

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“ESTUDIO SOBRE LOS FACTORES QUE AFECTAN EL APRENDIZAJE EN UN CURSO INTRODUCTORIO DE PROGRAMACIÓN”

ERNESTO SEBASTIÁN ARTURO BARRÍA ANDRADE

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Federico Meza Montoya
Profesor Correferente: Andrea Vásquez Guerra

Julio - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: ESTUDIO SOBRE LOS FACTORES QUE AFECTAN EL
APRENDIZAJE EN UN CURSO INTRODUCTORIO DE
PROGRAMACIÓN

Nombre del candidato(a): Ernesto Sebastián Arturo Barría Andrade

Carrera / Grado: Ingeniería Civil Informática

Campus: Santiago San Joaquín; **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Federico Meza Montoya, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 20/08/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 20/08/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

A mis padres, por su amor incondicional, su apoyo constante y por ser el pilar fundamental en cada etapa de mi vida. Su sacrificio y aliento me inspiraron a alcanzar esta meta.

A mis amigos del colegio, por los recuerdos imborrables y las constantes muestras de amistad, que a pesar de todo ha sido una constante a lo largo del tiempo.

A mis amigos de la universidad, por las largas noches de borr- estudio, apoyo, compañía y por hacer de esta etapa una experiencia inolvidable.

A mi pareja, por ser mi constante fuente de alegría y calma durante la realización de esta tesis.

A mis gatos, Kira y Silver, por ser mi compañía durante largas horas de estudio.

Y a quienes con su conocimiento y guía iluminaron mi trayectoria académica, compartiendo no solo sabiduría, sino también la pasión por el aprendizaje y la innovación.

Esta memoria es el reflejo de un esfuerzo conjunto y un tributo a su invaluable contribución.

AGRADECIMIENTOS

La culminación de esta memoria no habría sido posible sin la invaluable colaboración y el apoyo constante de diversas personas. A todos ellos, mi más sincero agradecimiento.

En primer lugar, expreso mi profunda gratitud a mis queridos padres, pareja y familia, quienes con su amor incondicional, su cariño constante y su apoyo inquebrantable me dieron la fuerza y la motivación para seguir adelante en este desafiante camino. Su fe en mí fue mi mayor inspiración.

Asimismo, mi más sincero agradecimiento a mi profesor guía, Federico Meza, cuyo acompañamiento fue fundamental a lo largo de todo este proceso. Su sabiduría, paciencia, constante motivación y, en particular, su invaluable ayuda para la obtención de los datos, fueron pilares esenciales que permitieron el desarrollo y la exitosa conclusión de esta investigación. Su guía experta y su compromiso excedieron toda expectativa.

Finalmente, agradezco a cada persona que, con su paciencia, comprensión y palabras de aliento, hizo más llevadero el camino y me impulsó a seguir adelante.

A todos ustedes, mi eterno reconocimiento.

RESUMEN

Resumen— Este estudio explora los factores que inciden en el rendimiento académico de estudiantes de primer año en un curso introductorio de programación en la Universidad Técnica Federico Santa María durante el año 2025. Empleando una **metodología cuantitativa, de corte transversal y diseño no experimental**, se analizan variables académicas previas (Puntaje NEM, Puntaje Ranking, Puntaje Matemática M1, Puntaje Matemática M2, Puntaje Lenguaje, Puntaje Ciencias, Puntaje Historia), habilidades cognitivas (Test de Lawson, Test de Diagnóstico de Matemática, Test de Pensamiento Computacional y sus dimensiones), factores psicológicos (motivación y autoeficacia), y datos cualitativos (carrera y campus).

Los datos se recopilieron de diversas fuentes institucionales y encuestas, y fueron sometidos a un riguroso proceso de limpieza y transformación en R. El análisis incluyó estadísticos descriptivos, correlaciones de Pearson y Spearman, y el desarrollo de **modelos de regresión lineal múltiple** para predecir el rendimiento en el primer certamen de programación y la nota final del curso.

Los resultados muestran que el **rendimiento en el Certamen 1** es el predictor más robusto de la nota final del curso ($r = 0,783$). Las **habilidades lógico-matemáticas** (Puntaje Matemática M1, M2) y las **dimensiones del pensamiento computacional** (especialmente Pensamiento Algorítmico y Abstracción) también demuestran una correlación positiva y significativa con el desempeño en programación. En contraste, las variables psicológicas como la motivación y autoeficacia, así como el Test de Lawson, mostraron correlaciones muy bajas, lo que podría deberse a sesgos en la recolección de datos y al enfoque de las pruebas.

Se evalúa un modelo predictivo para el Certamen 1 que explica el **39.71 % de la varianza**, y un modelo para la nota final del curso que explica el **67.6 % de la varianza**. Se proponen criterios de clasificación de riesgo para estudiantes, destacando la importancia de la reclasificación posterior al Certamen 1 para mejorar la precisión. Este trabajo subraya el potencial del análisis de datos para informar decisiones pedagógicas y diseñar intervenciones tempranas más efectivas, sentando las bases para futuras investigaciones y una cultura educativa basada en evidencias.

