de la interfaz de usuario que se encuentra entre las versiones de Moodle, los usuarios tienen más dificultades para aprender el funcionamiento de las diferentes interfaces.	2.Dependencia del centro de datos de la Universidad de la Amazonia y los beneficios de auto escalado son dependientes de la capacidad del servidor cliente de la plataforma Moodle (Rodríguez et al., 2010).
	3.Las configuraciones de seguridad imposibilitan la conexión a servicios externos del VPL Jail.

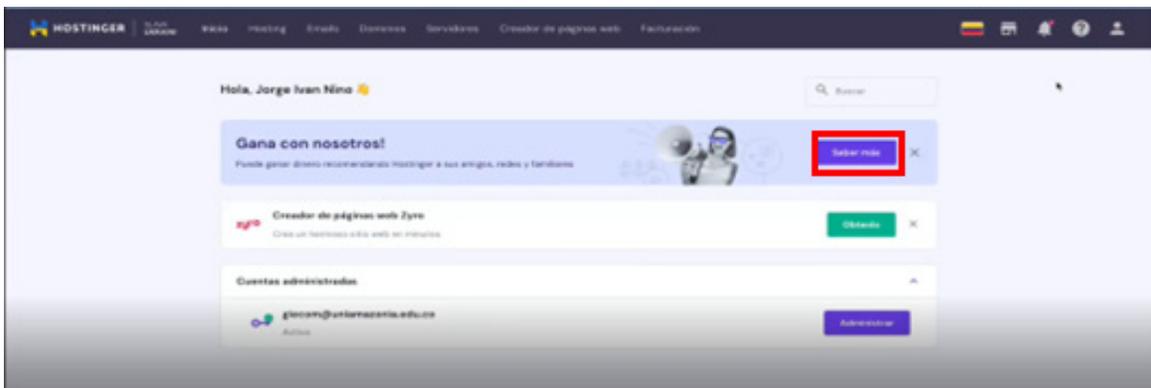
Fuente: Elaboración propia.

### 2.3.4 Fase 4: Puesta en marcha del Laboratorio Virtual de Programación (VPL) en la nube

1) Ingresar al sitio del proveedor Hostinger con las credenciales correspondientes y dar clic sobre la opción Administrar (Figura 8).

**Figura 8.**

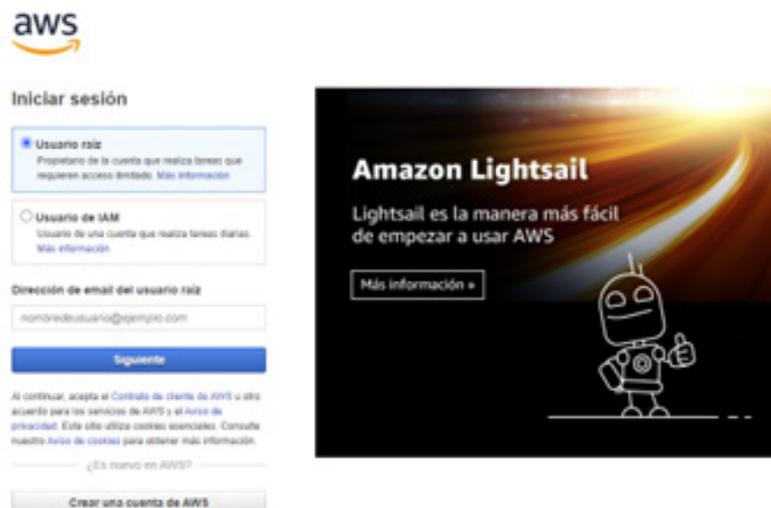
*Ingreso de credenciales de Hostinger.vFuente:*



Elaboración propia.

2) Iniciar sesión en AWS (Figura 9).

**Figura 9.** Inicio de sesión a AWS.

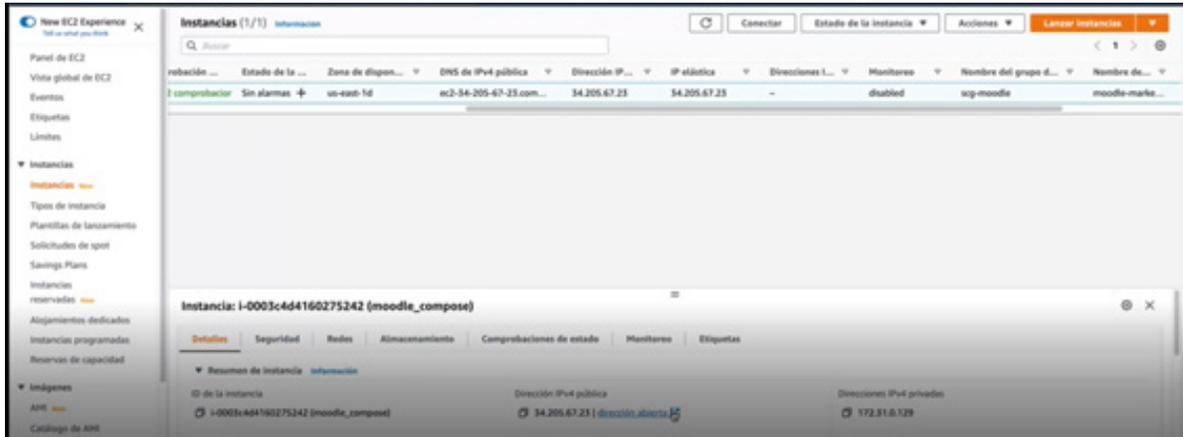


Fuente: Elaboración propia.

3) Dirigirse a las instancias EC2 de la Moodle en el servidor de AWS (Figura 10).

### Figura 10.

Instancias EC2 en el servidor AWS.

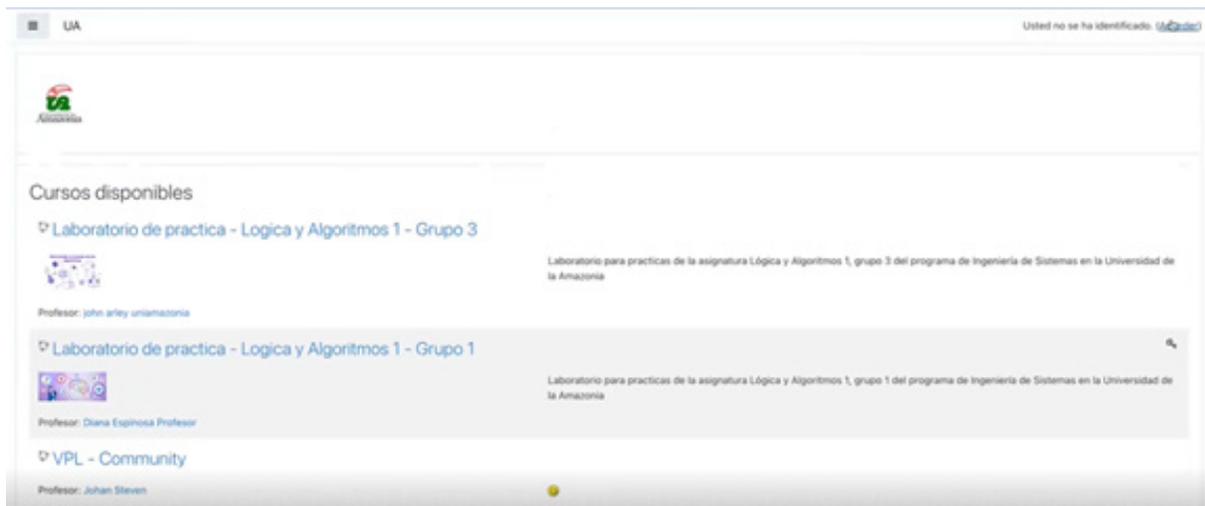


Fuente: Elaboración propia.

4) Accediendo a la dirección IP se redirecciona a los cursos alojados en el servidor y se da clic sobre la opción acceder (Figura 11).

### Figura 11.

Visualización de los cursos en el servidor.



Fuente: Elaboración propia.

5) Acceder con las credenciales indicadas (Figura 12).

**Figura 12.**

*Iniciar sesión con las credenciales.*

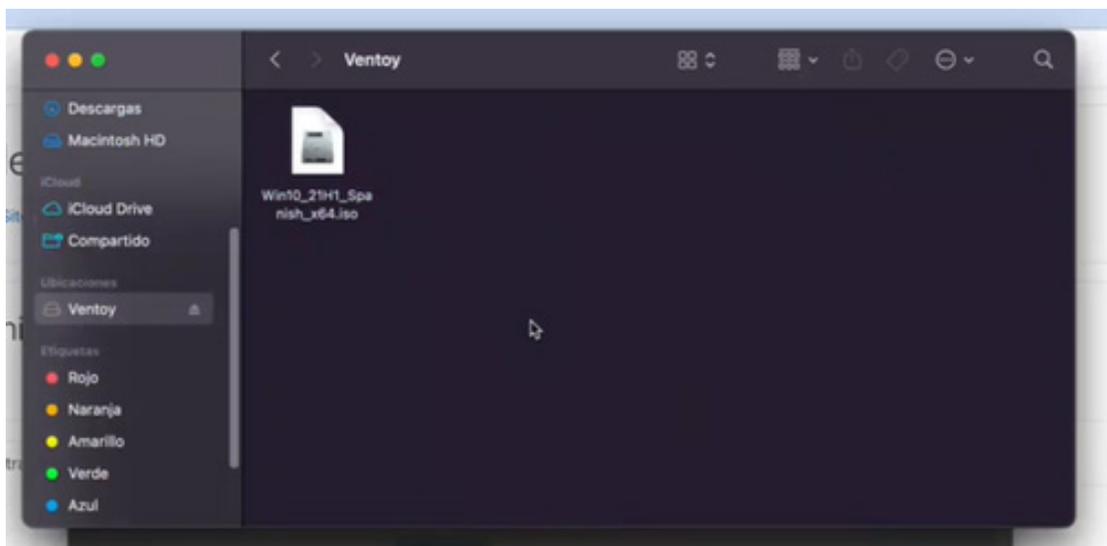


Fuente: Elaboración propia.

6) Buscar la key ssh que se genera al crear la máquina virtual (Figura 13).

**Figura 13.**

*Llave ssh alojada en el computador.*

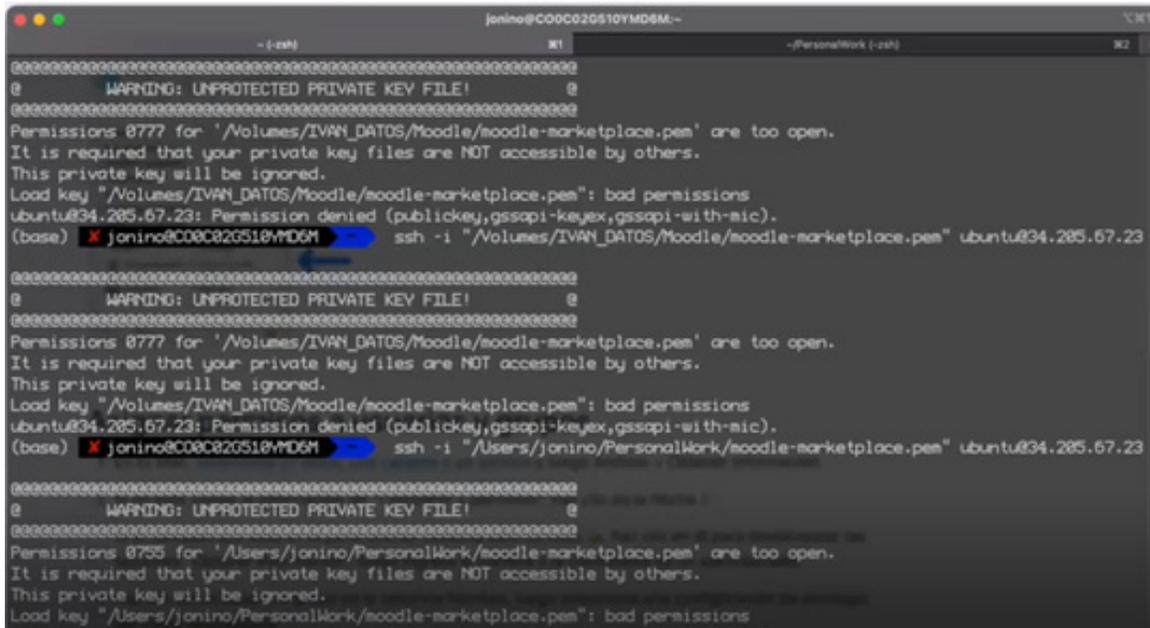


Fuente: Elaboración propia.

7) Cargar el archivo ssh (Figura 14).

**Figura 14.**

*Cargar archivo ssh.*



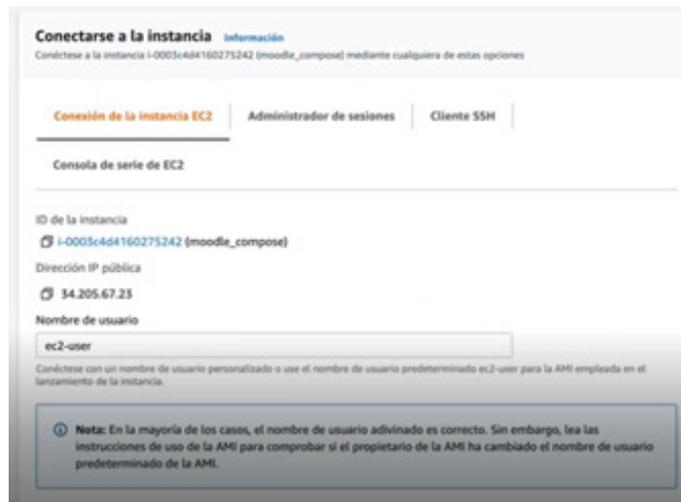
```
jonino@COOC020510YMD6M:~$ ssh -i "/Volumes/IVAN_DATOS/Moodle/moodle-marketplace.pem" ubuntu@34.205.67.23
Warning: Unprotected private key file!
Permissions 0777 for '/Volumes/IVAN_DATOS/Moodle/moodle-marketplace.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Volumes/IVAN_DATOS/Moodle/moodle-marketplace.pem": bad permissions
ubuntu@34.205.67.23: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
(base) jonino@COOC020510YMD6M:~$ ssh -i "/Users/jonino/PersonalWork/moodle-marketplace.pem" ubuntu@34.205.67.23
Warning: Unprotected private key file!
Permissions 0777 for '/Users/jonino/PersonalWork/moodle-marketplace.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/jonino/PersonalWork/moodle-marketplace.pem": bad permissions
```

Fuente: Elaboración propia.

8) Conectarse a la instancia EC2 (Figura 15).

**Figura 15.**

*Conectarse a la instancia.*

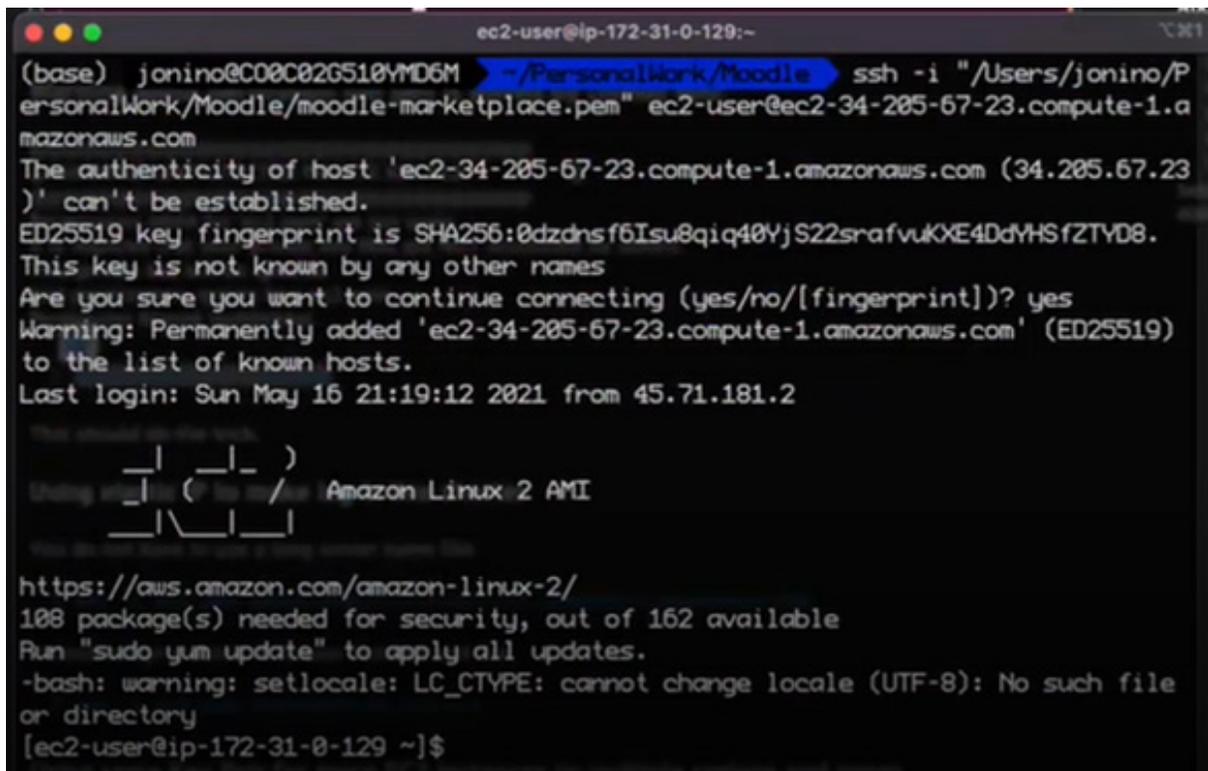


Fuente: Elaboración propia.

9) Dirigirse a la pestaña de Cliente SSH, copiar el código de la llave y pegarlo en la línea de comandos de la terminal. Se habilitaron los permisos de solo lectura para conectarse a la máquina virtual de AWS desde la cual se va a realizar la copia de seguridad de la información para migrarla de AWS a Hosting (Figura 16).

**Figura 16.**

*Código de la llave ssh en la terminal.*



```
ec2-user@ip-172-31-0-129:~  
(base) jonino@C00C02G510YMD6M ~/PersonalWork/Moodle ssh -i "/Users/jonino/PersonalWork/Moodle/moodle-marketplace.pem" ec2-user@ec2-34-205-67-23.compute-1.amazonaws.com  
The authenticity of host 'ec2-34-205-67-23.compute-1.amazonaws.com (34.205.67.23)' can't be established.  
ED25519 key fingerprint is SHA256:0dzdhsf6Isv8qiq40YjS22snafvuKXE4DdYHSfZTYD8.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-34-205-67-23.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
Last login: Sun May 16 21:19:12 2021 from 45.71.181.2  
  
This document lists the packages that are available for Amazon Linux 2 AMI.  
  
https://aws.amazon.com/amazon-linux-2/  
108 package(s) needed for security, out of 162 available  
Run "sudo yum update" to apply all updates.  
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory  
[ec2-user@ip-172-31-0-129 ~]$
```

Fuente: Elaboración propia.

**10)** Se descargan dos ficheros con los backups correspondientes, una copia de seguridad de la base de datos y otro de la plataforma Moodle (Figura 17).

**Figura 17.**

*Backups correspondientes.*

```
-rw-r--r-- 1 root root 35692539 Apr 21 2021 backup-moodle-db-v3.zip
-rw-r--r-- 1 root root 92559268 Apr 21 2021 backup-moodle-v3.zip
[ec2-user@ip-172-31-0-129 backups]$ _
```

Fuente: Elaboración propia.