Palabras Clave— Aprendizaje; Programación; Factores; Estudio; Correlación

ABSTRACT

Abstract— This study explores the factors influencing the academic performance of first-year students in an introductory programming course at Universidad Técnica Federico Santa María during 2025. Employing a **quantitative, cross-sectional, and non-experimental methodology**, the study analyzes prior academic variables (NEM Score, Ranking Score, Mathematics M1 Score, Mathematics M2 Score, Language Score, Science Score, History Score), cognitive skills (Lawson Test, Mathematics Diagnostic Test, Computational Thinking Test and its dimensions), psychological factors (motivation and self-efficacy), and qualitative data (major and campus).

The data were collected from various institutional sources and surveys, and subjected to a rigorous cleaning and transformation process in R. The analysis included descriptive statistics, Pearson and Spearman correlations, and the development of **multiple linear regression models** to predict performance in the first programming exam and the final course grade.

Results show that **performance in Exam 1** is the most robust predictor of the final course grade ($r = 0,783$). **Logical-mathematical skills** (Mathematics M1, M2 Scores) and **computational thinking dimensions** (especially Algorithmic Thinking and Abstraction) also demonstrate a positive and significant correlation with programming performance. In contrast, psychological variables such as motivation and self-efficacy, as well as the Lawson Test, showed very low correlations, which could be due to biases in data collection and the focus of the tests.

A predictive model for Exam 1 is evaluated, explaining **39.71 % of the variance**, and a model for the final course grade, explaining **67.6 % of the variance**. Student risk classification criteria are proposed, highlighting the importance of reclassification after Exam 1 to improve accuracy. This work underscores the potential of data analysis to inform pedagogical decisions and design more effective early interventions, laying the groundwork for future research and an evidence-based educational culture.

Keywords— Learning; Programming; Factors; Study; Correlation

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 Universidad y Asignatura	4
1.2 Actores involucrados	5
1.3 Objetivos del estudio	5
1.4 Árbol del problema	5
1.4.1 Problema central	6
1.4.2 Causas del problema	6
1.4.3 Efectos del problema	6
CAPÍTULO 2: MARCO CONCEPTUAL	9
2.1 Autoeficacia	9
2.2 Antecedentes matemáticos	9
2.3 Motivación	10
2.4 Pensamiento Computacional	11
2.4.1 Abstracción	11
2.4.2 Descomposición	12
2.4.3 Reconocimiento de Patrones y Generalización	12
2.4.4 Pensamiento Algorítmico	13
2.5 Errores conceptuales	14
2.6 Comprensión Lectora	14
2.7 Prueba de Acceso a la Educación Superior (PAES)	15
2.7.1 Estructura de la PAES	15
2.7.2 Relevancia de la PAES	16
2.8 El Test de Lawson en el Contexto de la Programación	16
2.9 Trabajos previos	16
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	18
3.1 Metodología	18
3.1.1 Participantes y Contexto del Estudio	18
3.2 Uso de datos	18
3.3 Recopilación de Datos	18
3.3.1 Datos de Ingreso Académico Previo	19
3.3.2 Test de Lawson	27

3.3.3	Test de Diagnóstico de Matemática	29
3.3.4	Test de Pensamiento Computacional	29
3.3.5	Encuestas de Factores Psicológicos	32
3.3.6	Datos cualitativos	39
3.3.7	Datos de Rendimiento en Programación	41
3.4	Transformación de Datos	44
3.5	Análisis de Datos	46
3.5.1	Estadísticos Descriptivos	46
3.5.2	Análisis de Correlaciones	46
3.5.3	Modelado de Regresión Lineal Múltiple	47
3.5.4	Análisis de Interacción Motivación x Autoeficacia	48
3.5.5	Análisis por Categoría	48
3.6	Validación del Modelo de Predicción	48
3.6.1	Realización de Predicciones y Análisis de Errores	49
3.7	Análisis de Correlación entre C1 y Nota Final	49
CAPÍTULO 4: Análisis y Resultados		50
4.1	Factores Asociados al Rendimiento en el Certamen 1	50
4.2	Comparación de Modelos Predictivos	51
4.3	Análisis del Modelo con Interacción: Motivación x Autoeficacia	52
4.3.1	Evaluación Global del Modelo	52
4.3.2	Interpretación de Coeficientes	53
4.4	Análisis por Carrera y Campus	53
4.5	Análisis del Test de Lawson	55
4.6	Evaluación del Modelo de Predicción para el Certamen 2	57
4.7	Correlación con la Nota Final del Curso	61
4.7.1	Modelo de Regresión Lineal Múltiple para la Nota Final del Curso	63
4.8	Propuestas de Clasificación	64
4.8.1	Criterio 1: Nota de aprobación como punto de corte central	64
4.8.2	Criterio 2: Clasificación basada en tramos equitativos de riesgo	65
5	CONCLUSIONES	66
5.1	Alcances y limitaciones	66
5.2	Perspectiva futura	67
5.3	Reflexión Final	67
REFERENCIAS BIBLIOGRÁFICAS		68