**11)** Realizar una copia de la llave ssh (Figura 18).

**Figura 18.**

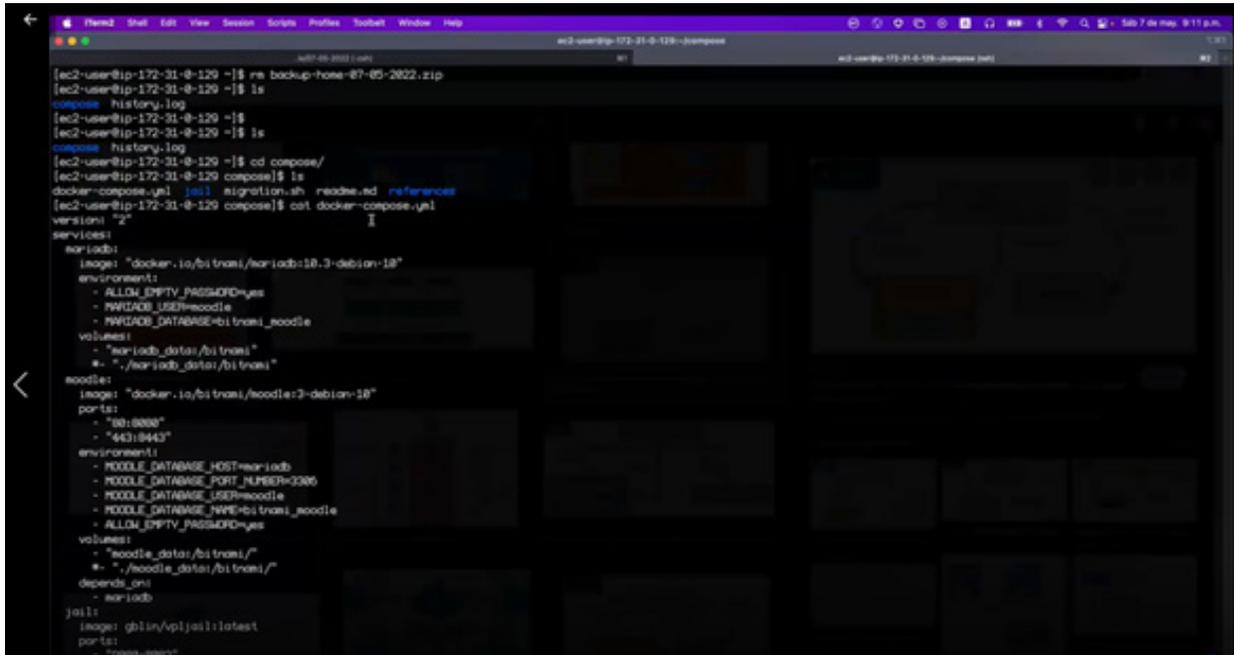
*Copia de la llave ssh.*

```
cd ~/.ssh
scp -i keykey.pem ec2-user@ipadress.zone.compute.amazonaws.com:/home/path/file ~/Des
```

Fuente: Elaboración propia.

**12)** En la Figura 19, se muestra la información del Docker donde se puede visualizar su estructura correspondiente (Servicio Base de Datos, Volúmenes, Moodle y Jail) (Medina, 2021),

**Figura 19.**  
*Estructura del Docker.*



```
[ec2-user@ip-172-31-0-129 ~]$ rs backup-home-07-05-2022.zip
[ec2-user@ip-172-31-0-129 ~]$ ls
compose  history.log
[ec2-user@ip-172-31-0-129 ~]$
[ec2-user@ip-172-31-0-129 ~]$ ls
[ec2-user@ip-172-31-0-129 ~]$ cd compose/
[ec2-user@ip-172-31-0-129 compose]$ ls
docker-compose.yml  jails  migration.sh  moodie.md  references
[ec2-user@ip-172-31-0-129 compose]$ cat docker-compose.yml
version: "2"
services:
  nor-iadb:
    image: "docker.io/bitnami/nor-iadb:18.3-debian-18"
    environment:
      - ALLOW_EMPTY_PASSWORD=yes
      - MYSQLDB_USER=moodle
      - MYSQLDB_DATABASE=bitnami_moodle
    volumes:
      - "nor-iadb_data:/bitnami/"
      #- "/nor-iadb_data:/bitnami/"
  moodie:
    image: "docker.io/bitnami/moodle:3-debian-18"
    ports:
      - "80:8080"
      - "443:8443"
    environment:
      - MOODLE_DATABASE_HOST=nor-iadb
      - MOODLE_DATABASE_PORT_NUMBER=3306
      - MOODLE_DATABASE_USER=moodle
      - MOODLE_DATABASE_NAME=bitnami_moodle
      - ALLOW_EMPTY_PASSWORD=yes
    volumes:
      - "moodle_data:/bitnami/"
      #- "/moodle_data:/bitnami/"
    depends_on:
      - nor-iadb
  jails:
    image: ghlin/vp/jails:latest
    ports:
      - "2222:2222"
```

Fuente: Elaboración propia.

**13)** Se realiza un binding entre el backup del volumen de la plataforma Moodle y el backup creado en la máquina virtual (Figura 20).

**Figura 20.**  
*Binding.*



```
Run 'docker volume COMMAND --help' for more information on a command.
[ec2-user@ip-172-31-0-129 compose]$ docker volume ls
DRIVER          VOLUME NAME
local          compose_nor-iadb_data
local          compose_moodle_data
[ec2-user@ip-172-31-0-129 compose]$ docker run --name moodie -v compose_moodle_data:/volume -v /psd/backup-moodle-v2:/backup busybox sh -c 'cp -r /volume /backup'
```

Fuente: Elaboración propia.

## Crear llaves ssh en Hostinger

1) Se puede visualizar toda la estructura del volumen (Jail, Moodle y Base de Datos) en la Figura 21.

### Figura 21.

Estructura del volumen.

```
[ec2-user@ip-172-31-0-129 ~]$ ls
07-05-2022  compose  history.log
[ec2-user@ip-172-31-0-129 ~]$ cd 07-05-2022/
[ec2-user@ip-172-31-0-129 07-05-2022]$ ls
backup-moodle-v2
[ec2-user@ip-172-31-0-129 07-05-2022]$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                                                                                            NAMES
465bc024dcae   gblin/vpljail:latest                "/entrypoint.sh"        19 months ago Up 19 months  0.0.0.0:9443->8080/tcp, 0.0.0.0:9080->8082/tcp                                                         compose_j
3c77cd9700d4   bitnami/moodle:3-debian-10          "/opt/bitnami/script..." 20 months ago Up 19 months  0.0.0.0:80->8080/tcp, 0.0.0.0:443->9443/tcp                                                         compose_m
cd95d191fe583   bitnami/mariadb:10.3-debian-10     "/opt/bitnami/script..." 20 months ago Up 19 months  3306/tcp                                                                                                                                    compose_m
ariadb_1
[ec2-user@ip-172-31-0-129 07-05-2022]$ ls
backup-moodle-v2
[ec2-user@ip-172-31-0-129 07-05-2022]$ cd backup-moodle-v2/
[ec2-user@ip-172-31-0-129 backup-moodle-v2]$ ls
volume
[ec2-user@ip-172-31-0-129 backup-moodle-v2]$ cd volume/
[ec2-user@ip-172-31-0-129 volume]$ ls
moodle  moodb  data
[ec2-user@ip-172-31-0-129 volume]$
```

Fuente: Elaboración propia.

2) Se comprime el backup de la Moodle y de la base de datos del servidor de AWS (Figura 22).

### Figura 22.

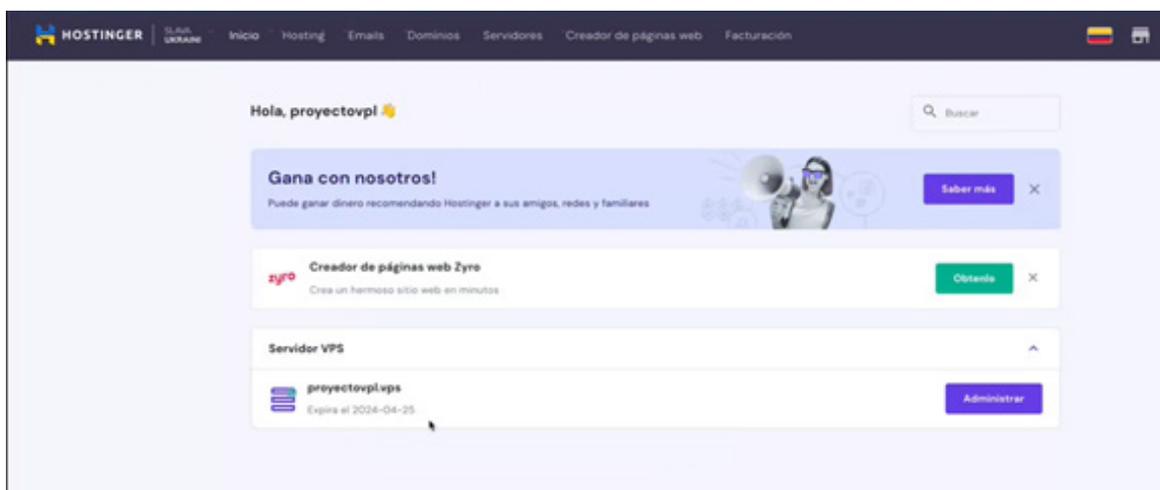
Copias de seguridad de la plataforma Moodle y la base de datos.

```
[ec2-user@ip-172-31-0-129 ~]$ ^Cp -r backup-moodle-and-mariadb-07-05-2022.zip
[ec2-user@ip-172-31-0-129 ~]$ ls
07-05-2022  compose  history.log
[ec2-user@ip-172-31-0-129 ~]$ cd
```

Fuente: Elaboración propia.

3) Se puede visualizar el servidor VPS (Figura 23).

**Figura 23.**  
*Servidor VPS.*



Fuente: Elaboración propia.

4) Administrar clave ssh (Figura 24)

*Figura 24.*  
*Administración clave ssh.*

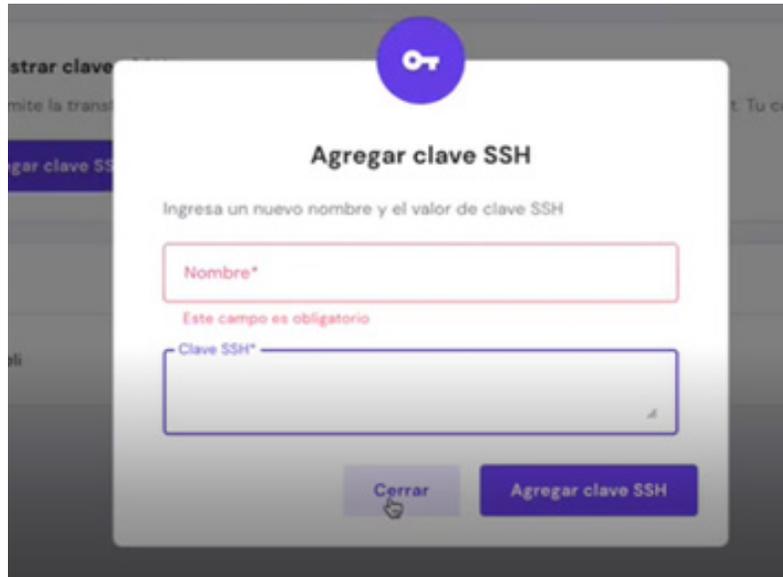


Fuente: Elaboración propia.

5) Se agrega la clave ssh para conectarnos al servidor de Hostinger (Figura 25).

**Figura 25.**

*Agregar clave ssh al servidor Hostinger.*



Fuente: Elaboración propia.

6) Se genera la llave ssh y se define dirección de almacenamiento (Figura 26).

**Figura 26.**

*Generar la llave ssh.*

```
(base) jonino@C08C82G518YMD6M ~/PersonalWork/Moodle cd ..
(base) jonino@C08C82G518YMD6M ~/PersonalWork ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/jonino/.ssh/id_rsa): /Users/jonino/.ssh/vpl_hostinger
/Users/jonino/.ssh/vpl_hostinger already exists.
Overwrite (y/n)?
```

Fuente: Elaboración propia.

7) Se agregó la llave ssh (Figura 27).

### Figura 27

*Agregar la llave ssh.*

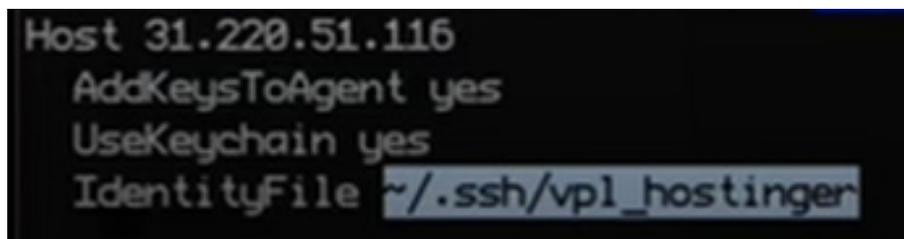


Fuente: Elaboración propia.

8) Las peticiones al servidor se realizan a través de la llave (Figura 28).

### Figura 28.

*Petición al servidor.*



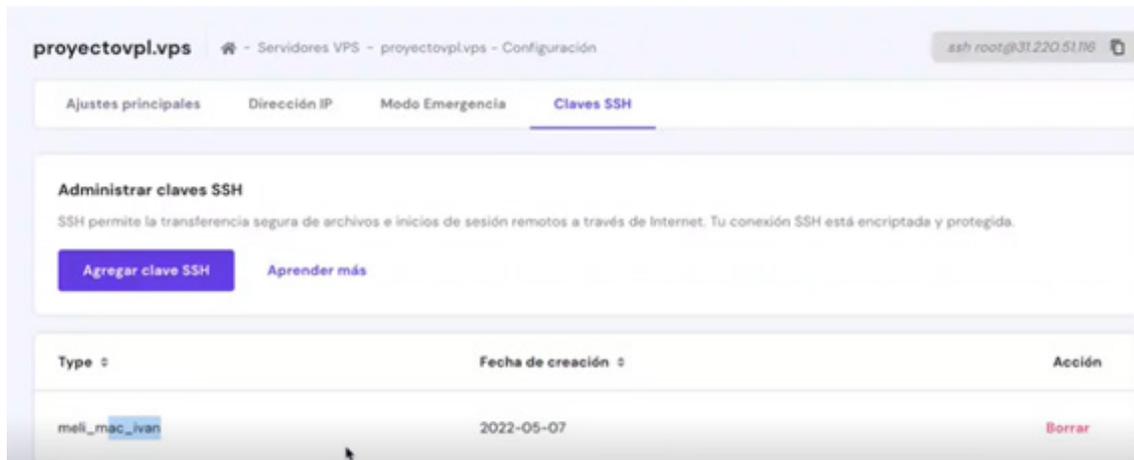
Fuente: Elaboración propia.

## Despliegue del VPL en Hostinger

1) Se agrega la llave SSH (Figura 29).

**Figura 29.**

*Agregar llave ssh.*

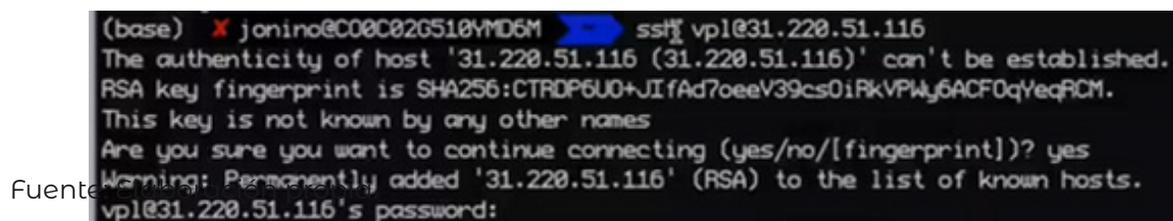


Fuente: Elaboración propia.

2) Para conectarse mediante ssh se utiliza dirección IP, usuario y contraseña asignada (Figura 30).

**Figura 30.**

*Conexión por ssh.*



3) Se conectó al VPS de Hostinger correctamente (Figura 31).

### Figura 31.

Conexión al VPS de Hostinger.

```
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/advantage
Last login: Sun May  8 00:05:52 2022 from 64.191.220.148
vp1@projectovp1:~$ _
```

Fuente: Elaboración propia.

4) Se actualizan referencias desde donde se van a descargar las librerías (Figura 32).

### Figura 32.

Actualización de referencias.

```
vp1@projectovp1:~$ sudo apt-get update
[sudo] password for vp1: _
```

Fuente: Elaboración propia.

5) Descargar Docker compose en la versión 1.24.0 (Figura 33).

### Figura 33.

Descarga de Docker Compose.

```
root@projectovp1:~# sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Fuente: Elaboración propia.

6) Ejecutar Docker agregando un usuario de ejecución (Figura 34).

### Figura 34.

Ejecutar Docker.

```
root@projectovp1:~# chmod +x /usr/local/bin/docker-compose
```

Fuente: Elaboración propia.

7) Se realiza la instalación de Docker en la versión 20.10.15 (Figura 35).

**Figura 35.**

*Instalación de Docker.*

```
Get:13 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-scan-plugin amd64 0.17.0-ubuntu-focal [3521 kB]
Fetched 187 MB in 9s (11.9 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 85218 files and directories currently installed.)
Preparing to unpack .../80-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package libpam-systemd:amd64.
Preparing to unpack .../81-libpam-systemd_237-3ubuntu10.53_amd64.deb ...
Unpacking libpam-systemd:amd64 (237-3ubuntu10.53) ...
Selecting previously unselected package apparmor.
Preparing to unpack .../82-apparmor_2.12-4ubuntu5.1_amd64.deb ...
Unpacking apparmor (2.12-4ubuntu5.1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../83-containerd.io_1.6.4-1_amd64.deb ...
Unpacking containerd.io (1.6.4-1) ...
Selecting previously unselected package dbus-user-session.
Preparing to unpack .../84-dbus-user-session_1.12.2-1ubuntu1.2_amd64.deb ...
Unpacking dbus-user-session (1.12.2-1ubuntu1.2) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../85-docker-ce-cli_5:20.10.15-3-0-ubuntu-focal_amd64.deb ...
Unpacking docker-ce-cli (5:20.10.15-3-0-ubuntu-focal) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../86-docker-ce_5:20.10.15-3-0-ubuntu-focal_amd64.deb ...
Unpacking docker-ce (5:20.10.15-3-0-ubuntu-focal) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../87-docker-ce-rootless-extras_5:20.10.15-3-0-ubuntu-focal_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:20.10.15-3-0-ubuntu-focal) ...
Selecting previously unselected package docker-scan-plugin.
Preparing to unpack .../88-docker-scan-plugin_0.17.0-ubuntu-focal_amd64.deb ...
Unpacking docker-scan-plugin (0.17.0-ubuntu-focal) ...
Selecting previously unselected package libcurl13-gnutls:amd64.
Preparing to unpack .../89-libcurl13-gnutls_7.58.0-2ubuntu3.17_amd64.deb ...
Unpacking libcurl13-gnutls:amd64 (7.58.0-2ubuntu3.17) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../90-liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../91-git-man_1:3a2.17.1-1ubuntu0.11_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.11) ...
Selecting previously unselected package git.
Preparing to unpack .../92-git_1:3a2.17.1-1ubuntu0.11_amd64.deb ...
```

Fuente: Elaboración propia.

8) Se copia al fichero de Docker compose para realizar el levantamiento de los servicios en Hostinger (Figura 36).

**Figura 36.**

*Levantamiento de los servicios.*

```
jail:
image: gblin/pjail:latest
ports:
  - "9080:9082"
  - "9443:9080"
volumes:
  mariadb_data:
    driver: local
  moodie_data:
    driver: local
(base) j@inu8208320518vmd09 [~/PersonalWork/Moodie/7-85-2022/backup-home-07-85-2022/compose] scp -i ~/PersonalWork/Moodie/moodie-marketplace.pem ec2-user@ec2-34-285-67-23
.compute-1.amazonaws.com:/home
(base) j@inu8208320518vmd09 [~/PersonalWork/Moodie/7-85-2022/backup-home-07-85-2022/compose] scp -i ~/PersonalWork/Moodie/7-85-2022/backup-home-07-85-2022/compose/docker-
compose.yml
```

Fuente: Elaboración propia.

9) Se utiliza el comando Docker compose up para crear la estructura desde cero en Hostinger (Figura 37).

**Figura 37.**

*Crear estructura de cero en Hostinger.*

```
7a8d56fb5f59: Pull complete
165191c9b7c1: Downloading [=====>]
637f59371795: Downloading [=====>]
28.34MB/22.81MBDownload complete
b14af9ceec22: Downloading [=====>]
458kB/2.924MBbiting
177b04d39680: Waiting
1fbbd9c9fc6b: Waiting
a1bdf65fabf3: Waiting
f2458e65a5b: Waiting
f1cf73dae6f0: Waiting
c77abd75d63e: Waiting
86ed8bc9c2b1: Waiting
32fa6c58a8d4: Waiting
2cd8b22ad8a7: Waiting
7bdec49a8819: Waiting
4cb94108dc3d: Waiting
87cfa1cda7ef: Waiting
ed126be7d3cd: Waiting
97f86289534e: Waiting
3b4316f513d0: Waiting
7b8d41bd7951: Waiting
929c9bf4498b: Waiting
```

Fuente: Elaboración propia.

### Pasar la información de AWS a Hostinger

1) Conexión al servidor de Hostinger mediante el usuario root (Figura 38).

**Figura 38.**

*Conexión al servidor de Hostinger.*

```
dev lib media proc sbin L: webin-setup.out
root@projectovpl:/* cd home/
root@projectovpl:/home# ls
vpl
root@projectovpl:/home# cd vpl/
root@projectovpl:/home/vpl# ls
docker-compose.yml
root@projectovpl:/home/vpl# docker-compose up
Creating network "vpl_default" with the default driver
Creating volume "vpl_mariadb_data" with local driver
Creating volume "vpl_moodle_data" with local driver
Pulling mariadb (docker.io/bitnami/mariadb:10.3-debian-10)...
10.3-debian-10: Pulling from bitnami/mariadb
fa7a704f09f0: Downloading [==>]
1.112MB/26.77MBDownload complete
87999ddd864c: Downloading [>]
228.6kB/21.72MBbiting
08a9ec225914: Waiting
b5805ea0e4b8: Waiting
f970634388b3: Waiting
ca98c9b0bd1f: Waiting
```

Fuente: Elaboración propia.

2) El comando docker-compose up se utiliza para descargar los contenedores base sobre los que se iba a ejecutar la migración (Figura 39).

### Figura 39.

*Descargar contenedores.*

```
vp1
root@projectovp1:/home# cd vp1/
root@projectovp1:/home/vp1# ls
docker-compose.yml
root@projectovp1:/home/vp1# docker-compose up
```

Fuente: Elaboración propia.

3) Al realizar el levantamiento de los servicios, se puede visualizar un error, debido a que queríamos utilizar un puerto que ya Hostinger asignaba por defecto por lo tanto ya estaba en uso (Figura 40).

### Figura 40.

*Error por puerto ocupado.*

```
57a2da5f72a: Pull complete
8f52bd1e5cb8: Pull complete
bc49cae45772: Pull complete
373ca1f5b663: Pull complete
2daa725a29c6: Pull complete
83a9a9762beb: Pull complete
Digest: sha256:8648a5d5c192232f97e44d7594a4ebdbae69a614623e587bc8b68f038c6d77c
Status: Downloaded newer image for gblin/vp1jail:latest
Creating vp1_jail_1 ... done
Creating vp1_seriadb_1 ... done
Creating vp1_moodle_1 ... error

ERROR: for vp1_moodle_1 Cannot start service moodle: b'driver failed programming external connectivity on endpoint vp1_moodle_1 (79f6979d87491975eb9e044baff11876c13eeelfcd29aec832befaa50d): Error starting userland proxy: listen tcp4 0.0.0.0:80: bind: address already in use'

ERROR: for moodle Cannot start service moodle: b'driver failed programming external connectivity on endpoint vp1_moodle_1 (79f6979d87491975eb9e044baff11876c13eeelfcd29aec832befaa50d): Error starting userland proxy: listen tcp4 0.0.0.0:80: bind: address already in use'

ERROR: Encountered errors while bringing up the project.
root@projectovp1:/home/vp1# ls
docker-compose.yml
root@projectovp1:/home/vp1#
```

Fuente: Elaboración propia.

4) A través de este comando se consultan los puertos para saber cuáles de estos están en uso y cuáles no (Figura 41).

### Figura 41.

Consulta de los puertos ocupados.