ÍNDICE DE FIGURAS

1	Total de inscritos versus total de aprobados por carrera	3
2	Porcentaje de aprobación por carrera	4
3	Árbol del problema	8
4	Distribución de puntajes NEM	20
5	Boxplot de puntajes NEM	20
6	Histograma de puntajes ranking	21
7	Boxplot de puntajes ranking	21
8	Distribución de puntajes Matemática M1	23
9	Boxplot de puntajes Matemática M1	23
10	Distribución de puntajes Matemática M2	24
11	Boxplot de puntajes Matemática M2	24
12	Distribución de puntajes Lenguaje	25
13	Boxplot de puntajes Lenguaje	25
14	Distribución de puntajes Ciencias	26
15	Boxplot de puntajes Ciencias	26
16	Distribución de puntajes Historia	28
17	Boxplot de puntajes Historia	28
18	Distribución de puntajes Lawson	30
19	Boxplot de puntajes Lawson	30
20	Distribución de puntajes Matemática	31
21	Boxplot de puntajes Matemática	31
22	Distribución de puntajes Abstracción	33
23	Boxplot de puntajes Abstracción	33
24	Distribución de puntajes Pensamiento Algorítmico	34
25	Boxplot de puntajes Pensamiento Algorítmico	34
26	Distribución de puntajes Reconocimiento de Patrones y Generalización	35
27	Boxplot de puntajes Reconocimiento de Patrones y Generalización	35
28	Distribución de puntajes Descomposición	36
29	Boxplot de puntajes Descomposición	36
30	Distribución de puntajes Pensamiento Computacional	37
31	Boxplot de puntajes Pensamiento Computacional	37
32	Cantidad de alumnos por carrera	40
33	Cantidad de alumnos por campus	40
34	Distribución de notas Certamen 1	42
35	Boxplot de notas Certamen 1	42
36	Distribución de notas Certamen 2	43
37	Boxplot de notas Certamen 2	43
38	Distribución de notas finales	45
39	Boxplot de notas Finales	45
40	Variables más correlacionadas con certamen 1	50
41	Distribución de notas de certamen 1 por campus	54
42	Distribución de notas de certamen 1 por carrera	55

43	Distribución de Notas de Certamen 1 por nivel de test de Lawson	56
44	Distribución de puntaje de Abstracción por nivel de test de Lawson	57
45	Gráfico de dispersión nota predicción vs nota real certamen 2	60
46	Distribución de error de predicción certamen 2	60
47	Gráfico de barras comparativo entre correlaciones con certamen 1 vs nota final	62

ÍNDICE DE TABLAS

1	Estadísticas descriptivas de los puntajes NEM	19
2	Estadísticas descriptivas de los puntajes ranking	19
3	Estadísticas descriptivas de los puntajes de la prueba de matemáticas M1	22
4	Estadísticas descriptivas de los puntajes de la prueba de matemáticas M2	22
5	Estadísticas descriptivas de 'Puntaje Lenguaje'	27
6	Estadísticas descriptivas de los puntajes de la prueba de Ciencias	27
7	Estadísticas descriptivas de los puntajes de la prueba de Historia	27
8	Estadísticas descriptivas de los puntajes del test de Lawson	29
9	Estadísticas descriptivas de los puntajes de la prueba de diagnóstico de matemáticas	29
10	Estadísticas descriptivas consolidadas de Pensamiento Computacional y sus dimensiones	32
11	Ítems de la Encuesta de Motivación	38
12	Estadísticas descriptivas del promedio de motivación	38
13	Ítems de la Encuesta de Autoeficacia	39
14	Estadísticas descriptivas del promedio de autoeficacia	39
15	Estadísticas descriptivas de las notas del primer certamen.	41
16	Estadísticas descriptivas de las notas del Certamen 2	41
17	Estadísticas descriptivas de las Notas Finales del curso	44
18	Interpretación de la magnitud del coeficiente de correlación.	47
19	Estadísticos descriptivos de motivación y autoeficacia promedio.	51
20	Comparación de modelos predictivos del rendimiento en el Certamen 1.	52
21	Coeficientes del Modelo de Regresión Lineal: Interacción entre Motivación y Autoeficacia	53
22	Métricas de Ajuste del Modelo de Regresión	54
23	Resumen estadístico del Certamen 1 según grupos de Puntajes Lawson	56
24	Resumen estadístico de la dimensión <i>Abstracción</i> según grupos de LAWSON PRE	57
25	Estadísticas Descriptivas de las Predicciones del Certamen 2	58
26	Estadísticas Descriptivas de las Notas Reales del Certamen 2	58
27	Estadísticas del Error de Predicción para el Certamen 2	59
28	Estadísticas de la Diferencia Absoluta ($ Predicción - Real $) para el Certamen 2	59
29	Correlación de Variables con la Nota Final del Curso	61
30	Coeficientes del modelo de regresión lineal para la nota final del curso	63