```
root@projectovpl:/home/vpl# ls
docker-compose.yml
root@projectovpl:/home/vpl# sudo netstat -pna | grep_
```

Fuente: Elaboración propia

5) Se listan los servicios que estaban activos en los puertos y se procede a detener el servicio de apache, a través del comando `service apache2 stop` (Figura 42).

### Figura 42.

Servicios activos.

```
[ - ] hcclock.sh
[ - ] keyboard-setup.sh
[ + ] kmod
[ ? ] modules_dep.sh
[ + ] networking
[ + ] procps
[ - ] quota
[ - ] quotarpc
[ + ] resolvconf
[ - ] rpcbind
[ - ] rsysync
[ + ] rsyslog
[ + ] saslauthd
[ - ] screen-cleanup
[ + ] ssh
[ + ] udev
[ + ] unattended-upgrades
[ - ] uuid
[ + ] webmin
[ - ] wide-dhcpv6-client
[ + ] xinetd
root@projectovpl:/home/vpl# service apache2 stop
root@projectovpl:/home/vpl# ls
docker-compose.yml
root@projectovpl:/home/vpl# service apache2 stop_
```

Fuente: Elaboración propia.

6) Después de detener el servicio se intenta instalar de nuevo la arquitectura debido a que este proceso antes no era posible (Figura 43).

### Figura 43.

Instalar nuevamente la arquitectura.

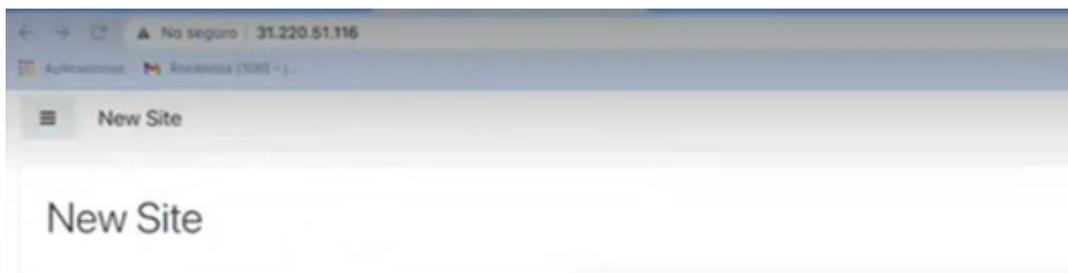
```
config      Validate and view the Compose file
create     Create services
down       Stop and remove containers, networks, images, and volumes
events     Receive real time events from containers
exec       Execute a command in a running container
help       Get help on a command
images     List images
kill       Kill containers
logs       View output from containers
pause     Pause services
port      Print the public port for a port binding
ps         List containers
pull      Pull service images
push      Push service images
restart    Restart services
rm         Remove stopped containers
run        Run a one-off command
scale     Set number of containers for a service
start     Start services
stop      Stop services
top       Display the running processes
unpause   Unpause services
up        Create and start containers
version   Show the Docker-Compose version information
root@projectvpl:/home/vpl# docker-compose _
```

Fuente: Elaboración propia.

7) Se verificó que al abrir el puerto estuviera la plataforma Moodle desde cero (Figura 44).

### Figura 44.

Visualización de la plataforma Moodle de cero.



Fuente: Elaboración propia.

8) Se puede visualizar el puerto que está ocupando la plataforma Moodle (Figura 45).

### Figura 45.

Puerto ocupado.

```
48f8e98614f3 bitnami/moodle:3-debian-18 "/opt/bitnami/script..." 18 minu
tes ago Up 14 minutes 0.0.0.0:80->8080/tcp, :::80->8080/tcp, 0.0.0.0:443->84
```

Fuente: Elaboración propia.

9) Se descargó otra copia de seguridad debido a que entre el paso número 8 y los siguientes pasos había transcurrido un periodo de tiempo de 7 días y se ejecutaron cambios (Figura 46).

### Figura 46.

*Descargar nueva copia de seguridad.*

```
(base) jonino@C00C020518YMD6M ~$ cd /home/backup-noodle/
87-05-2022 Info vieja
(base) jonino@C00C020518YMD6M ~$ cd 87-05-2022
(base) jonino@C00C020518YMD6M ~$ ls
backup-home-87-05-2022 backup-noodle-and-mariadb-87-05-2022.zip backups-21-04-2021
backup-home-87-05-2022.zip backups-02-05-2021
(base) jonino@C00C020518YMD6M ~$ scp ./backup-noodle-and-mariadb-87-05-2022.zip vpl831.228.5
1.116:/home/vpl/backup-noodle-and-mariadb-87-05-2022.zip
vpl831.228.51.116's password:
backup-noodle-and-mariadb-87-05-2022.zip 100% 91MB 629.7KB/s 02:28
(base) jonino@C00C020518YMD6M ~$
```

Fuente: Elaboración propia.

10) Al estar conectados con el usuario de AWS se comprime la copia de seguridad y se guarda para posteriormente almacenarlo en el servidor de Hostinger (Figura 47).

### Figura 47.

*Comprimir copia de seguridad.*

```
[ec2-user@ip-172-31-8-129 15-05-2022]$ ls
backup-noodle-db-v2
[ec2-user@ip-172-31-8-129 15-05-2022]$ sudo zip -r backup-noodle-db-v2
[ec2-user@ip-172-31-8-129 15-05-2022]$ sudo zip -r backup-noodle-db-v2.zip backup-noodle-db-v2/
```

Fuente: Elaboración propia.

11) Con este comando se descarga el fichero almacenado en AWS (Figura 48).

### Figura 48.

*Descarga de fichero almacenado en AWS.*

```
Connection to ec2-34-205-67-23.compute-1.amazonaws.com closed.
(base) jonino@C00C020518YMD6M ~$ scp -i ../noodle-marketplace.pem ec2-user@ec2-34-205-67-23.compute-1.amazonaws.com:/home/
ec2-user/backups/backup-noodle-db-v3.zip ./backup-noodle-db-v3.zip
```

Fuente: Elaboración propia.

12) En la Figura 49, se utiliza el comando para subir la información descargada del fichero al servidor de Hostinger

## Figura 49.

Subir información a Hostinger.

```
base) x jonino@C08020518YMD0M scp backup-moodle-db-v2.zip root@31.228.51.116:/home/vpl/b
ckup-mariadb.zip
backup-moodle-db-v2.zip 95% 39MB 694.5KB/s 00:02 ETA
```

Fuente: Elaboración propia.

13) Con este comando se transfiere recursivamente la información almacenada del volumen de la base de datos MariaDB al servidor de Hostinger (Figura 50).

## Figura 50.

Transferir información de Maria DB a Hostinger.

```
07-05-2022 backup-mariadb.zip backup-moodle-db-v2 backup-moodle-and-mariadb-07-05-2022.zip compose
root@projectovpl:/home/vpl# docker run --rm -v compose_mariadb_data:/volume -v 'pid' /backup-moodle-db-v2:/backup busybox sh -c '
cp -r /backup /volume'
```

Fuente: Elaboración propia.

14) Al realizar el proceso anterior se podía visualizar un error al pasar la información (Figura 51).

## Figura 51.

Error al transferir información.

```
jail_1 | During handling of the above exception, another exception occurred:
jail_1 |
jail_1 | Traceback (most recent call last):
jail_1 |   File "/sethostrase.py", line 7, in <module>
jail_1 |     all_containers = cli.containers()
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/docker/api/container.py", line 85, in containers
jail_1 |     res = self._result(self._get(u, params=params), True)
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/docker/utils/decorators.py", line 47, in inner
jail_1 |     return f(self, *args, **kwargs)
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/docker/client.py", line 139, in _get
jail_1 |     return self.get(url, **self._set_request_timeout(kwargs))
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/requests/sessions.py", line 546, in get
jail_1 |     return self.request('GET', url, **kwargs)
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/requests/sessions.py", line 533, in request
jail_1 |     resp = self.send(prep, **send_kwargs)
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/requests/sessions.py", line 646, in send
jail_1 |     r = adapter.send(request, **kwargs)
jail_1 |   File "/usr/local/lib/python3.5/dist-packages/requests/adapters.py", line 498, in send
jail_1 |     raise ConnectionError(err, request=request)
jail_1 | requests.exceptions.ConnectionError: ('Connection aborted.', FileNotFoundError(2, 'No such file or directory'))
^C [Ctrl] Aborting.
root@projectovpl:/home/vpl/compose# docker-compose lo_
```

Fuente: Elaboración propia.

13) Para solucionar el error anterior se utilizó el comando especificado en la ilustración que tiene la función de eliminar el contenido del volumen antes de pasar la copia de seguridad de la base de datos y de la plataforma Moodle al servidor de Hostinger (Figura 52).

### Figura 52.

*Eliminar el contenido del volumen.*

```
^C^C^C: Aborting.
root@projectovpl1:/home/vpl/compose# ls
docker-compose.yml
root@projectovpl1:/home/vpl/compose# docker run -it --rm -v compose_moodle_data:/volume_moodle -v compose_mariadb_data:/volume_ma
riadb busybox sh
/* ls
```

Fuente: Elaboración propia.

15) Al revisar los volúmenes nos damos cuenta de que los backups están realizados correctamente (Figura 53).

### Figura 53.

*Copias realizadas correctamente.*

```
riadb busybox sh
/* ls
bin          etc          proc         sys          usr          volume_mariadb
dev          home        root         tap          var          volume_moodle
/* cd volume_moodle/
/volume_moodle * ls
moodle      moodledata
/volume_moodle * cd ..
/* ls
bin          etc          proc         sys          usr          volume_mariadb
dev          home        root         tap          var          volume_moodle
/* cd volume_mariadb/
/volume_mariadb * ls
mariadb
/volume_mariadb * cds
sh: cds: not found
/volume_mariadb * cd mariadb/
/volume_mariadb/mariadb * ls
data
/volume_mariadb/mariadb * cd data/
/volume_mariadb/mariadb/data * ls
aria_log.000000001  ib_buffer_pool  ibdata1      mysql        tc.log
aria_log_control  ib_logfile0    ibtapi       mysql_upgrade_info  test
bitnami_moodle    ib_logfile1    multi-master.info  performance_schema
```

Fuente: Elaboración propia.

16) Al reiniciar los servidores se puede visualizar que todo está funcionando correctamente. (Figura 54).

### Figura 54.

Despliegue del VPL realizado correctamente.



Fuente: Elaboración propia.

### Comandos más importantes

En la realización del despliegue del VPL se utilizaron múltiples comandos sin embargo en la Figura 55, se pueden visualizar lo más importantes para el proyecto.

### Figura 55.

Comandos principales.

```
1 # copy backups files docker volumes
2 scp ./backup-moodle-and-mariadb-07-05-2022.zip vpl@31.220.51.116:/home/vpl/backup-moodle-and-mariadb-07-05-2022.zip
3
4 # unzip backup on server
5 sudo unzip file_backups.zip
6
7 # restore backups with busybox
8 ## moodle data
9 docker run --rm -v compose_moodle_data:/volume -v `pwd`/backup-moodle-v2:/backup busybox sh -c 'cp -r /backup /volume'
10
11 ## mariadb data
12 docker run --rm -v compose_mariadb_data:/volume -v `pwd`/backup-moodle-db-v2:/backup busybox sh -c 'cp -r /backup /volume'
13
14
15 ## valide data volumes from busybox
16 docker run -it --rm -v compose_moodle_data:/volume_moodle -v compose_mariadb_data:/volume_mariadb busybox sh
```

Fuente: Elaboración propia.

### **Copy backups files Docker volumes**

Se utiliza para copiar información de una ruta local desde el computador a un servidor y carpeta específica.

### **Unzip backup en server**

Se utiliza para descomprimir los ficheros de una copia de seguridad.

### **Restore backups with busybox**

Para pasar la información alojada en el servidor al volumen de Docker y asegurarse que en ese espacio antes no hubiera nada.

### **Mariadb data**

Para pasar la información alojada en la base de datos al volumen de Docker y asegurarse que en ese espacio antes no hubiera nada.

### **Valide data volumes from busybox**

Con este comando se realiza una conexión entre los volúmenes (Moodle y mariadb) y la máquina virtual para borrar la información e incorporar la nueva actualizada de las copias de seguridad.

## 2.4 CONCLUSIONES

Al realizar el despliegue del VPL, se presentan percances propios de este tipo de proyectos Cloud, dado que las pruebas iniciales se llevaron a cabo en un servidor AWS en capa gratuita, pero al avanzar en el proyecto debido a requerimientos técnicos y de presupuesto, fue necesario migrar a un plan de pago en Hostinger. Esto implica realizar una copia de seguridad tanto de la plataforma Moodle como de la base de datos de MariaDB. Posteriormente crear las llaves ssh en el servidor de Hostinger, descargar Docker y Docker compose en el nuevo servidor, y restaurar la información alojada en las copias de seguridad en el nuevo servidor. Dicho proceso requirió realizar todas las fases de despliegue que previamente ya se habían realizado, pero esta vez en el nuevo servidor huésped.

Al realizar el levantamiento de los servicios de Hostinger se pudo visualizar un error debido a que el puerto que se quería utilizar ya estaba asignado por defecto; por este motivo fue necesario a través de comandos determinar los puertos disponibles, detener el servicio apache e instalar la arquitectura nuevamente para corregir el error. Cuando se realiza la migración de la Base de Datos MariaDB al servidor de Hostinger se produjo un error. Este inconveniente se soluciona eliminando el contenido del volumen antes de restaurar la copia de seguridad de la Base de Datos y de la plataforma Moodle al servidor de Hostinger.

Debido a los inconvenientes descritos en el despliegue del VPL estas actividades requirieron la ampliación del tiempo del proyecto, resultando bastante enriquecedoras en cuando a la experiencia obtenida en el proceso de despliegue en el servidor definitivo obteniendo como resultado su implementación efectiva la plataforma Moodle, lo que ha permitido que se encuentra funcionando correctamente sin interrupciones.

## 2.5 REFERENCIAS BIBLIOGRAFICAS

Castañeda, J. A., Ortega Álvarez, C. I., & Galvis Ibarra, J. C. I. (2019). Dockerizando Un Laboratorio Virtual De Programación (Vpl) Y Moodle En Google Cloud. Encuentro Internacional de Educación En Ingeniería, SE-. <https://acofipapers.org/index.php/eiei/article/view/212>

DigitalHerramienta. (2022). Todo lo que necesitas saber sobre Hostinger. <https://digitalherramienta.com/hostinger/>

Hostinger. (2020). Hostinger: Planes, Precios, Análisis y Opiniones. <https://trotahosting.com/hostinger/>

Hostinger. (2022). Planes de hosting amplios con más espacio SSD. <https://www.hostinger.co/>

Hostingplus. (2020). Qué es MariaDB y cuáles son sus características. <https://www.hostingplus.com.co/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/>

Medina, D. (2021). Curso Jakarta EE 9 (9). Docker (2): imágenes y contenedores. <https://danielme.com/2021/05/06/curso-jakarta-ee-docker-trabajando-con-imagenes-y-contenedores/>

Moodle (2020). Características de Moodle 3.8. [https://docs.moodle.org/all/es/Caracteristicas\\_de\\_Moodle\\_3.8](https://docs.moodle.org/all/es/Caracteristicas_de_Moodle_3.8)

Moodle. (2015). Acerca de Moodle. [https://docs.moodle.org/all/es/Acerca\\_de\\_Moodle](https://docs.moodle.org/all/es/Acerca_de_Moodle)

MuyComputer. (2018). Hostinger, el mejor aliado para hacer crecer tu negocio.

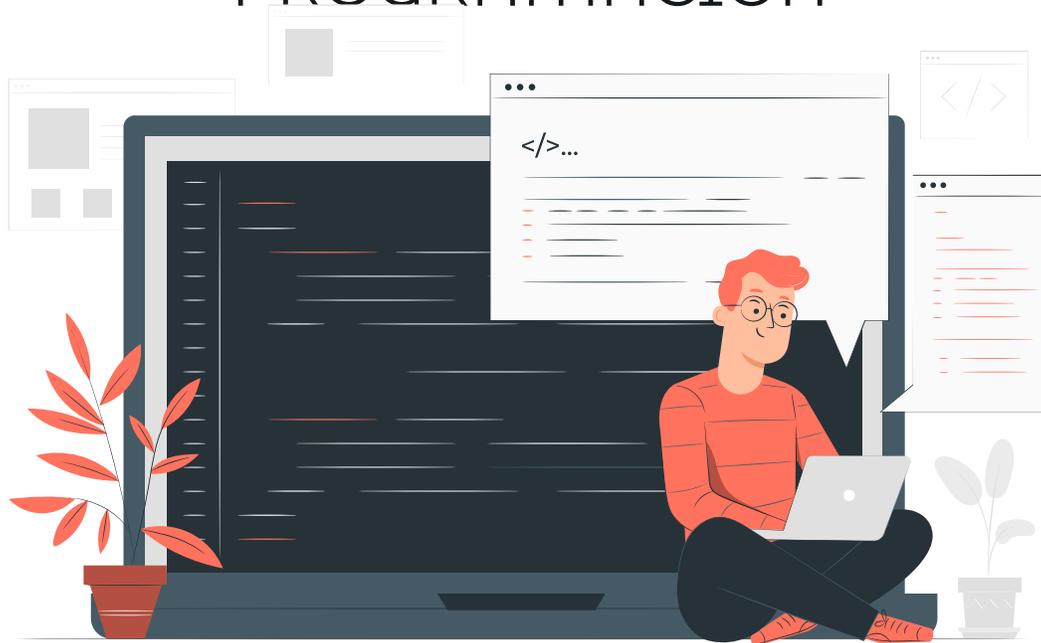
Niño Monje , J. I. (2022). Laboratorio virtual de programación integrado a moodle soportado en cloud computing para el fortalecimiento de las competencias en lógica algorítmica en los estudiantes de la facultad de ingeniería de la universidad de la amazonia.

Rodríguez del Pino, J. C., Rubio Royo , E., & Hernández Figueroa, Z. (2010). VPL: Laboratorio Virtual de Programación para Moodle.

Thiébaut, D. (2015). Automatic evaluation of computer programs using Moodle's virtual programming lab (VPL) plug-in. *Journal of Computing Sciences in Colleges*, 30(6), 145–151.

# CAPÍTULO III

## CONFIGURACIÓN DEL LABORATORIO VIRTUAL DE PROGRAMACIÓN



### **Diana María Espinosa Sarmiento**

Magister en Tecnologías de la Información y las Comunicaciones. Universidad de la Amazonia. Profesor categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-1873-0808>

### **John Arley García Quintero**

Magister en E-learning y Redes Sociales. Universidad Internacional de la Rioja. Estudiante del Doctorado en Ciencias de la Educación con Énfasis en: Investigación, Evaluación y Formulación de Proyectos Educativos de la Universidad Metropolitana de Educación, Ciencia y Tecnología (UMECIT). Profesor categoría Asociado, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-5530-1746>

### **Heriberto Fernando Vargas Losada**

Magister en Ciencias de la Información. Universidad Distrital. Estudiante del Doctorado en Educación y Cultura Ambiental, Universidad de la Amazonia. Profesor categoría Asociado, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación Giecom del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-8561-0793>

#### **CITACIÓN SUGERIDA:**

Espinosa Sarmiento, D, M. García Quintero, J, A. Vargas Losada, H, F. (2024). Configuración del laboratorio virtual de programación. Planteamiento del problema. Laboratorio virtual para el aprendizaje de fundamentos de programación. Editorial. Universidad de la Amazonia. p.p 121

### **3.1 PRESENTACIÓN**

El propósito de este capítulo es presentar los aspectos que permiten implementar un curso en Moodle utilizando actividades de tipo Laboratorio Virtual, para ello se describe el esquema de los ejercicios aplicados en la asignatura de Lógica y Algoritmos I, los aspectos básicos para su creación y configuración, y finalmente las recomendaciones tanto para los docentes como estudiantes al momento de utilizar el curso. Así, cabe destacar que el Laboratorio Virtual de Programación es una alternativa que se ajusta a los requerimientos particulares para el aprendizaje de la programación, convirtiéndose en una herramienta de apoyo para que los docentes puedan diseñar entornos de práctica de la programación controlados, a partir del diseño y aplicación de actividades de evaluación automática basada en casos de entrada y salida, la creación de tareas y evaluación, verificación de plagio entre otras (Ramos et al., 2021).

Sumado a lo anterior, las características de Moodle, la arquitectura en la nube y la dockerización de la aplicación brinda eficiencia al proceso de recibir, compilar, ejecutar, probar y enviar el resultado, bajo condiciones de alta demanda sobre la misma (Castañeda et al., 2019), lo que permite que el estudiante pueda practicar en cualquier momento, con una validación automática de su código y también con la posibilidad de recibir la retroalimentación por parte del docente, aspectos muy importantes para fortalecer el aprendizaje autónomo de los estudiantes (Muñoz et al., 2021; Rade et al., 2021).

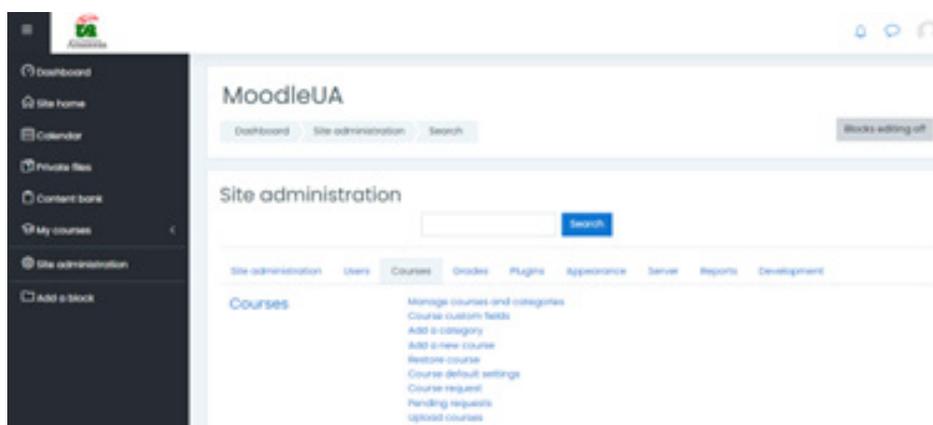
### 3.2 CONFIGURACIÓN DEL CURSO EN MOODLE

En primera instancia es necesario tener un curso dentro de la plataforma Moodle, para ello, las opciones más utilizadas son: crear un nuevo curso (vacio) o restaurar una copia de seguridad desde otro curso donde ya se cuente con la configuración requerida, ambas acciones deben realizarse desde el rol de Administrador (Moodle, 2021). Para la creación del curso se deben tener en cuenta los siguientes pasos:

- 1) Crear un nuevo curso: Ingresar a la opción de Administración (Site administration), en la pestaña Cursos (Courses) seleccionar la opción Agregar un nuevo curso (Add a new course) como se muestra en la Figura 56.

**Figura 56.**

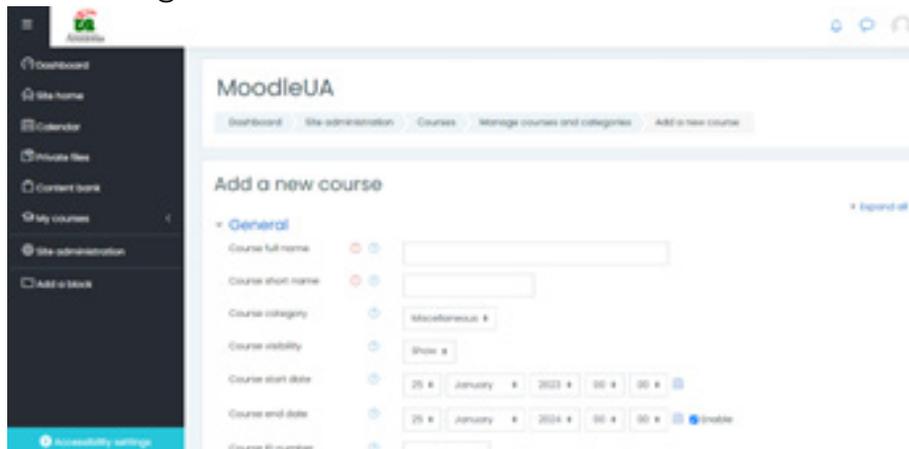
*Creación nuevo curso.*



Fuente: Elaboración propia.

- 2) Configurar los ajustes del curso: se debe diligenciar el formulario de configuración que comprende aspectos como nombre del curso, visibilidad, fechas de apertura y cierre, ID, descripción, formato, apariencia, manejo de grupos y tamaño de archivos, entre otros. Los datos de carácter obligatorio se encuentran marcados con un signo de admiración rojo y el entorno cuenta con la opción de ayudas en línea a través del signo de interrogación azul ubicado frente a cada ítem (Figura 57).

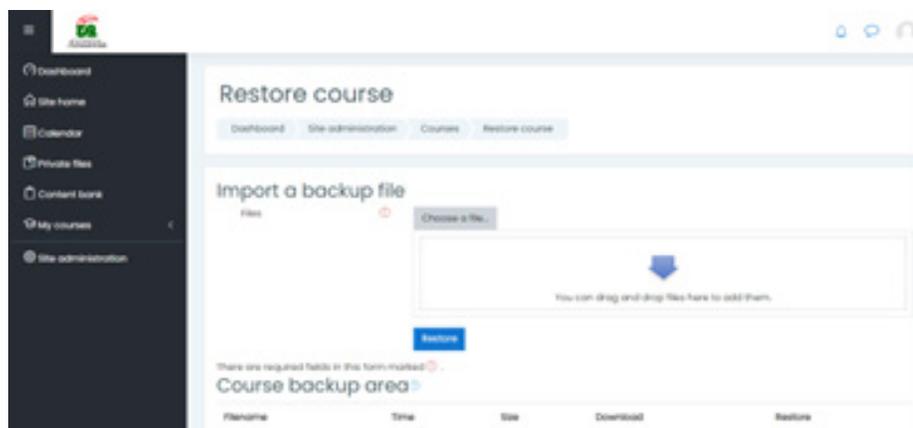
**Figura 57.**  
*Formulario de configuración.*



Fuente: Elaboración propia.

- 3) Restauración de cursos: la opción de restaurar un curso existente desde una copia requiere contar con el archivo de respaldo que debe tener la extensión .mbz. Se debe ingresar a la opción de Administración (Site administration), en la pestaña Cursos (Courses) seleccionar la opción Restaurar curso (Restore course) como se muestra en la Figura 58:

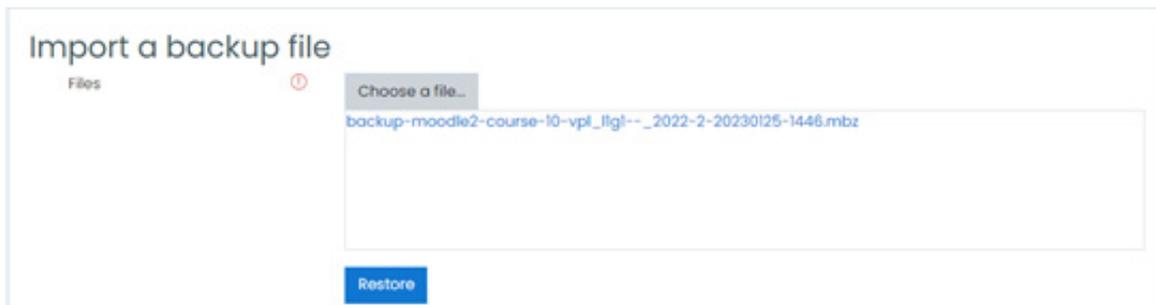
**Figura 58.**  
*Restaurar curso.*



Fuente: Elaboración propia.

Posteriormente, se debe seleccionar el archivo a restaurar (Choose a file) o realizar arrastre hasta el espacio de archivos y marcar el botón restaurar (Restore), según Figura 59 para iniciar el proceso.

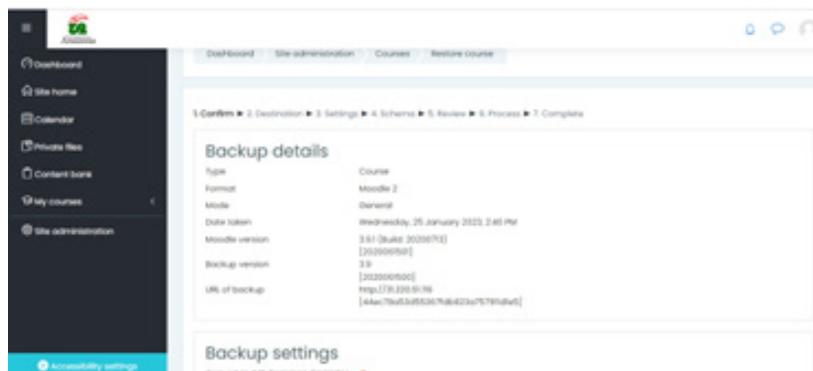
**Figura 59.**  
*Importar backup.*



Fuente: Elaboración propia.

En la Figura 60 se presenta la pantalla con los detalles del backup permitiendo de esta manera verificar el contenido del archivo antes de continuar el proceso.

**Figura 60.**  
*Detalles del backup.*

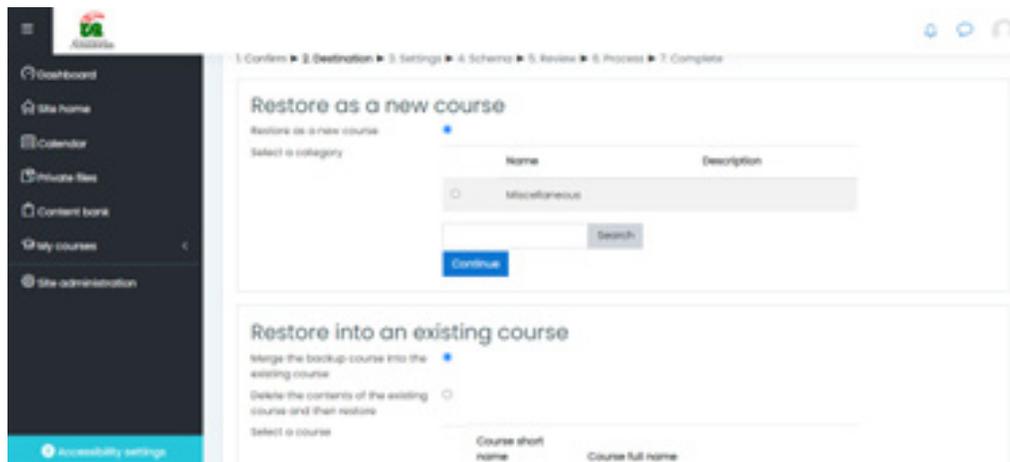


Fuente: Elaboración propia.

Moodle permite realizar la restauración como un curso nuevo, o en el caso de ya existir el curso puede reemplazarse o realizar una combinación entre ellos, Figura 61. En cada caso es necesario realizar las selecciones adecuadas, siendo un proceso sencillo e intuitivo soportado por la ayuda en línea o videos y documentación de fácil acceso disponible en Internet.

**Figura 61.**

*Restaurar nuevo curso.*



Fuente: Elaboración propia.

Una vez configurado el curso se pueden realizar otras acciones que de manera general se mencionan a continuación:

**4) Agregar módulos y recursos:** los módulos y recursos son elementos adicionales que se adicionan para enriquecer el contenido del curso. Algunos de los módulos pueden ser: foros de discusión, cuestionarios, tareas y lecciones, así como recursos como archivos, enlaces y páginas web.

**5) Configurar la evaluación:** Moodle permite configurar cuestionarios, tareas y otras actividades para que sean evaluadas automáticamente o por un profesor, para el caso específico del VPL, la explicación se dará más adelante.

**6) Personalizar la apariencia del curso:** Moodle ofrece diferentes temas que se pueden aplicar para mejorar la parte visual del curso, existen varios temas gratuitos y de pago disponibles en línea que se pueden descargar e instalar en Moodle.

**7) Agregar participantes al curso:** realizar el registro de los estudiantes, este proceso puede realizarse de varias maneras: por medio del cargue desde un archivo, realizando el registro individual (uno a uno) o activando la opción de auto-matriculación.

**8) Probar y revisar el curso:** antes de publicar el curso, es importante probar y revisar todas las actividades, módulos y recursos para asegurarse de que

funcionen correctamente y se ajusten a las necesidades del curso.

### 3.3 ESQUEMA DE EJERCICIOS PARA EL CURSO

Un aspecto que cobra relevancia es la forma como se presentan los ejercicios de práctica a los estudiantes, donde además de emplear un lenguaje sencillo para describir la situación propuesta es fundamental detallar los datos conocidos, el proceso a realizar, las condiciones que aplican y las respuestas esperadas. Siempre alineados al esquema mental de Caja Negra: ENTRADA – PROCESO - SALIDA de la Figura 62, el cual sirve como modelo para la elaboración de algoritmos y se enfatiza a lo largo de los diferentes momentos de estudio del curso.

#### Figura 62.

*Esquema de los ejercicios.*



Fuente: Elaboración propia.

Ante esta necesidad de presentar todos los elementos que le permitan al estudiante comprender los ejercicios propuestos y plantear una solución correcta, se estableció un esquema (ver Tabla 7) que comprende todos los requerimientos del VPL para adicionar los ejercicios en la plataforma de forma adecuada (Von Wangenheim, 2017).

**Tabla 7.** Esquema para la formulación de ejercicios

Requerimiento	Descripción
<b>Nombre del ejercicio</b>	<p>Este elemento permite identificar y diferenciar los ejercicios presentados de manera que en cualquier momento se puede hacer referencia a ellos y ubican al estudiante en el contexto y solución propuesta.</p>
<b>Estructura para aplicar</b>	<p>En este apartado se especifica la estructura o sentencia principal que se debe incluir al plantear la solución del ejercicio, aclarando que, aunque pueda solucionarse de otra manera es la que se recomienda, buscando así que el estudiante ponga en práctica las diferentes estructuras lógicas que se aplican en el curso.</p>
<b>Enunciado</b>	<p>Aquí se presenta la descripción general del ejercicio, donde se contextualiza el problema a resolver. Usualmente se describe una situación cotidiana o un escenario relacionado con un tema actual para facilitarle al estudiante la comprensión de la situación presentada, estableciendo las reglas y procesos que se deben cumplir de manera satisfactoria para producir las salidas esperadas del algoritmo. Para los casos que requieren la aplicación de fórmulas matemáticas se recomienda la inclusión de imágenes o tablas que faciliten describir las secuencias de operaciones.</p>
<b>Entradas</b>	<p>Esta parte especifica los datos conocidos, rangos de valores permitidos y la secuencia en que se deben ingresar. Este apartado es especialmente relevante toda vez que la evaluación automática se realiza a partir de un conjunto de datos de entrada con una estructura definida y en un orden establecido que se debe mantener.</p>
<b>Salidas</b>	<p>Aquí se describe la información esperada como respuesta del algoritmo, prestando especial atención a aspectos como el formato, redondeos, mayúsculas/minúsculas, espacios, saltos de línea, ortografía y secuencia de presentación. De forma similar a las entradas, cualquier modificación que se realice incide directamente en el resultado entregado por la evaluación automática</p>
<b>Ejemplos</b>	<p>En esta sección se presenta una simulación de la ejecución del algoritmo con los valores de entrada y las salidas esperadas de acuerdo con el enunciado; su propósito es servir como guía al momento de probar el código y como punto de partida para su depuración.</p>

## Casos de prueba

Si bien esta sección no será visible al estudiante es fundamental para la implementación en el VPL, porque contiene los diferentes escenarios con los cuales será verificado el algoritmo. Aunque no existe una fórmula o criterio que determine la cantidad de casos de prueba que se requieren para cada situación, es recomendable que se incluyan casos para verificar cada uno de los flujos que puede seguir el algoritmo y así determinar que el resultado obtenido cumple totalmente con lo solicitado en el enunciado.

Fuente: Elaboración propia.

### 3.3.1 Configuración de ejercicios (desde el montaje del docente-configuración)

Dado que el Laboratorio Virtual de Programación se despliega sobre Moodle, la creación de los ejercicios de programación se debe configurar como una actividad de la plataforma en donde se deben ingresar los parámetros necesarios para la evaluación automática de los envíos realizados por los estudiantes, a continuación, se describe el proceso el cual requiere iniciar sesión en el curso con un usuario que tenga dentro de su rol permisos de edición.

1. El primer paso es activar el modo edición el cual se encuentra al costado superior derecho de la pantalla, como se visualiza en la Figura 63

#### Figura 63.

Activar edición.



Fuente: Elaboración propia.

2. Según la configuración que se tenga del curso, ubicar la semana o tema donde se desea crear la actividad e ingresar a la opción Adicionar actividad o recurso (ver Figura 64)

**Figura 64.**

*Agregar actividad o recurso.*

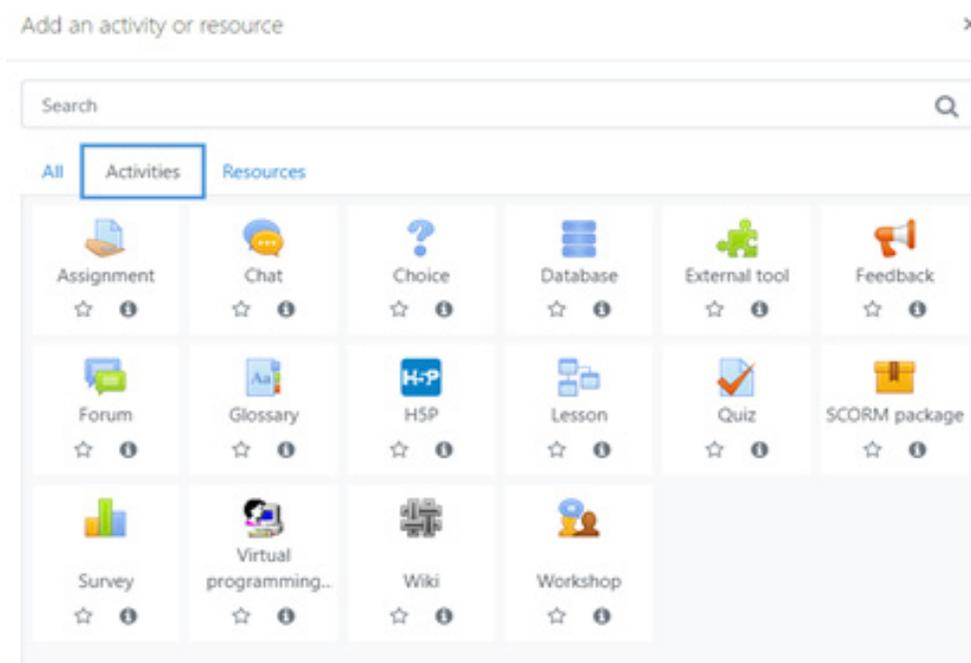


Fuente: Elaboración propia.

3. A continuación, se desplegarán todos los recursos y actividades propios de Moodle, en esta ventana nos aseguraremos de seleccionar la actividad de Laboratorio Virtual de Programación, como se visualiza en la Figura 65

**Figura 65.**

*Agregar Actividad.*



Fuente: Elaboración propia.

4. Ahora se visualizará el formulario general de las actividades en Moodle, en donde se registrará la información básica como son el nombre de la actividad (obligatorio), nombre corto y descripción (ver Figura 66).

### Figura 66.

*Creación de un ejercicio de Laboratorio virtual de programación.*



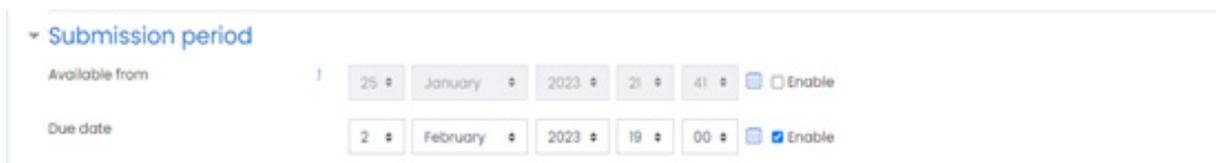
The screenshot shows the Moodle interface for adding a new Virtual programming lab. The title is "Adding a new Virtual programming lab to Conociendo la plataforma". Under the "General" section, there are three fields: "Name" (required), "Short description", and "Full description". The "Full description" field is a rich text editor with a toolbar. Below the fields is a checkbox labeled "Display description on course page".

Fuente: Elaboración propia.

Así mismo, se puede configurar la fecha desde la cual estará disponible la actividad y la fecha de cierre si se desea. No es obligatorio, es posible dejar la actividad abierta, en la Figura 67 se visualiza esta configuración.

### Figura 67.

*Configuración del periodo.*



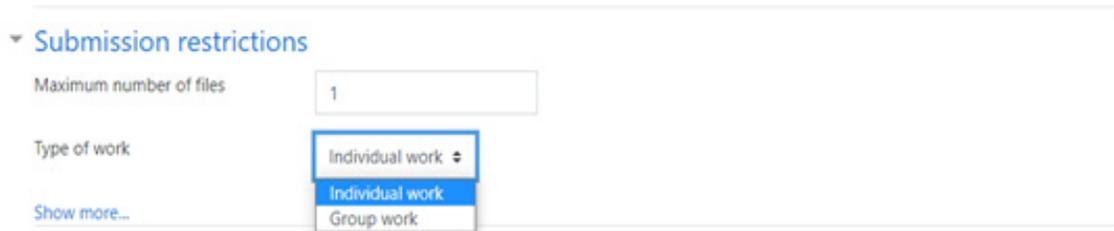
The screenshot shows the Moodle interface for configuring the submission period. The section is titled "Submission period". It has two rows: "Available from" and "Due date". Each row has dropdown menus for day, month, and year, and a time selector. There are "Enable" checkboxes for each row.

Fuente: Elaboración propia.

Una de las opciones de configuración de los ejercicios del Laboratorio Virtual de Programación, son las restricciones de los envíos que se evidencia en la Figura 68, donde se define el número máximo de archivos por ejercicio y si la entrega se realizará de manera individual o en grupo.

## Figura 68.

*Restricción de envíos.*



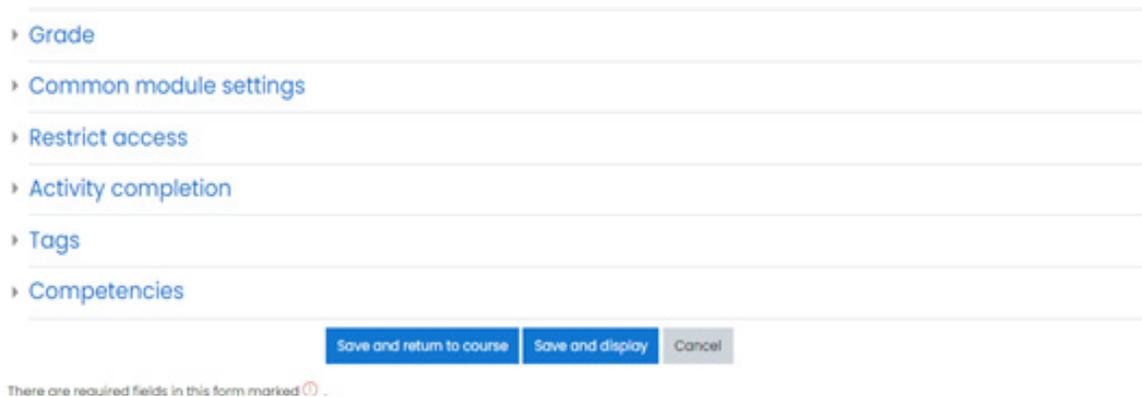
The image shows a Moodle form titled "Submission restrictions". It contains three main elements: a text input field for "Maximum number of files" with the value "1", a dropdown menu for "Type of work" currently set to "Individual work" with a sub-menu open showing "Individual work" and "Group work" options, and a "Show more..." link.

Fuente: Elaboración propia.

Así mismo, se dispone de otras opciones de parametrización como se observa en la Figura 69, las cuales están asociadas a la configuración general de actividades en Moodle, pero no están ligadas específicamente a los ejercicios de programación. De esta manera, cuando se tenga la configuración básica, es indispensable asegurarnos, de guardar y regresar al curso, o tan solo guardar y ver el ejercicio.

## Figura 69.

*Opciones de configuración.*



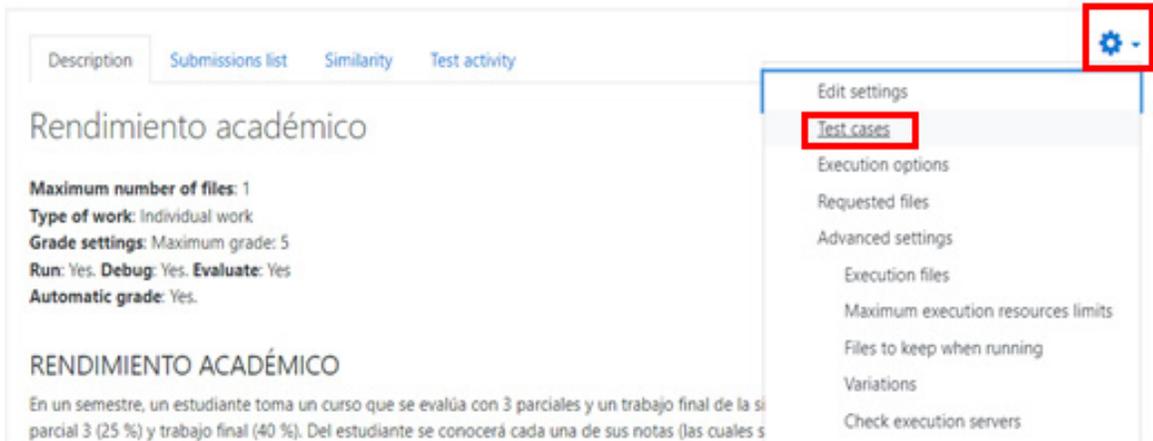
The image shows a Moodle configuration page with a list of expandable sections: "Grade", "Common module settings", "Restrict access", "Activity completion", "Tags", and "Competencies". At the bottom, there are three buttons: "Save and return to course", "Save and display", and "Cancel". A small note at the bottom left states "There are required fields in this form marked [red circle with exclamation mark]".

Fuente: Elaboración propia.

5. Una vez realizada la configuración general de la actividad, es momento de ajustar los parámetros para la evaluación del ejercicio. Para la implementación de nuestro Laboratorio Virtual se definió realizar la evaluación a través de casos de prueba. Para ello, ingresaremos a la configuración de la actividad y posteriormente Test cases según Figura 70.

**Figura 70.**

*Opciones de configuración.*



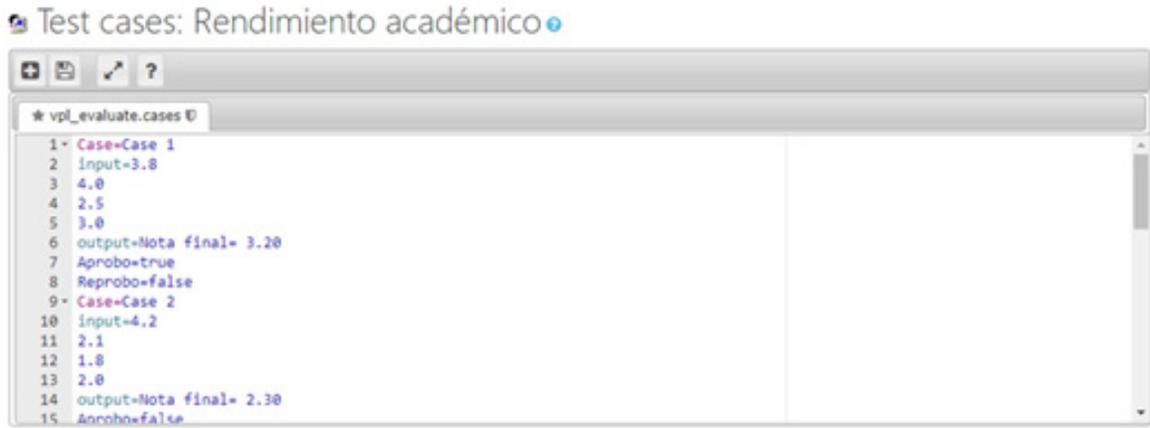
Fuente: Elaboración propia.

6. Los casos de prueba permiten al docente establecer diferentes escenarios para medir las soluciones desarrolladas por los estudiantes a través de la definición de parámetros de entrada y de salida que deben cumplirse para obtener una evaluación satisfactoria.

En la Figura 71 se presenta la ventana de configuración donde cada caso se identifica con un encabezado que contiene la palabra en Inglés case seguida de un nombre, en nuestro caso utilizamos numeración: case 1, case 2 ... case n. Para los datos de entrada se ubicará la palabra input junto al símbolo de igual (=) y a continuación todas las entradas necesarias, guardando cuidado que correspondan en cantidad, orden y tipo con la descripción del ejercicio. Para la salida, se escribirá, output junto al símbolo de igualdad (=), acompañado de la línea o líneas que debe arrojar la solución. Cada caso finaliza con el registro de un nuevo encabezado. Para finalizar, se debe presionar el botón guardar ubicado en la barra superior.

## Figura 71.

Creación de casos de prueba.