INTRODUCCIÓN

El **aprendizaje de la programación** se ha consolidado como una habilidad fundamental en el siglo XXI, trascendiendo las fronteras de las disciplinas puramente informáticas para convertirse en un pilar del pensamiento lógico y la resolución de problemas en diversas áreas profesionales. Sin embargo, su enseñanza a nivel universitario, especialmente en cursos introductorios, enfrenta el desafío de una **gran heterogeneidad en los conocimientos y habilidades previas de los estudiantes**, lo que a menudo se traduce en rendimientos dispares y altas tasas de reprobación. Es en este contexto que surge la necesidad de **comprender mejor los factores que influyen en el éxito inicial** en programación y desarrollar herramientas que permitan identificar y apoyar a los alumnos en riesgo de manera proactiva.

La presente memoria aborda la problemática del **bajo rendimiento en el curso de Introducción a la Programación** de la Universidad Técnica Federico Santa María. Este curso, que congrega a un promedio de 1500 estudiantes por semestre de diversas carreras, evidencia una marcada disparidad en la aprobación. El problema central se manifiesta en un alto porcentaje de reprobación, desmotivación estudiantil, dificultades en la comprensión conceptual y, en última instancia, una adquisición incompleta de habilidades computacionales esenciales. Las causas identificadas son multifactoriales, abarcando desde la **diversidad de antecedentes académicos y la falta de habilidades de pensamiento abstracto** hasta una baja autoeficacia, antecedentes matemáticos insuficientes y una falta de personalización en la enseñanza. Ante este panorama, el objetivo de este estudio es identificar y analizar estos factores, clasificar a los estudiantes según su nivel de riesgo y proponer estrategias de intervención temprana, justificando así la necesidad de una solución basada en datos.

Para abordar esta compleja situación, se propone un enfoque basado en el **análisis estadístico y el modelado predictivo**. La memoria emplea una **metodología cuantitativa, de corte transversal y diseño no experimental**. El proceso inició con la **recopilación exhaustiva de datos** de diversas fuentes institucionales y encuestas específicas. Esto incluyó el uso de puntajes de ingreso académico (PAES como NEM, Ranking, Matemática M1 y M2, Lenguaje, Ciencias e Historia), resultados de pruebas cognitivas como el Test de Lawson (razonamiento científico) y el Test de Pensamiento Computacional (abstracción, algorítmico, descomposición, reconocimiento de patrones), así como datos de encuestas sobre factores psicológicos como la motivación y la autoeficacia.

Una vez recolectados, los datos fueron sometidos a un **riguroso proceso de limpieza y transformación** utilizando el software R y RStudio, asegurando la anonimización de la información sensible y la integración de todas las variables de interés en un único *dataframe* base. El **análisis de datos** se llevó a cabo mediante estadísticos descriptivos y, fundamentalmente, a través del **análisis de correlaciones** (Pearson y Spearman) para identificar la fuerza y dirección de las relaciones entre las variables y el rendimiento en programación. El paso clave fue el **modelado de regresión lineal múltiple**, donde se desarrollaron y compararon tres modelos predictivos para el rendimiento en el Certamen 1, buscando el que mejor explicara la varian-

za y presentara el menor error. Adicionalmente, se exploró la interacción entre motivación y autoeficacia, y se realizó un análisis descriptivo por carrera y campus para contextualizar el rendimiento. La **validación del modelo de predicción** se realizó utilizando las notas del Certamen 2, y se complementó con un análisis de correlación con la Nota Final del curso para determinar la robustez de los predictores a lo largo del semestre.

El presente documento se organiza en los siguientes capítulos:

- El **Marco Conceptual** establece las bases teóricas y define los conceptos fundamentales para la comprensión del estudio.
- La **Definición del Problema** describe detalladamente la situación actual, los actores involucrados, los objetivos de la investigación y el árbol del problema que contextualiza las causas y efectos del bajo rendimiento.
- La **Propuesta de Solución** explica la metodología de investigación, la estrategia de uso y recopilación de datos, el proceso de transformación de datos y las técnicas de análisis empleadas para construir los modelos predictivos.
- El capítulo de **Análisis y Resultados** presenta los hallazgos clave, incluyendo los factores asociados al rendimiento, la comparación de modelos predictivos, el análisis de variables categóricas y la validación de las predicciones, así como la correlación con la nota final del curso.
- Finalmente, las **Conclusiones** sintetizan los aportes del trabajo, discuten sus alcances y limitaciones, y proponen futuras líneas de investigación y acción para la mejora continua del proceso de enseñanza-aprendizaje de la programación.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