```
* vpl_evaluate.cases 0
1- Case-Case 1
2 input=3.8
3 4.0
4 2.5
5 3.0
6 output=Nota final= 3.20
7 Aprobo=true
8 Reprobo=false
9- Case-Case 2
10 input=4.2
11 2.1
12 1.8
13 2.0
14 output=Nota final= 2.30
15 Aprobo=false
```

Fuente: Elaboración propia.

7. El componente del output requiere especial atención pues como se ha mencionado, establece la respuesta correcta del programa ejecutado para la entrada de cada caso de prueba.

La plataforma cuenta con cuatro tipos de opciones de salida: números, texto, texto exacto y expresión regular. Cuando se trata de números (enteros o flotantes) la evaluación elimina los textos y compara los números resultantes con la salida establecida. Para el caso de texto se realiza una comparación de texto no estricta (ignora mayúsculas, minúsculas, espacios, tabulaciones, nueva líneas y signos de puntuación). Para un chequeo de texto exacto el output debe escribirse entre comillas dobles. Finalmente, las expresiones regulares funcionan de la misma manera que en JavaScript aplicando el estándar POSIX.

8. De manera opcional pueden incluirse en cada caso otros aspectos para fortalecer las pruebas, entre los que se encuentran penalizaciones por casos no satisfactorios utilizando la cláusula “Grade reduction = [ value | percent% ]”, mensajes personalizados cuando el caso de prueba falla al incluir la línea “Fail message = message”, ejecutar otros programas para verificar otras características del envío usando “Program to run = path” o envío de datos como argumentos en la línea de comando con “Program arguments = arg1 arg2 ...”.

### **3.3.2 Presentación del caso de prueba (temáticas, tiempo, grupos, ejercicios, requerimientos)**

A manera de ejemplo se presenta el paso a paso realizado para la implementación de un ejercicio tipo VPL.

El ejercicio propuesto corresponde al capítulo de estructuras **SECUENCIALES** y tiene por objetivo validar el uso de operadores matemáticos y relacionales.

**Nombre del ejercicio:** Rendimiento académico

**Estructura para aplicar:** Secuencial

Enunciado: En un semestre, un estudiante toma un curso que se evalúa con 3 parciales y un trabajo final de la siguiente manera: parcial 1 (15 %), parcial 2 (20 %), parcial 3 (25 %) y trabajo final (40 %). Del estudiante se conocerá cada una de sus notas (las cuales siempre estarán entre 0.0 y 5.0), considerando los valores iguales o superiores a 3.0 como aprobados. El algoritmo debe mostrar la nota definitiva de la asignatura e informarle al estudiante si ha sido aprobado (Verdadero - Falso) o reprobado (Verdadero - Falso). (Ver Figura 72)

**Figura 72.**

Visualización del ejercicio en el editor.



Fuente: Elaboración propia.

**Entradas:** La entrada contiene cuatro líneas, una por cada nota del estudiante.

**Salidas:** Para cada caso de prueba imprima una línea con la nota definitiva redondeada a dos decimales, una línea para indicar si aprobó y otra para indicar si reprobó (ver Figura 73).

**Figura 73.**

Ejemplos de Entrada – Salida.

ENTRADA	SALIDA
1.5 2.8 5.0 3.5	Nota final= 3.44 Aprobo= true Reprobo= false
4.2 2.1 1.8 2.0	Nota final= 2.30 Aprobo= false Reprobo= true

Fuente: Elaboración propia.

**Casos de prueba:** A continuación, se presentan algunos casos de prueba y su correspondiente respuesta.

En este caso de prueba se configuró una salida de texto flexible la cual en la línea 6 (ver Figura 74), presenta el formato de salida con la letra inicial en mayúsculas, sin embargo, la propuesta genera un texto en mayúsculas fijas que al ser evaluado da por aceptada la solución (ver Figura 75).

### Figura 74.

*Ejemplo caso de prueba.*

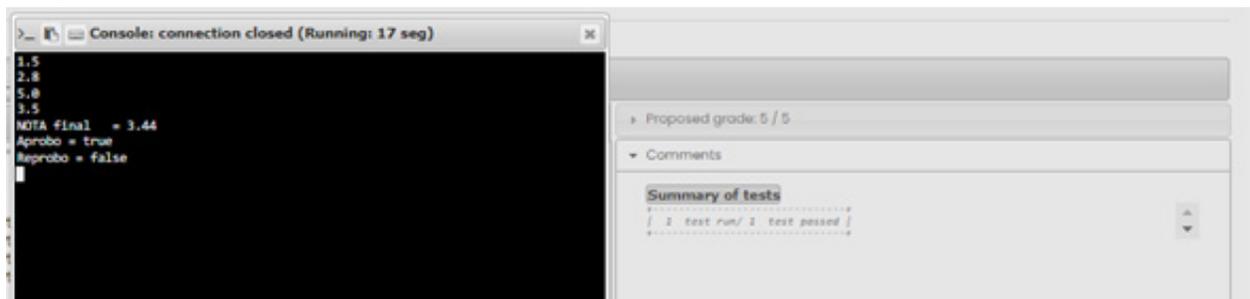


```
1- Case=Case 1 - aprobado flexible
2 input=1.5
3 2.8
4 5.0
5 3.5
6 output=
7 Nota final=3.44
8 Aprobo=true
9 Reprobo=false
```

Fuente: Elaboración propia.

### Figura 75.

*Solución de salida.*

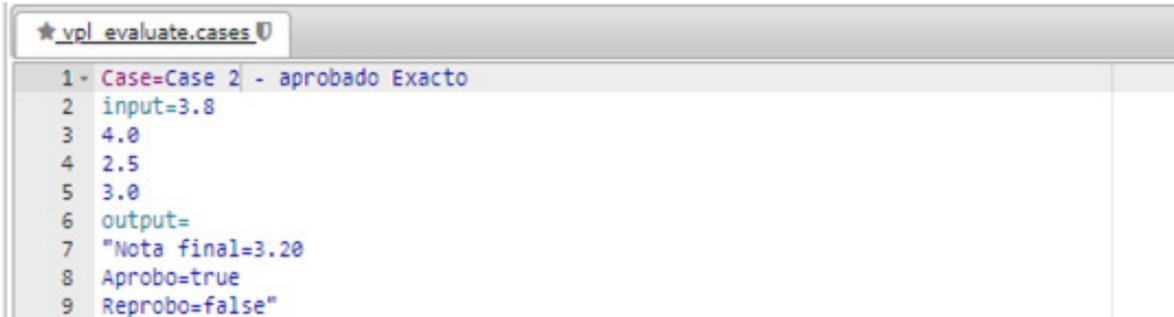


Fuente: Elaboración propia.

Para este segundo caso de prueba con texto exacto (por usar la comilla doble en el output) (ver Figura 76), la evaluación del VPL es incorrecta, pues detecta la palabra en mayúsculas fija, (ver figura).

**Figura 76.**

*Segundo caso de prueba.*

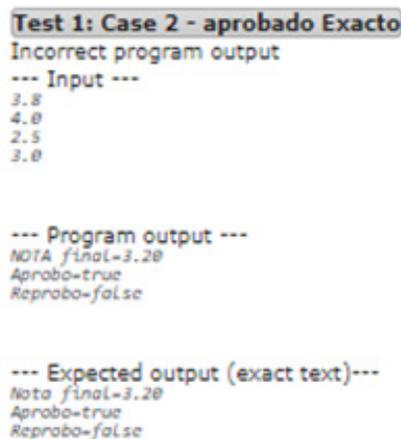


```
★ vpl evaluate.cases 0
1 - Case=Case 2 - aprobado Exacto
2 input=3.8
3 4.0
4 2.5
5 3.0
6 output=
7 "Nota final=3.20
8 Aprobo=true
9 Reprobo=false"
```

Fuente: Elaboración propia.

**Figura 77.**

*Entrada -salida caso de prueba 2.*



```
Test 1: Case 2 - aprobado Exacto
Incorrect program output
--- Input ---
3.8
4.0
2.5
3.0