La programación es una habilidad crucial en el mundo laboral contemporáneo, con aplicaciones transversales que no solo potencian el desarrollo del pensamiento lógico, sino que también permiten adaptarse a los desafíos de la era digital. Sin embargo, aprender a programar puede resultar difícil debido a una variedad de factores, tanto internos como externos al estudiante [Guzdial, 2010]. En el contexto del curso de Introducción a la Programación (IWI-131 e INF-129) de la Universidad Técnica Federico Santa María, los estudiantes provienen de diversos orígenes y carreras, generando rendimientos dispares en la asignatura (Ver figuras 1 y 2). Eso sumado a que la cantidad de alumnos promedio por semestre es de 1500, es necesario enfocarse en los que tienen un menor rendimiento o un mayor riesgo a reprobar el ramo.

Es fundamental identificar los factores que afectan tanto positiva como negativamente el aprendizaje de la programación en este contexto. Estos factores pueden incluir la motivación [Ryan y Deci, 2000], los antecedentes matemáticos [Ambrosio *et al.*, 2011], la autoeficacia [Bergin y Reilly, 2005] y las habilidades previas de pensamiento abstracto [Fonden Calzadilla, 2020], entre otros. Detectar estos elementos permitirá filtrar a los estudiantes que presentan un mayor riesgo de bajo rendimiento, facilitando la implementación de estrategias para mitigar las dificultades que puedan surgir a lo largo del curso. De esta manera, se pueden optimizar los recursos y métodos pedagógicos para asegurar que los estudiantes logren adquirir las competencias necesarias.

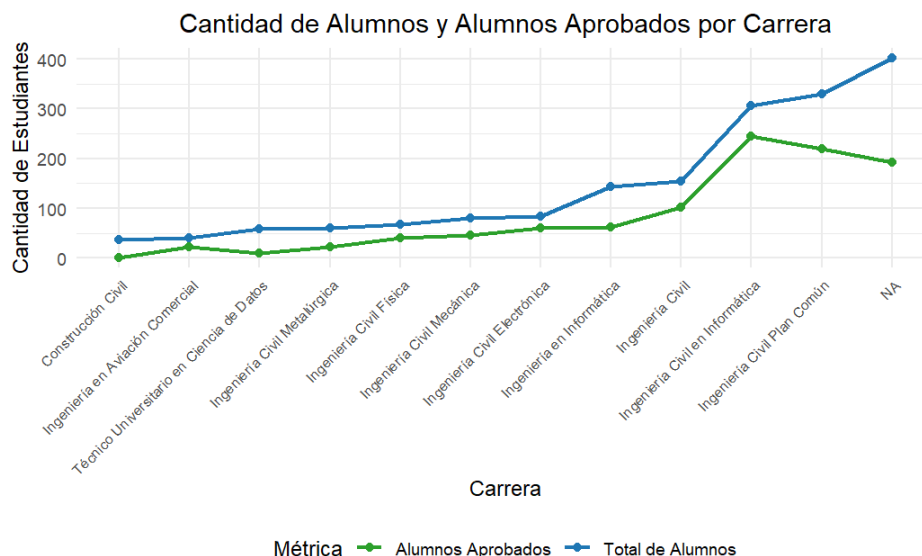


Figura 1: Total de inscritos versus total de aprobados por carrera
Fuente: Asignatura IWI-131 e INF-129 2025-1.

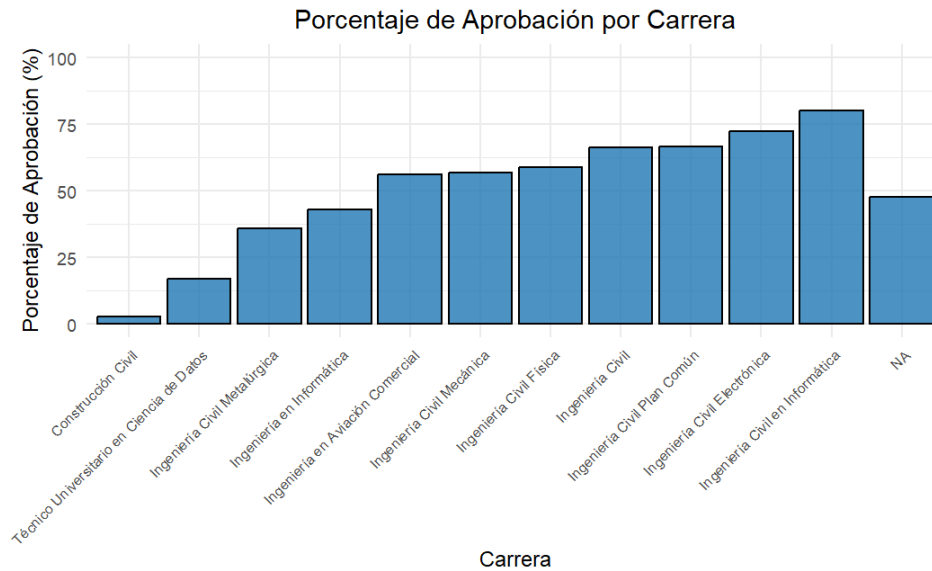


Figura 2: Porcentaje de aprobación por carrera
Fuente: Asignatura IWI-131 e INF-129 2025-1.

1.1. Universidad y Asignatura

La Universidad Técnica Federico Santa María (UTFSM) es una institución de educación superior con un fuerte sello científico-tecnológico, distribuida en cinco emplazamientos principales: Campus Casa Central Valparaíso, Sede Viña del Mar, Campus San Joaquín, Campus Vitacura y Sede Concepción. Su oferta académica de pregrado abarca ingenierías en diversas especialidades (tanto de base científica, conocidas como ingenierías civiles en Chile, como tecnológicas), licenciaturas de carácter científico, así como carreras en ingeniería comercial, arquitectura, ingeniería en aviación comercial, construcción civil, y una amplia gama de carreras técnicas universitarias de corta duración. Es importante destacar que la UTFSM cuenta con diversas vías de ingreso especial, incluyendo una vía directa para sus carreras técnicas universitarias que no exige la rendición de la Prueba de Acceso a la Educación Superior (PAES), lo que explica la presencia de estudiantes en la universidad que no poseen este puntaje.

La asignatura de Introducción a la Programación se ofrece bajo dos siglas distintas, IWI-131 e INF-129, si bien ambas comparten los mismos resultados de aprendizaje y utilizan los mismos instrumentos de evaluación. La evaluación del curso se compone de dos exámenes escritos, denominados certámenes, tareas de programación, controles (quizzes), y evaluaciones asincrónicas en línea. Los certámenes constituyen el principal instrumento de evaluación, no solo por su alta ponderación en la nota final (50 %), sino porque su aprobación, en promedio, es un requisito obligatorio para aprobar la asignatura. Existe un certamen recuperativo opcional que permite reemplazar la menor nota de los dos certámenes. La nota de aprobación en la UTFSM se consigue con una calificación de 55 en la escala de 0 a 100.

1.2. Actores involucrados

Este análisis se centra en los estudiantes del curso de Introducción a la Programación de la Universidad Técnica Federico Santa María. Los actores involucrados incluyen, además de los estudiantes, a los profesores y ayudantes del curso, quienes desempeñan un papel crucial en el proceso de enseñanza, y a los coordinadores académicos, que pueden influir en la estructura del programa educativo y en la adopción de medidas para apoyar el aprendizaje de los estudiantes.

1.3. Objetivos del estudio

El objetivo principal de este estudio es identificar y analizar los factores que influyen en el rendimiento de los estudiantes en el curso Introducción a la Programación. Estos factores pueden ser tanto individuales (como la motivación, la confianza en su capacidad para aprender a programar o sus habilidades cognitivas previas) como contextuales (como el apoyo recibido durante el curso o los métodos de enseñanza utilizados).

Los objetivos específicos del estudio son los siguientes:

1. Identificar factores internos y externos que afectan el desempeño académico de los estudiantes en el curso de programación y buscar fuentes de información que puedan medirlos.
2. Clasificar a los estudiantes de acuerdo con su nivel de riesgo en función de los factores identificados, y proponer estrategias de intervención temprana para aquellos que se encuentren en situación de vulnerabilidad.
3. Evaluar la capacidad predictiva de la clasificación de riesgo a través de métricas estadísticas, para determinar su efectividad en la identificación temprana de estudiantes con potencial de bajo rendimiento.

1.4. Árbol del problema

El árbol del problema 3 ilustra los factores que contribuyen al bajo rendimiento en el curso de Introducción a la Programación en la Universidad Técnica Federico Santa María. A través de la identificación de causas, efectos y relaciones, se puede comprender mejor los desafíos que enfrentan los estudiantes en este curso.

1.4.1. Problema central

El eje del árbol identifica el problema principal: "Bajo rendimiento en el curso de Introducción a la Programación". Este problema afecta directamente el avance académico de los estudiantes y su motivación hacia las materias relacionadas con la programación y la informática.