--- Program output ---
NOTA final=3.20
Aprobo=true
Reprobo=false

--- Expected output (exact text)---
Nota final=3.20
Aprobo=true
Reprobo=false
```

Fuente: Elaboración propia

### **3.4 RECOMENDACIONES PARA EL USO DEL VPL**

Con el propósito de lograr una implementación adecuada del VPL, se sugiere considerar algunas recomendaciones que han surgido a partir de la experiencia recopilada con respecto al diseño y aplicación de ejercicios y casos de prueba orientados a alcanzar los resultados de aprendizaje, tanto desde la perspectiva del docente como la del estudiante, con esto se pretende dar unas orientaciones más precisas para futuras intervenciones.

### 3.4.1 Perspectiva del docente

Diseño de los ejercicios, lo más importante y recomendable es escribir el enunciado con un lenguaje sencillo y presentar contextos pertinentes para la edad de los estudiantes, entre estos se pueden tener en cuenta: gustos, intereses y vivencias, de manera que sea más fácil la comprensión de cada situación particular y por ende proponer una solución eficiente a la problemática presentada. De igual forma, es necesario resaltar que la complejidad de los ejercicios debe ser incremental, buscando en primera medida que los primeros sirvan para que los estudiantes logren familiarizarse con la plataforma y la estructura de los problemas y posteriormente aumentar el nivel en correspondencia con las estructuras a abordar.

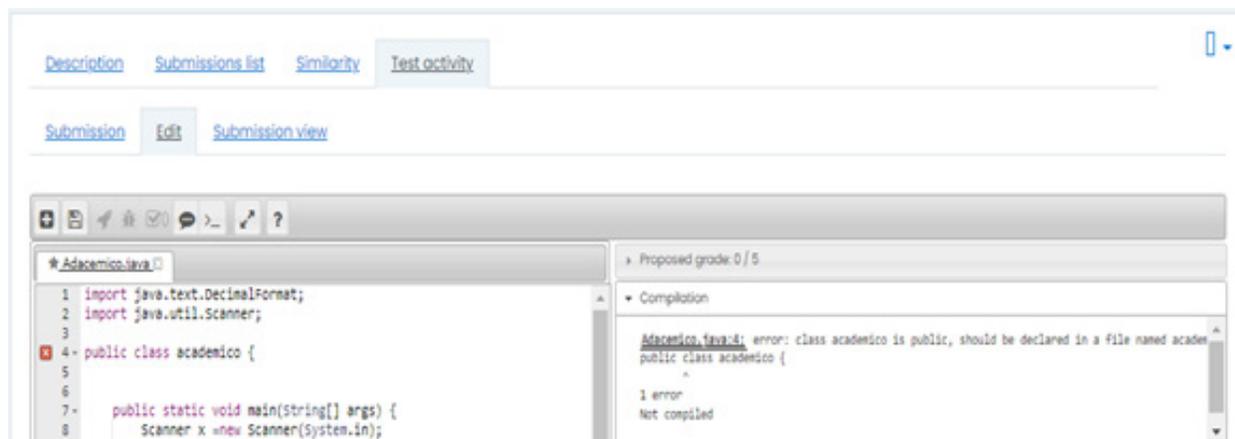
Administración de los cursos, a través de los diferentes periodos académicos es recomendable activar la gestión de grupos, de manera que los estudiantes antiguos cuenten con acceso al material en todo momento sin interferir con el curso actual.

### 3.4.2 Perspectiva del estudiante

Cuando se utilice la opción Edit para realizar el envío del código, es necesario asegurarse de colocar el nombre correcto del archivo y la extensión correspondiente al lenguaje utilizado, en este caso por ser JAVA debe ir igual al nombre de la clase y la extensión .java, de no hacerlo se generará un error en la plataforma tal como se muestra en la Figura 78.

**Figura 78.**

*Datos de la entrega.*

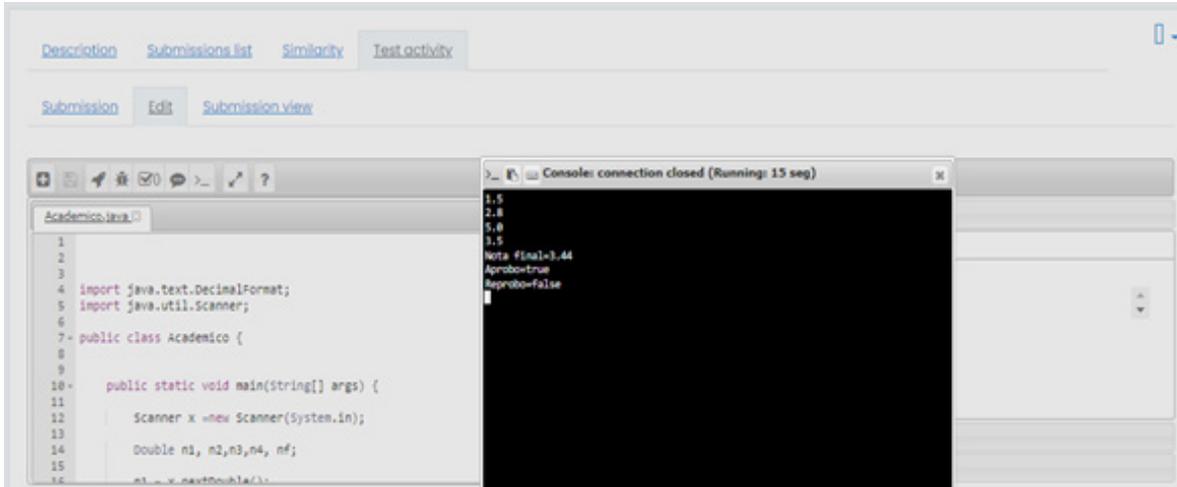


Fuente: Elaboración propia.

Para realizar la verificación de la solución propuesta, se puede recurrir a dos opciones, la primera consiste en hacerlo directamente en el entorno de desarrollo utilizado, la otra es por medio de la opción Run disponible en la plataforma, tal como se muestra en la Figura 79.

### Figura 79.

*Verificación de la solución*



Fuente: Elaboración propia.

### 3.5 CONCLUSIONES

La experiencia de configurar el VPL ha sido un proceso extenso aunque satisfactorio, teniendo en cuenta que tanto el diseño del ejercicio bajo el esquema establecido, como el planteamiento de los casos de prueba requieren de bastante tiempo de preparación, sin embargo en la medida que se elaboran más ejercicios, temáticas y se aumenta la complejidad de los mismos, se logra constituir un repositorio que puede ser usado posteriormente sin tener que iniciar de nuevo el proceso, haciendo más fácil las futuras intervenciones.

Usar VPL en la enseñanza de los fundamentos de programación tiene múltiples ventajas para favorecer los aprendizajes de los estudiantes, dentro de estas se destacan:

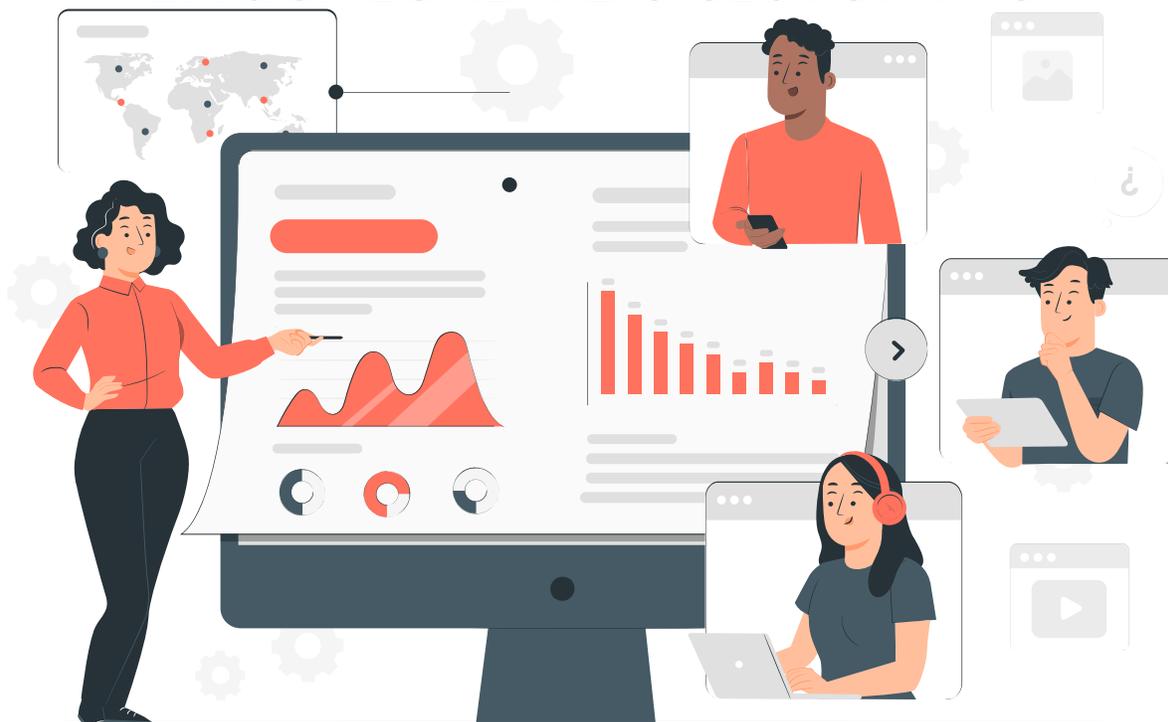
- **Fácil acceso:** los estudiantes pueden ingresar al VPL, en cualquier momento y lugar para practicar, en este sentido juega un papel vital el diseño del ambiente de aprendizaje que haya realizado el docente para apoyar el trabajo autónomo de los estudiantes.
- **Flexibilidad y variedad de recursos:** debido a que el VPL se integra a Moodle, proporciona diferentes herramientas y recursos que permiten al docente establecer diferentes actividades para fortalecer la comunicación, el trabajo colaborativo y por ende el aprendizaje.
- **Retroalimentación inmediata:** es fundamental que tanto docentes como estudiantes logren identificar los errores en el momento que se presentan para así poder tomar decisiones oportunas. A través del VPL, es posible conocer los errores del código de manera inmediata, esto da la posibilidad de centrar la atención en dónde es importante para realizar las correcciones pertinentes.
- **Actualización constante:** el hecho de que tanto Moodle como el VPL, sean de código abierto, permite vincular a muchas personas en el mejoramiento tanto de la plataforma de aprendizaje como en la actualización de los lenguajes y tendencias de programación.

### 3.6 REFERENCIAS BIBLIOGRÁFICAS

- Castañeda, J. A., Ortega Álvarez, C. I., & Galvis Ibarra, J. C. I. (2019). Docke-  
rizando Un Laboratorio Virtual De Programación (Vpl) Y Moodle En  
Google Cloud. Encuentro Internacional de Educación En Ingeniería, SE-  
<https://acofipapers.org/index.php/eiei/article/view/212>
- Moodle. (2021). Guía rápida del profesor - MoodleDocs. Moodle.org. [https://  
docs.moodle.org/all/es/Gu%C3%ADa\\_r%C3%A1pida\\_del\\_profesor](https://docs.moodle.org/all/es/Gu%C3%ADa_r%C3%A1pida_del_profesor)
- Muñoz, W. M., León, A. M., Nogueira, Y. E. M., & Mora, G. R. V. (2021). Moodle:  
Entorno Virtual para el fortalecimiento del aprendizaje autónomo. Revis-  
ta UNIANDES Episteme, 8(1), 137-152.
- Rade, L. Y. V., Alcívar, M. V., & Gangotena, M. W. T. (2021). La plataforma  
Moodle como ambiente de aprendizaje de estudiantes universitarios. Re-  
vista Publicando, 8(31), 61-70.
- Ramos, V. F. C., Cechinel, C., Magé, L., & Lemos, R. (2021). Student and Lectu-  
rer Perceptions of Usability of the Virtual Programming Lab Module for  
Moodle. Informatics in Education, 20(2), 1-19. [https://doi.org/10.15388/infe-  
du.2021.14](https://doi.org/10.15388/infe-<br/>du.2021.14)
- Von Wangenheim, A. (2017). Developing Programming Courses with Moodle  
and VPL. Bookess Editora LTDA-ME.

# CAPÍTULO IV

## CASO ESTUDIANTES INGENIERÍA DE SISTEMAS



### **John Arley García Quintero**

Magister en E-learning y Redes Sociales. Universidad Internacional de la Rioja. Estudiante del Doctorado en Doctorado en Ciencias de la Educación con Énfasis en: Investigación, Evaluación y Formulación de Proyectos Educativos de la Universidad Metropolitana de Educación, Ciencia y Tecnología (UMECIT). Profesor categoría Asociado, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0002-5530-1746>

### **Elio Fabio Sánchez Trujillo**

Magister Maestría en Matemáticas Aplicadas. Universidad EAFIT. Profesor categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0001-6845-6870>

### **Denis Lorena Álvarez Guayara**

Magister en proyectos mediados por TIC. Universidad de la Sabana. Estudiante del Doctorado en Educación y Cultura Ambiental, Universidad de la Amazonia. Profesora categoría Asistente, Facultad de Ingeniería de la Universidad de la Amazonia. Integrante del grupo de Investigación en Informática, Innovación y Tecnología- GITUA del Programa Ingeniería de Sistemas de la Universidad de la Amazonia. ORCID: <https://orcid.org/0000-0001-8939-8139>

### **CITACIÓN SUGERIDA:**

García Quintero, J, A. Sánchez Trujillo, E, F. Álvarez Guayara, D, L. (2024). Caso estudiantes Ingeniería de Sistemas. Planteamiento del problema. Laboratorio virtual para el aprendizaje de fundamentos de programación. Editoria. Universidad de la Amazonia. p.p 121

## 4.1 PRESENTACIÓN

En el presente capítulo se presenta el caso de implementación del Laboratorio Virtual de Programación (VPL) en el programa de Ingeniería de Sistemas de la Universidad de la Amazonia, específicamente en el curso de introducción a la programación de computadores denominado Lógica y Algoritmos I, donde históricamente se han presentado altos niveles de pérdida y repitencia (Claros-Moreno, et al., 2019). Así mismo, se presenta la población en la cual se realizó la intervención, describiendo los ejercicios diseñados para la prueba y la percepción de los estudiantes en el proceso de interacción con el VPL. Gracias a la implementación fue posible obtener información importante para el mejoramiento de las estrategias de aprendizaje y la planeación de actividades prácticas en el ambiente de aprendizaje, orientadas al desarrollo de las habilidades en programación de los estudiantes, tan requeridas y demandadas en la actualidad.

La enseñanza de la programación de computadores es un desafío que deben enfrentar los docentes de carreras relacionadas con la ingenierías de sistemas, software y afines (Alammary et al., 2012 y Cheah, 2020); así mismo, los cursos iniciales de programación representan para muchos docentes un esfuerzo adicional en su quehacer, ya que el seguimiento, corrección y orientación que se debe realizar a los ejercicios enviados por los estudiantes es mayor (Cardoso et al., 2021), y en muchos casos no se logra llevar a cabo una retroalimentación adecuada que favorezca los aprendizajes, que para el caso específico de la programación es indispensable además, desarrollar procesos cognitivos como la abstracción y la resolución de problemas (Insuasti, 2016), habilidades fundamentales y que muchos estudiantes no logran desarrollar antes de iniciar con el aprendizaje de programación, lo que les representa serias problemáticas a lo largo de sus carreras (Byrne y Lyones, 2001, Gomes y Mendes, 2014 y Savage y Piwek, 2019);

Reconociendo previamente las problemáticas relacionadas con los procesos de enseñanza y aprendizaje, surge la consideración sobre el uso de herramientas tecnológicas que apoyen al docente en la aplicación de la evaluación y retroalimentación a los estudiantes, es por esto que en el programa de Ingeniería de Sistemas de la Universidad de la Amazonia, se implementó el módulo de Moodle denominado VPL para desarrollar estas actividades de forma eficiente, pero también, permitir el fortalecimiento de las habilidades de programación mediante el envío de ejercicios y la posterior depuración de

los mismos, de acuerdo a la información suministrada por la herramienta en tiempo real, y que es un insumo para que el docente pueda tomar oportunas decisiones pedagógicas respecto a las dificultades de los estudiantes.

## **4.2 POBLACIÓN**

La población estuvo conformada por dos grupos de estudiantes del programa de Ingeniería de Sistemas de la Universidad de la Amazonia, uno de ellos caracterizado por haber cursado la asignatura de Lógica y Algoritmos I, por lo menos una vez, y el otro grupo por cursar por primera vez el espacio académico. En el grupo de observación total participaron de manera voluntaria 58 estudiantes y la intervención práctica se realizó en el segundo semestre del año 2022.

## **4.3 PRESENTACIÓN DEL CASO DE PRUEBA**

### **4.3.1 Temática**

En el proceso de intervención en el laboratorio virtual de programación se trabajó con ejercicios de estructuras secuenciales, condicionales y cíclicas, ya que es el segmento de fundamentación para el aprendizaje de la programación y en donde se debe enfatizar y fortalecer el esquema de entrada, salida y proceso, indispensable para el aprendizaje de la resolución de problemas algorítmicos. Para la selección de los ejercicios se tomó en cuenta la realización de procesos básicos matemáticos para que el estudiante fortaleciera la creación de un algoritmo por medio de un lenguaje de programación y se familiarizara con el entorno del laboratorio virtual.

Para las prácticas a realizar se seleccionan y clasifican ejercicios en complejidad creciente de acuerdo con la siguiente jerarquía:

### **Ejercicios introductorios:**

- Saludo
- Datos personales
- Fecha de nacimiento

### **Ejercicios secuenciales:**

- Los inversionistas
- Área de un rectángulo
- Área y perímetro de un círculo
- El almacén
- Rendimiento académico
- PH
- Índice de masa corporal -IMC

### **Ejercicios con estructuras condicionales:**

- Mayor de edad
- ¿Aprobado?
- ¿Par o impar?
- Almacén El Camaleón
- Dos Números
- Los inversionistas 2
- Vocales o consonantes
- Tablero de ajedrez
- Centro recreacional - descuentos
- Manejo de SubString

### **Ejercicios con estructuras cíclicas:**

- Múltiplos de un número
- Busca letras
- Cuenta Números
- Palabritas
- Academia programar
- Polidivisible
- Escalera – 1
- Escalera – 2
- Factorial de un número
- Banco Ganamas
- Sala de belleza CRESPOS

### 4.3.2 Tiempos de trabajo

La intervención realizada con el Laboratorio Virtual de Programación para el curso de Lógica y Algoritmos I se realizó en el marco de los tiempos de trabajo académico del estudiante de acuerdo con lo establecido en el estatuto estudiantil de la Universidad, distribuidos de la siguiente forma:

**Trabajo directo:** Es el momento en trabajo en el cual, el docente realiza las orientaciones teóricas generales de la temática y las instrucciones de la práctica académica. En este espacio el docente presenta y guía la clase con relación a la plataforma educativa utilizada, los recursos educativos a disposición y los ejercicios de referencia para la realización de las diferentes prácticas del Laboratorio Virtual.

**Trabajo dirigido:** Es el momento en el cual, el docente brinda tutoría y orientación frente a las actividades que deben realizar los estudiantes para el desarrollo de la práctica académica. En este espacio el docente adicionalmente realiza retroalimentación grupal sobre los ejercicios que se realizaron en clase sobre el Laboratorio Virtual de Programación.

**Trabajo independiente:** Es el momento donde el estudiante, en su tiempo libre, sea dentro de la universidad o fuera de ella, destina una parte para realizar actividades adicionales a las programadas en clase. En este espacio el estudiante debe fortalecer los conceptos aprendidos, interactuar con los recursos educativos puestos a su disposición y reforzar las habilidades de programación mediante la realización de ejercicio en el laboratorio de programación.

De esta manera, la intervención del caso de prueba se realizó en 3 meses, en donde cada mes se desarrolló una unidad: estructura secuencial, estructuras condicionales (*if*, *if-else*, *if-else if* y *switch*) y estructuras cíclicas (*for*, *while* y *do - while*).

### 4.3.3 Análisis de los ejercicios enviados en la práctica del Laboratorio Virtual de Programación

Los ejercicios se crean a partir del esquema Entrada, Proceso y Salida. Este esquema viene acompañado de la descripción general del ejercicio en donde se contextualiza el problema a resolver. Posteriormente, se presenta la forma en cómo se debe ingresar los datos y las salidas que se espera tener luego de la

elaboración del algoritmo. Así mismo, se presentan ejemplos con los valores de las entradas y las salidas esperadas de acuerdo con cada ejercicio.

A continuación, se describe la forma de visualizar los envíos realizados por los estudiantes y la cantidad enviada por los mismos para el caso específico del ejercicio denominado Saludo.

Una vez los estudiantes han realizado los envíos de soluciones al ejercicio, para visualizarlos, es necesario ingresar a cada ejercicio y luego dar clic, en la opción Submissions list, tal como se puede apreciar en la Figura 80.

### Figura 80.

*Envíos realizados*



Fuente: Elaboración propia.

A continuación, se despliega la información del estudiante, el número de envíos realizados (**Submissions**), la calificación obtenida (**Grade**) o si está no ha sido calificada por contener algún error de configuración (**No grade**), así mismo, se informa si la calificación fue realizada de manera automática (**Automatic grade**) o si el docente realizó la revisión de los envíos y su respectiva calificación, tal como se puede visualizar en la Figura 81.

**Figura 81.**

*Información del estudiante relacionado con los envíos*

	Surname	Submitted on	Submissions	Grade	Evaluator	Evaluated on
1	Andres Silven Alvarez Vera		8	0.00 / 5.00	Automatic grade	
2	Anderson Arias Calderon		4	5.00 / 5.00	Automatic grade	
2	Kevin Andres Cadenas Cuelgar		2	No grade		
4	Ivan Darío Carrvajal Reina		12	2.00 / 5.00 (0.00000)	John Arley Garcia	
2	Juan David Chaux Montaa		1	5.00 / 5.00	Automatic grade	
4	Oliver Alexis Farran		1	No grade		

Fuente: Elaboración propia.

Si el docente desea revisar cada uno de los envíos, se debe dar clic en los números que se ubican en la columna de **submissions** y es posible revisar el detalle de cada uno.

Durante el proceso de la práctica del laboratorio virtual de programación, se realizaron 3 ejercicios de entrada y salida de datos, ejercicios introductorios los cuales se implementaron para dar a conocer el entorno de la plataforma y la manera en cómo los estudiantes podían cargar un archivo, editarlo, ejecutarlo y finalmente evaluarlo. En la Tabla 8, es posible visualizar la cantidad de envíos realizados en los 3 ejercicios de contextualización de la plataforma, logrando adicionalmente evidenciar una suma total de 1041 envíos por los dos grupos participantes.

**Tabla 8.**

*Ejercicios introductorios*

Nombre del ejercicio	Número de envíos	
	Grupo 1	Grupo 2
Saludo	232	106
Datos personales	366	109
Fecha de nacimiento	130	98

Fuente: Elaboración propia.

Así mismo, en la Tabla 9, se logra visualizar los 7 ejercicios relacionados con las estructuras secuenciales, los cuales además de trabajar las entradas y las salidas de datos, se implementaron ejercicios en los cuales debían elaborar algoritmos para el cálculo del área de algunas figuras geométricas, así mismo, se relacionaron ejercicios de contextos reales y cercanos a los estudiantes donde estos debían calcular porcentajes de descuento, calcular promedios y así como implementar los diferentes operadores lógicos, matemáticos y relacionales. En total los estudiantes de los dos grupos participantes realizaron 1736 envíos al Laboratorio Virtual de Programación.

**Tabla 9.**  
*Ejercicios secuenciales*

<b>Nombre del ejercicio</b>	<b>Número de envíos – Grupo 1</b>	<b>Número de envíos – Grupo 2</b>
Los inversionistas	189	195
Área de un rectángulo	89	95
Área y perímetro de un círculo	105	41
El almacén	232	44
Rendimiento académico	174	108
PH	184	61
Índice de masa corporal -IMC	86	133

Fuente: Elaboración propia

En la Tabla 10, es posible evidenciar los ejercicios implementados para las estructuras condicionales, así como el número de envíos realizados por cada grupo en cada ejercicio. En total se realizaron 1542 envíos y en esta unidad se plantearon ejercicios en donde los estudiantes debían implementar algoritmos utilizando estructuras como el if, if-else, if-else if y Switch.

**Tabla 10.***Ejercicios de estructuras condicionales.*

<b>Nombre del ejercicio</b>	<b>Número de envíos – Grupo 1</b>	<b>Número de envíos – Grupo 2</b>
Mayor de edad	125	44
¿Aprobado?	134	47
¿Par o impar?	65	24
Almacén El Camaleón	55	67
Dos Números	86	75
Los inversionistas 2	81	64
Vocales o consonantes	139	89
Tablero de ajedrez	59	51
Centro recreacional - descuentos	75	40
Manejo de SubString	149	73

Fuente: Elaboración propia.

Así mismo, en la Tabla 11 es posible evidenciar los ejercicios realizados por los estudiantes, retos o tareas caracterizadas por tener que implementar las estructuras cíclicas como for, while y el do-while. Igualmente, estos ejercicios relacionan la implementación de estructuras condicionales como el if, if-else, if-else if y el Switch. En total, en esta tercera unidad, los estudiantes realizaron 1787 envíos al laboratorio virtual de programación.

**Tabla 11.***Ejercicios de estructuras cíclicas*

<b>Nombre del archivo</b>	<b>Número de envíos – Grupo 1</b>	<b>Número de envíos – Grupo 2</b>
Múltiplos de un número	110	82
Buscalettras	59	75
Cuenta Números	73	43
Palabritas	66	57
Academia programar	170	207

Polidivisible	126	48
Escalera - 1	48	31
Escalera - 2	16	17
Factorial de un número	21	25
Banco Ganamas	171	200
Sala de belleza CRESPOS	72	70

Fuente: Elaboración propia.

Finalmente, la Tabla 12 muestra la cantidad de envíos totales a lo largo de la práctica del laboratorio de programación destacando el gran número de ejercicios enviados por los estudiantes, tan solo para los ejercicios introductorios se registraron 1040 envíos, ejercicios secuenciales 1736, para los ejercicios con condicionales 1542 y finalmente, para los ejercicios con estructuras cíclicas se enviaron 1787.

**Tabla 12.**

*Número de ejercicios por unidad*

<b>Unidad</b>	<b>Número de ejercicios enviados</b>
Ejercicios introductorios	1041
Ejercicios secuenciales	1736
Estructuras condicionales	1542
Estructuras cíclicas	1787

Fuente: Elaboración propia.

La cantidad de envíos por unidad es bastante representativa y más allá de su magnitud, es posible evidenciar el trabajo de los estudiantes al realizar sus envíos y de la retroalimentación inmediata que brinda la plataforma. Así, se comprueba el gran apoyo que representa para los docentes de la asignatura de Lógica y Algoritmos I, plataformas como el Laboratorio Virtual de Programación implementado, ya que permite a los docentes realizar retroalimentaciones generales de los ejercicios, y centrar la atención en los estudiantes que presentan mayores dificultades de aprehensión. En contraposición, para estudiantes que no presentan inconvenientes con las soluciones, les brinda la oportunidad de avanzar en la resolución de los demás ejercicios publicados en el Laboratorio, fomentando de esa manera, autonomía en su proceso de aprendizaje.

#### **4.4 PERCEPCIÓN DE LOS ESTUDIANTES RESPECTO A LA INTERACCIÓN CON EL LABORATORIO VIRTUAL DE PROGRAMACIÓN A PARTIR DE LOS CASOS DE PRUEBA.**

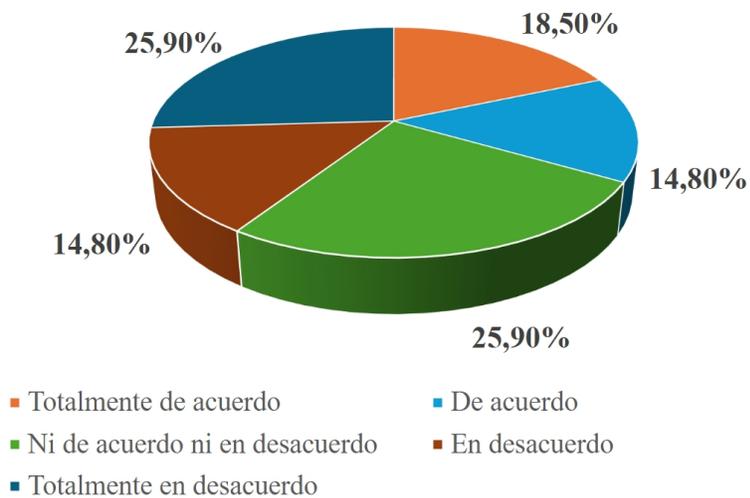
Para conocer las apreciaciones de los estudiantes con relación al uso de la plataforma VPL como estrategia didáctica mediada por las TIC para el fortalecimiento de las habilidades relacionadas con los fundamentos de la programación de computadores, el equipo de investigadores diseñó un instrumento de recolección de datos el cual recogió las apreciaciones del estudiantado con relación al apoyo que ofrece el laboratorio en el proceso de aprendizaje de la programación de computadores y la interacción con la plataforma. A continuación, se relacionarán los resultados obtenidos:

##### ***4.4.1 Interacción con la plataforma.***

En el primer esquema del instrumento relacionado con la usabilidad del Laboratorio Virtual de Programación, la primera pregunta fue, ¿Considera sencillo el ingreso a la plataforma?, los resultados se presentan en la Figura 82 y reflejan que para el 33,3% de los estudiantes, el ingreso a la plataforma fue sencillo ya que lo reflejaron en las opciones de respuesta de totalmente de acuerdo y de acuerdo respectivamente, frente a un 40,7% de los estudiantes, los cuales indicaron que estaban de desacuerdo y totalmente en desacuerdo. Así mismo, el 25.9% de los estudiantes indicaban que no estaban ni de acuerdo ni en desacuerdo.

**Figura 82.**

*¿Considera sencillo el ingreso a la plataforma?*

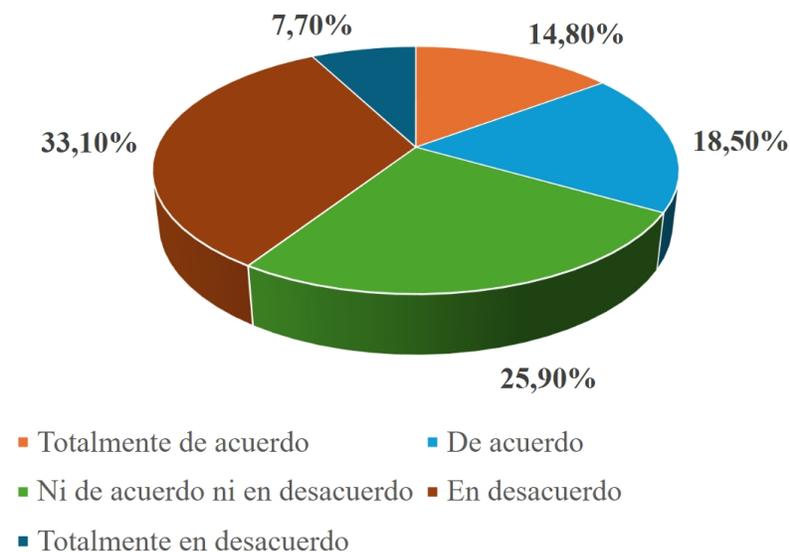


Fuente: Elaboración propia.

Los datos visualizados en la Figura 82, muestran que uno de los aspectos por mejorar del Laboratorio Virtual de Programación es la facilidad de ingreso, ya que más del 66.6% de los estudiantes, lo que representa dos terceras partes de la población, tienen posturas imparciales o negativas frente al acceso, lo que puede representar inconvenientes en el uso de la solución planteada.

**Figura 83.**

*¿El acceso a los ejercicios está disponible de forma adecuada?*



Fuente: Elaboración propia.

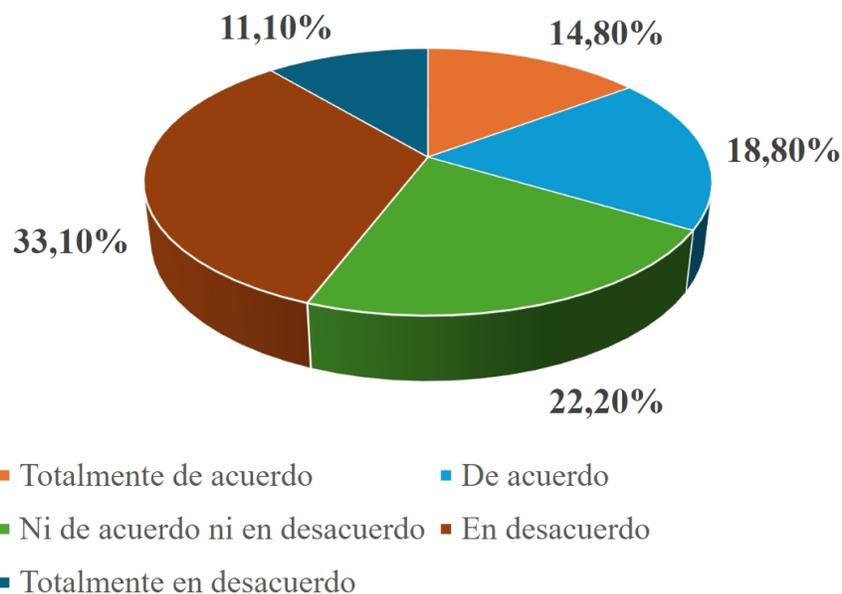
Por otro lado, otra de las preguntas realizadas a los estudiantes fue, ¿El acceso a los ejercicios está disponible de forma adecuada? a lo que el 33% de los encuestados respondieron de forma positiva con posturas de estar de acuerdo y totalmente de acuerdo respectivamente, así mismo, un 25,9% indicó imparcialidad, sin embargo, un 41% plantea posturas negativas al responder que se encontraban en desacuerdo y totalmente en desacuerdo.

La Figura 83 refleja que menos del 35% de los estudiantes consideran que el acceso a los ejercicios no se encuentra de forma adecuada, lo que indica que la mayoría de ellos, tuvieron inconvenientes en el momento de ingresar a las tareas a resolver.

Así mismo, se les preguntó a los estudiantes si era clara la forma de subir el archivo del código desarrollado en el IDE de programación y el 55% de los estudiantes no lo percibieron de forma conveniente, el 11,1% no estuvo de acuerdo, sin embargo, tampoco en desacuerdo y el 33,3% respondieron que era precisa la forma de subir el archivo.

**Figura 84.**

*¿Es clara la forma de subir el archivo del código desarrollado en el IDE?*



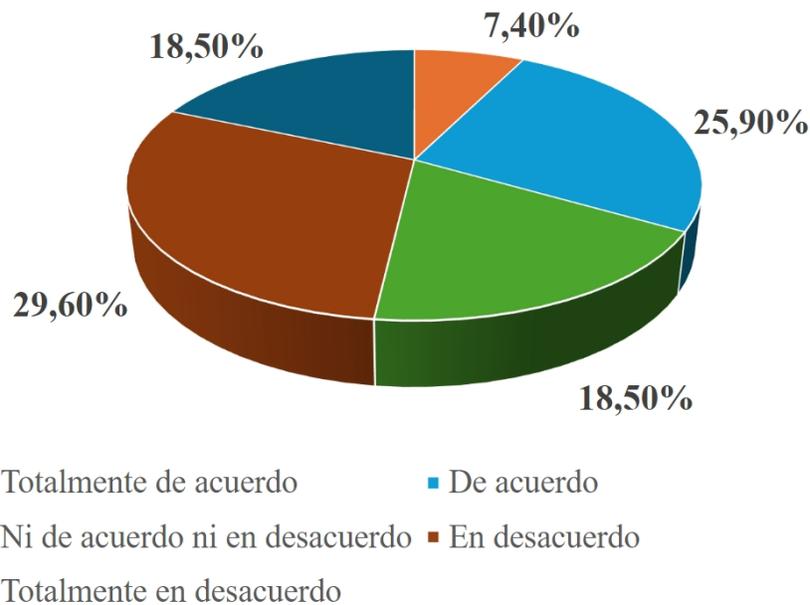
Fuente: Elaboración propia.

Los resultados que se presentan en la Figura 84, demuestran que más de la mitad de los estudiantes realizaron la actividad de subir el archivo, sin embargo, implicó dificultad ya que no les fue clara la forma de hacerlo.

También se les indicó a los estudiantes que respondieran la pregunta ¿Es clara la forma de editar el código de la solución al ejercicio propuesto?, a lo que el 33,3% respondieron de forma afirmativa indicando que estaban de acuerdo y totalmente de acuerdo con la forma de editar el código, sin embargo, el 18,5% no se sintió de acuerdo ni en desacuerdo, pero el 48,1% percibieron de forma inadecuada la forma de editar el código.

**Figura 85.**

*¿Es clara la forma de editar el código de la solución al ejercicio propuesto?*



Fuente: Elaboración propia.

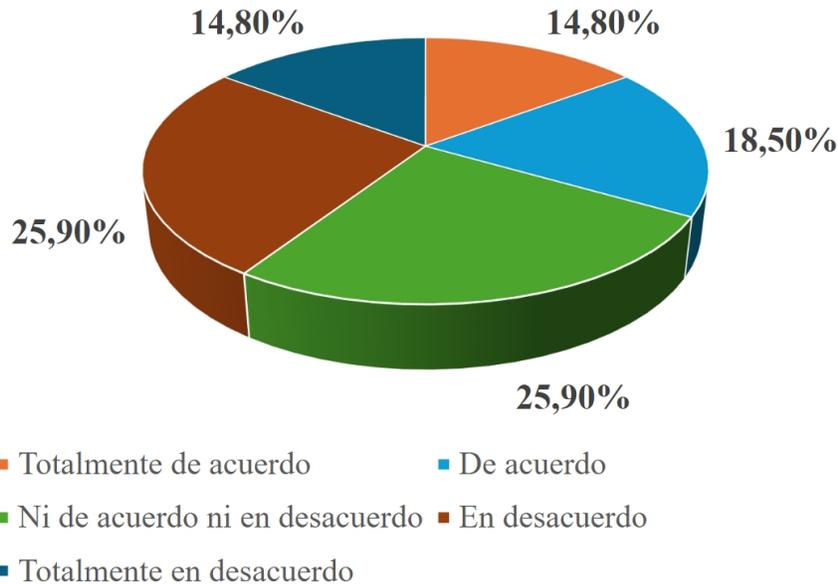
La Figura 85 refleja nuevamente los inconvenientes que presentaron los estudiantes, en esta oportunidad en el proceso de editar el código realizado o subido en el laboratorio, a pesar de haber bajado el porcentaje de estudiantes que indicaron que la forma de editar el código no es clara, sigue siendo un porcentaje alto.

Igualmente, se les preguntó a los estudiantes si es clara la forma en cómo deben enviar los ejercicios, a lo cual el 14,8% indica haber estado totalmente en desacuerdo, el 25,9% en desacuerdo, un 14,8% de los estudiantes indicó que no estuvo de acuerdo, pero tampoco en desacuerdo, un 18,5% reveló estar de acuerdo y un 14,8% sintió estar completamente de acuerdo. De esta manera, en la Figura 86, se evidencia que el porcentaje de estudiantes que les fue clara la forma de enviar los ejercicios continúa siendo del 33,3%, así mismo, más del 50% de los estudiantes sintieron que no fue clara la manera de remitirlos,