1.4.2. Causas del problema

En la parte inferior del árbol se encuentran las raíces que simbolizan las causas. Estas están divididas en diferentes categorías:

- Diversidad de antecedentes académicos de los estudiantes: Algunos estudiantes llegan con conocimientos previos desiguales, lo que genera una diferencia en sus capacidades iniciales.
- Falta de habilidades previas de pensamiento abstracto: La programación requiere de habilidades de pensamiento lógico y abstracto que no todos los estudiantes han desarrollado antes de tomar el curso.
- Baja autoeficacia: Muchos estudiantes no confían en su capacidad para aprender a programar, lo que afecta su rendimiento.
- Antecedentes matemáticos insuficientes: Un conocimiento débil en matemáticas puede dificultar la comprensión de los conceptos de programación.
- Errores conceptuales ("misconceptions"): Los estudiantes a menudo malinterpretan los conceptos clave, lo que lleva a un mal aprendizaje y a cometer errores frecuentes.
- Escasa motivación intrínseca: La falta de interés personal en la programación disminuye el esfuerzo que los estudiantes invierten en el curso.
- Falta de personalización en la enseñanza: La metodología de enseñanza no se ajusta a las necesidades individuales de los estudiantes, en especial a aquellas derivadas de sus falencias específicas. Los estudiantes que presentan dificultades previas, como la falta de habilidades matemáticas o de pensamiento abstracto, no reciben un apoyo focalizado en estas áreas clave. En lugar de abordar estas deficiencias, se aplica una enseñanza homogénea que no permite compensar las diferencias en preparación académica. Esto incrementa la dificultad de comprender los conceptos y avanzar en el curso.

1.4.3. Efectos del problema

En la parte superior del árbol se visualizan los efectos del bajo rendimiento en el curso:

- Alto porcentaje de reprobación: Como resultado del bajo rendimiento, muchos estudiantes no logran aprobar el curso, lo que afecta su progreso académico.
- Desmotivación: La frustración y la falta de éxito en el curso resultan en una menor motivación por parte de los estudiantes, tanto para seguir aprendiendo programación como para participar en materias relacionadas.
- Dificultades en la comprensión y aplicación de conceptos: Los estudiantes que no desarrollan las habilidades de programación adecuadas enfrentan problemas para aplicar los conceptos aprendidos.
- Dificultad para aprender a programar: La falta de una base sólida dificulta el aprendizaje de la programación, lo que se convierte en un ciclo que alimenta el bajo rendimiento.
- No se logran entregar habilidades computacionales a cabalidad: Finalmente, los estudiantes no adquieren las habilidades computacionales que se espera que desarrollen a través del curso, lo que impacta su desempeño en futuros retos académicos y profesionales.