resultados similares a los registrados en los gráficos anteriores.

**Figura 86.**

*¿Es clara la forma en cómo se debe enviar los ejercicios?*

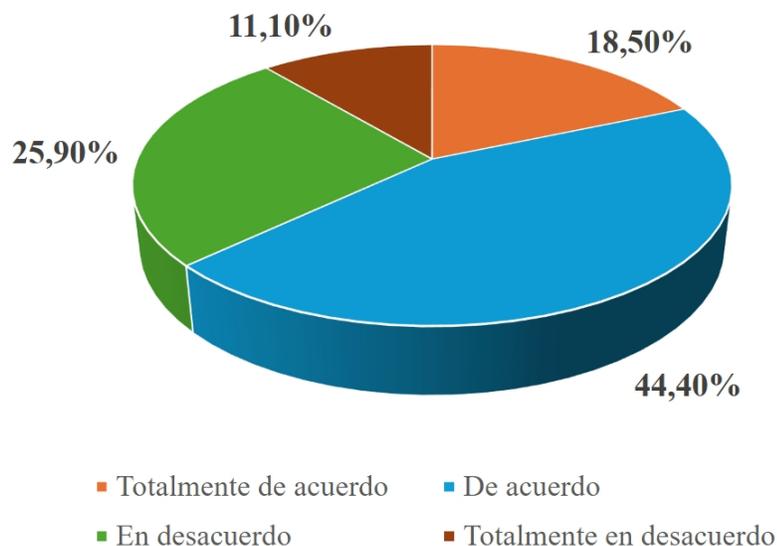


Fuente: Elaboración propia.

Por otro lado, la presentación de los ejercicios de programación viene acompañado de un esquema en el cual se relacionan diversos casos, en los cuales se toman ejemplos de datos de entrada, y con relación a estos se presentan datos de salida específicos, lo que les permite a los estudiantes tener valores de prueba frente al proceso que deben realizar. Conforme al bosquejo mencionado, se les preguntó a los estudiantes si este esquema era comprensible, ante esto, el 37% de los estudiantes indicaron que están totalmente en desacuerdo y en desacuerdo, respectivamente, sin embargo, el 25% estuvo totalmente de acuerdo y el 44,4% estuvo de acuerdo. Los resultados presentados en la Figura 87, reflejan que más del 70% de los estudiantes comprenden el esquema de entrada – salida, modelo importante para el proceso de identificación y resolución de problemas algorítmicos.

### Figura 87.

¿Comprende el esquema entrada – salida utilizado para la presentación de los enunciados?



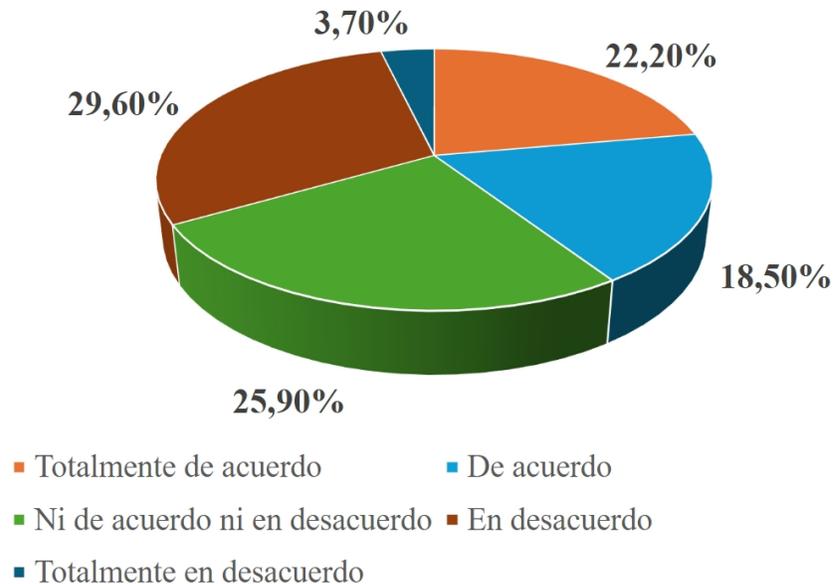
Fuente: Elaboración propia.

Finalmente, la última pregunta frente a la usabilidad de la plataforma fue, ¿Comprende el resultado que arroja la plataforma en el momento de enviar los ejercicios a evaluar?, de este modo, 3,7% estuvo totalmente en desacuerdo y el 29,6% en desacuerdo, así mismo, el 25,9% indicó estar ni de acuerdo ni en desacuerdo, el 18,5% estuvo de acuerdo y el 22,2% se encontraba totalmente de acuerdo.

La plataforma arroja una retroalimentación (resultados) frente a los ejercicios enviados por los estudiantes, en los cuales, si esta todo correcto le indica la nota que obtuvo, sin embargo, si se presentan errores, se reflejan los casos en los cuales, la solución planteada por el estudiante no los satisface o cumple, de esta manera, el estudiante debe realizar un proceso de revisión de la solución trazada y tomar correctivos. En este sentido, la Figura 88 muestra el 25,9% de los estudiantes no comprendieron los resultados arrojados por la plataforma, en cambio un 40,7% indica comprenderlos, sin embargo, un poco más del 25,9% de los estudiantes no demuestra tener una postura clara frente a la comprensión de los resultados arrojados por la plataforma.

**Figura 88.**

*¿Comprende el resultado que arroja la plataforma en el momento de enviar los ejercicios a evaluar?*



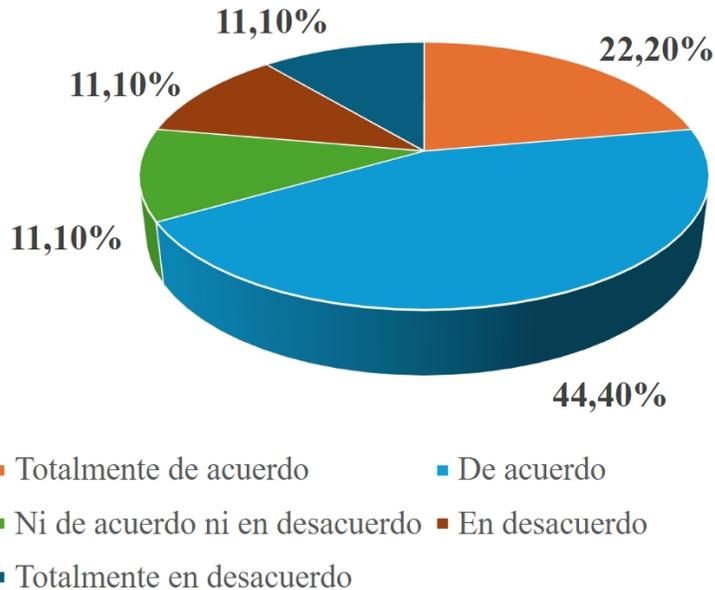
Fuente: Elaboración propia.

#### **4.4.2 Apoyo al proceso de aprendizaje**

En el segundo apartado del instrumento, desde la perspectiva de los estudiantes se indagó sobre cómo la implementación del Laboratorio Virtual de Programación apoya el proceso de aprendizaje, por lo cual se realizaron tres preguntas relacionadas con el aprendizaje, el autoaprendizaje y el desarrollo de habilidades de programación. En este sentido la primera pregunta fue: *¿Considera que la plataforma le facilita el aprendizaje de un lenguaje de programación?*, los resultados evidenciaron que el 11,1% de los estudiantes están totalmente en desacuerdo, seguido de otro 11,1% de los cuales estaban en desacuerdo, así mismo, otro 11,1% indicaron que ni estaban de acuerdo ni en desacuerdo, sin embargo, un 22,2% estuvo totalmente de acuerdo y finalmente un 44,4% reflejaron haber estado de acuerdo, tal como se puede apreciar en la Figura 89.

**Figura 89.**

*¿Considera que la plataforma le facilita el aprendizaje de un lenguaje de programación?*



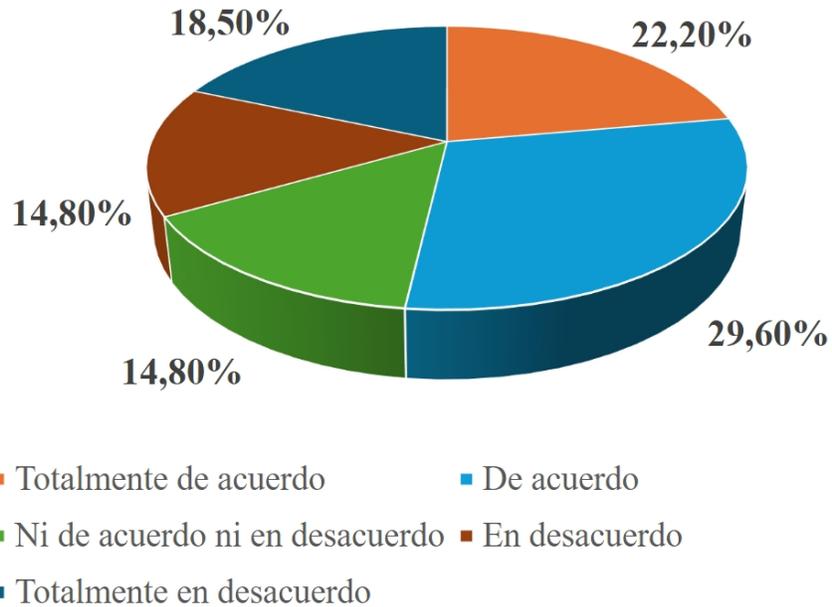
Fuente: Elaboración propia.

Los resultados arrojados en la Figura 89 son positivos, ya que es apreciable que para más del 66% de los estudiantes consideran que el Laboratorio Virtual de Programación es una herramienta que facilita el proceso de aprendizaje por medio de la cual, es posible aprender un lenguaje de programación, sin embargo, es importante conocer las apreciaciones más a fondo del 22,2% que no lo consideran útil, además del 11,1% de los estudiantes que no muestra una postura clara.

Del mismo modo, ante la pregunta ¿Considera que la plataforma facilitaría el proceso de autoaprendizaje en la línea de programación de computadores? Los estudiantes respondieron, el 22,2% están totalmente en desacuerdo, el 14,8% indican estar en desacuerdo y el 14,8% reflejan ni estar de acuerdo ni en desacuerdo, sin embargo, el 29,6% consideran estar de acuerdo y finalmente un 22,2% se encuentran totalmente de acuerdo, estos resultados pueden apreciarse en el figura 90.

**Figura 90.**

*¿Considera que la plataforma facilitaría el proceso de autoaprendizaje en la línea de programación de computadores?*



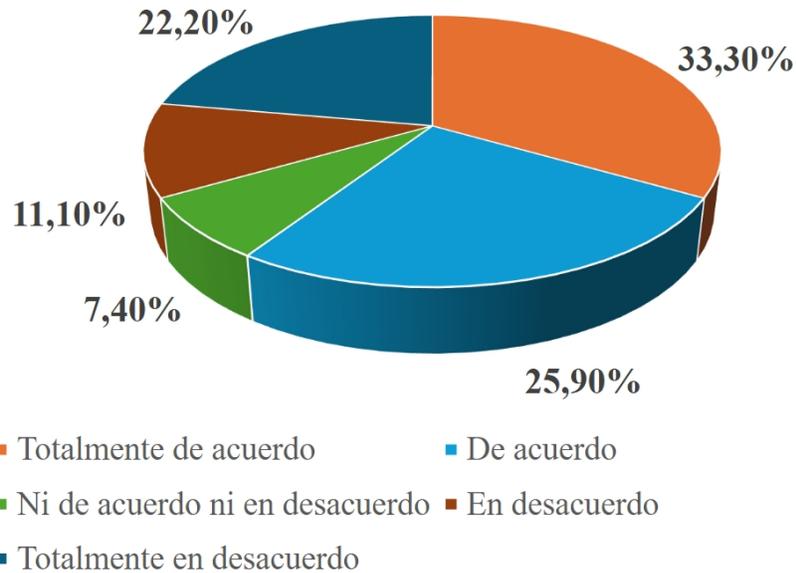
Fuente: Elaboración propia.

En la Figura 90, es posible apreciar que para más del 51% de los estudiantes, el Laboratorio Virtual de Programación es un medio por medio del cual es posible fortalecer su proceso de autoaprendizaje en la línea de programación, las cuales incluyen asignaturas como la programación orientada a objetos y las estructuras de datos; sin embargo, en comparación con los resultados del figura 89, el porcentaje de estudiantes que no consideran la plataforma como un apoyo para el fomento del autoaprendizaje de la programación aumentó a un 33,3%.

Así mismo, se les preguntó a los estudiantes si les gustaría seguir utilizando el Laboratorio Virtual de Programación para mejorar las habilidades en la construcción de algoritmos, y el 22,2% indicaron estar totalmente en desacuerdo, un 11,1% en desacuerdo, seguido de un 7,4% los cuales no se encuentran ni de acuerdo ni en desacuerdo, por otro lado, un 25,9% se inclinaron por estar de acuerdo y finalmente, un 33,3% están totalmente de acuerdo en utilizar la plataforma para mejorar las habilidades de programación. Estos datos pueden apreciarse en el figura 91.

**Figura 91.**

*¿Le gustaría seguir utilizando la plataforma para mejorar las habilidades en la construcción de algoritmos?*



Fuente: Elaboración propia.

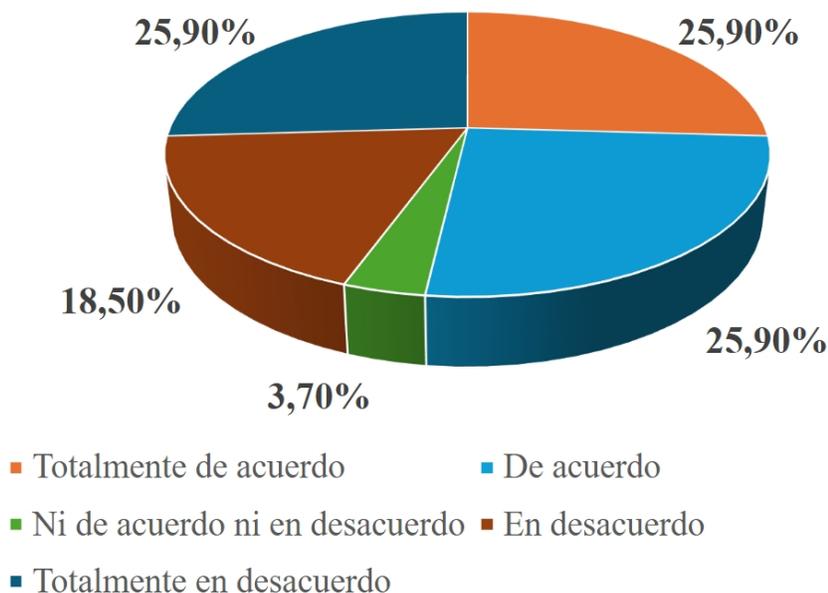
Los datos arrojados en la Figura 91, reflejan la intención de la mayoría de los estudiantes por continuar utilizando el Laboratorio Virtual de Programación, ese 59,2% posiblemente han evidenciado el potencial de la plataforma en el desarrollo de sus habilidades de programación. Por otro lado, el porcentaje de estudiantes restantes pueden tener diferentes razones por las cuales, no utilizarían la plataforma y es posible relacionar estas posturas dado los resultados en el apartado de usabilidad.

#### **4.4.2 Aspectos Adicionales**

De la misma manera, se realizaron preguntas con el objetivo de profundizar en la percepción que tienen los estudiantes frente a la plataforma, de este modo, la primera pregunta planteada a los estudiantes fue, ¿Comprende la intención que tiene la plataforma?, a lo que el 25,9% indicó estar totalmente en desacuerdo, el 18,5% estaba en desacuerdo, el 3,7% no estuvo ni de acuerdo ni en desacuerdo, el 25,9% se inclinó por estar de acuerdo y finalmente otro 25,9% estuvo totalmente de acuerdo.

**Figura 92.**

*¿Comprende la intención que tiene la plataforma?*



Fuente: Elaboración propia.

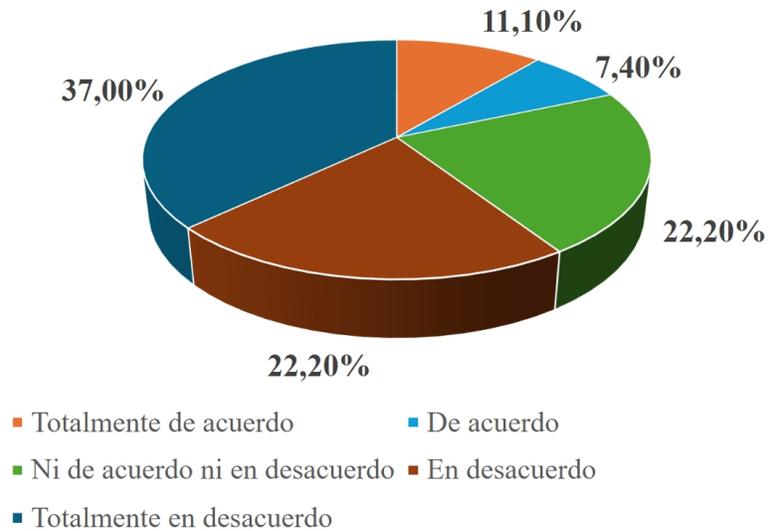
Con los resultados que se reflejan en la figura 92, es importante anotar que más del 50% de los estudiantes comprenden la intención del Laboratorio Virtual de Programación, sin embargo, con un porcentaje muy cercano a la mitad, el resto de las estudiantes no logran alcanzar a percibir el propósito de la plataforma.

Adicionalmente, se les realizó a los estudiantes, la siguiente pregunta ¿Piensa que podrá utiliza esta aplicación sin requerir capacitación adicional?, a lo que el 37% indicó estar totalmente en desacuerdo, el 22,2% se encontró en desacuerdo, un 22,2% no está ni de acuerdo ni en desacuerdo, así mismo, un 7,4% mostró una inclinación por estar de acuerdo y finalmente un 11,1% se encontraba totalmente de acuerdo.

Los resultados que se muestran en la figura 93, reflejan una alta inclinación del estudiantado por requerir de una constante capacitación de la plataforma, ya que más del 59% de la población indicó estar en desacuerdo en utilizar nuevamente el laboratorio sin requerir acompañamiento, lo que genera información valiosa para mejorar los procesos de formación.

**Figura 93.**

*¿Piensa que podrá utilizar esta plataforma sin requerir capacitación adicional?*

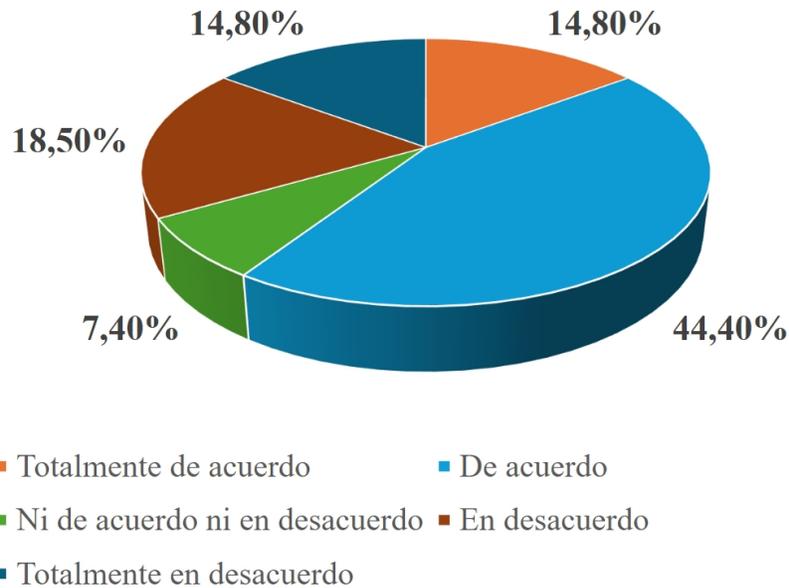


Fuente: Elaboración propia.

Finalmente, se les preguntó a los estudiantes si consideran adecuado el uso de la plataforma VPL en las asignaturas de la línea de programación y el 14,8% reflejó estas totalmente en desacuerdo, el 18,5% estaba en desacuerdo, un 7,4% se encontraba ni de acuerdo ni en desacuerdo, del mismo modo, un 44,4% indicó estar de acuerdo y finalmente un 14,8% tuvo una inclinación por estar totalmente de acuerdo.

**Figura 94.**

*¿Considera adecuado el uso de la plataforma VPL en las asignaturas de la línea de programación?*



Fuente: Elaboración propia.

Por último, la Figura 94 indica que más del 59% de los estudiantes consideran adecuado el uso del Laboratorio Virtual en las diferentes asignaturas de programación reflejando de esta manera, que, para más de la mitad de los estudiantes, la plataforma se constituye como una herramienta que contribuye al fortalecimiento de las habilidades de programación.

## 4.5 CONCLUSIONES

Los resultados arrojados de la percepción de los estudiantes respecto a la interacción con el laboratorio virtual de programación proyectan resultados interesantes frente a los trabajos futuros con la herramienta, en primer lugar, el apartado inicial reflejó el nivel de complejidad que percibieron los estudiantes frente a la interacción con la plataforma fue en términos generales bajo, muy similar a los resultados presentados por la investigación de Ramos et al. (2022) quienes evaluaron el módulo VPL de Moodle en 5 aspectos (satisfacción, eficiencia, capacidad de aprendizaje, utilidad y control). Se concluye que VPL no es fácil de usar, además de no ser intuitiva y no ofrecer un módulo de ayuda, sin embargo, no deja de ser una herramienta útil e interesante de trabajar, ya que permite a los estudiantes interactuar en una misma plataforma educativa y que todos los registros de rendimiento académico de los estudiantes se encuentran en esta misma, favoreciendo el proceso de evaluación, administración y control por parte de los docentes.

Por otro lado, a pesar que la evaluación de interacción no es la más positiva, si es interesante y positivo analizar los resultados frente a la percepción de los estudiantes en el proceso de enseñanza de la programación, ya que en este caso, en la mayoría de las preguntas realizadas, un porcentaje mayor y significativo, comprende que el laboratorio busca mejorar las habilidades de programación, además de percibirlo como un proceso, de apoyo en su proceso educativo, así mismo, se identifica como una excelente oportunidad para ser integrado a los diferentes cursos de programación de la propuesta curricular de la carrera de Ingeniería de Sistemas de la Universidad de la Amazonia.

## 4.6 REFERENCIAS BIBLIOGRÁFICAS

Alammary, A. (2019). Blended learning models for introductory programming courses: A systematic review. *PloS one*, 14(9), e0221765. <https://doi.org/10.1371/journal.pone.0221765>.

Byrne, P., Lyons, J. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin*, 33:49-52. <https://doi.org/10.1145/507758.377467>

Cardoso, M., Marques, R., de Castro, A. V., & Rocha, Á. (2021). Using Virtual Programming Lab to improve learning programming: The case of Algorithms and Programming. *Expert Systems*, 38(4), e12531.

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2), ep272.

Claros-Moreno, F. C., Sánchez-Lopera, S. A., & Millán-Rojas, E. E. (2019). Variables generadoras de la repitencia y procrastinación en estudiantes de Lógica y Algoritmos I y II en la Universidad de la Amazonia. *Encuentros*, 17(02), 108-117. <https://doi.org/10.15665/encuent.v17i02.1994>

Gomes, A., & Mendes, A. (2014, October). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. In 2014 IEEE Frontiers in Education Conference (FIE) Proceedings.1-8. <https://doi.org/10.1109/FIE.2014.7044086>

Ramos, V. F., Cechinel, C., Magé, L., & Lemos, R. (2022). Student and Lecturer Perceptions of Usability of the Virtual Programming Lab Module for Moodle. *Informatics in Education*, 20(2), 297-315.

Savage, S., & Piwek, P. (2019). Full report on challenges with learning to program and problem solve: an analysis of first year undergraduate Open University distance learning students' online discussions. [http://oro.open.ac.uk/68073/1/IOC\\_TM112\\_Python\\_Help\\_Forum\\_Research.pdf](http://oro.open.ac.uk/68073/1/IOC_TM112_Python_Help_Forum_Research.pdf)