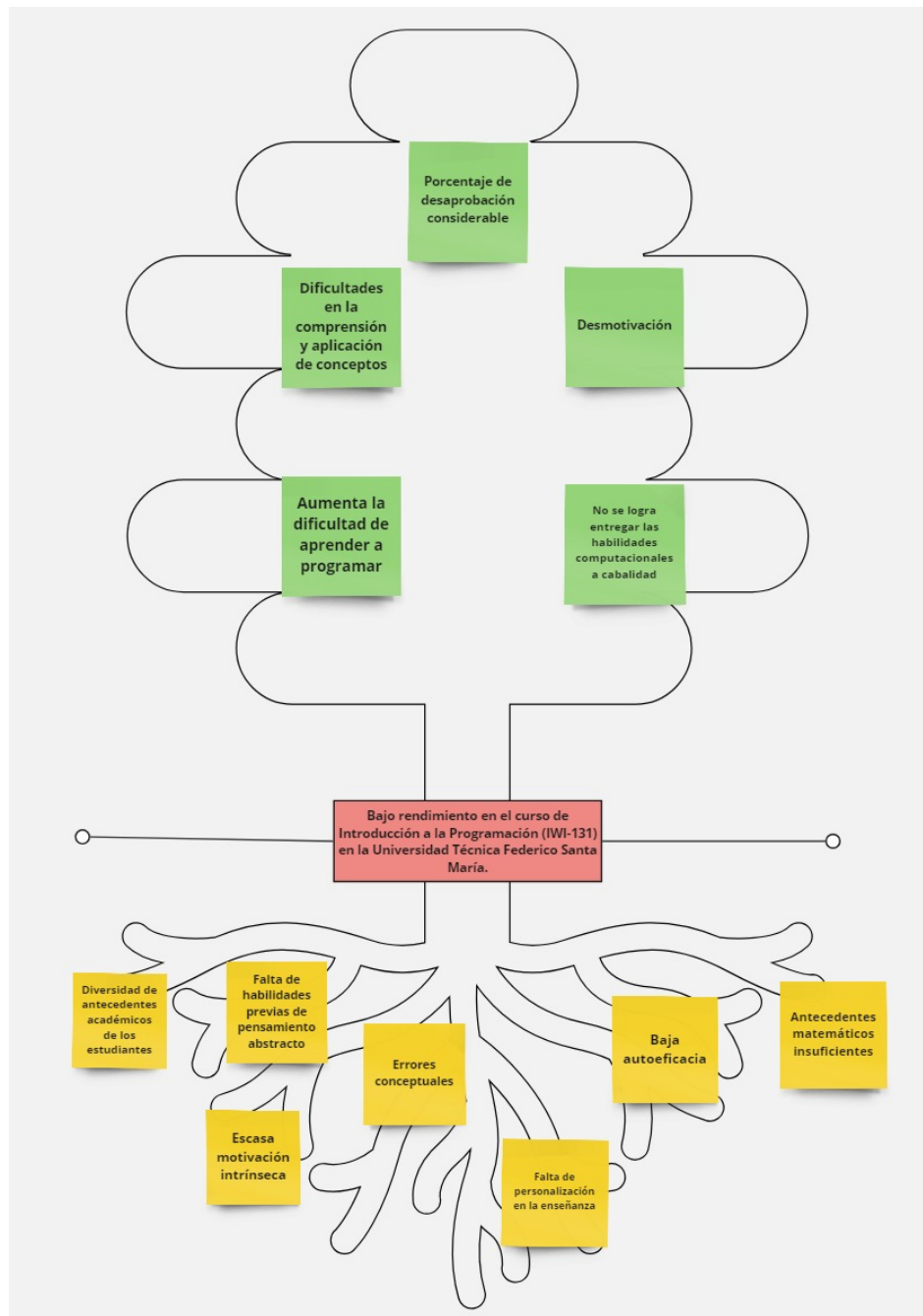


Figura 3: Árbol del problema
Fuente: Elaboración propia.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Autoeficacia

Según Bandura y Wessels [Bandura y Wessels, 1997], la autoeficacia es la creencia de una persona en su capacidad para ejecutar con éxito una tarea específica. Esta percepción influye directamente en cómo un individuo afronta los desafíos: si considera que no puede realizar una tarea, tenderá a evitarla o abordarla con menos esfuerzo, lo que disminuye significativamente las probabilidades de éxito. En contraste, aquellos con una alta autoeficacia tienden a enfrentar los desafíos con mayor motivación y persistencia, lo que incrementa sus posibilidades de éxito.

En el ámbito de la programación, esta creencia se refleja en la confianza de los estudiantes tanto a nivel general como en temas específicos. Los estudiantes que tienen una fuerte autoeficacia creen que son capaces de comprender y aplicar conceptos de programación, lo que los impulsa a dedicar más tiempo y esfuerzo en su aprendizaje. Por el contrario, aquellos con baja autoeficacia pueden sentir ansiedad, lo que a menudo lleva a evitar tareas más desafiantes y afecta negativamente su desempeño.

Bergin y Reilly [Bergin y Reilly, 2005] estudiaron la relación entre la autoeficacia y el rendimiento en estudiantes de programación, demostrando que los estudiantes con mayores niveles de autoeficacia tienden a obtener mejores resultados en los módulos evaluados. Este estudio subraya que la autoeficacia no solo influye en el rendimiento académico, sino que también está relacionada con la capacidad del estudiante para perseverar ante la complejidad de los problemas, gestionar mejor el estrés y adaptarse a los errores, que son inherentes al proceso de aprender a programar.

Estos hallazgos destacan la importancia de la percepción de competencia en el proceso de aprendizaje, influyendo tanto en el esfuerzo invertido como en la capacidad para superar obstáculos. Además, sugieren que fortalecer la autoeficacia a través de prácticas pedagógicas adecuadas, como la retroalimentación positiva y el refuerzo de pequeños logros, puede mejorar significativamente los resultados de los estudiantes en programación.

2.2. Antecedentes matemáticos

Según algunos autores, los antecedentes matemáticos pueden ser un buen indicador para determinar la afinidad inicial del estudiante con la programación [Ambrosio *et al.*, 2011]. Esto se debe a que tanto las matemáticas como la programación requieren habilidades similares, como el pensamiento lógico, la resolución de problemas y la capacidad de abstraer

conceptos complejos. La programación, al igual que las matemáticas, implica la descomposición de problemas grandes en partes más pequeñas, la identificación de patrones y la aplicación de reglas formales para alcanzar una solución, lo que genera una relación directa entre ambas disciplinas.

Estudios previos han sugerido que los estudiantes con un buen desempeño en matemáticas tienden a adaptarse mejor al pensamiento lógico, aumentando su rendimiento al programar [Owolabi *et al.*, 2014].

2.3. Motivación

La motivación es un factor crucial en el proceso de aprendizaje, especialmente en campos complejos como la programación. La habilidad para programar no solo requiere conocimientos técnicos y habilidades lógicas, sino también un alto nivel de persistencia y entusiasmo. La motivación, tanto intrínseca como extrínseca, influye significativamente en cómo los estudiantes abordan el aprendizaje y superan los desafíos asociados con la programación.

La motivación intrínseca se refiere al deseo interno de aprender y resolver problemas por el simple placer de hacerlo. Los estudiantes motivados intrínsecamente suelen estar más comprometidos y dispuestos a dedicar tiempo y esfuerzo a la programación. Según Ryan y Deci [Ryan y Deci, 2000], cuando los individuos están motivados intrínsecamente, tienden a experimentar un mayor sentido de competencia y satisfacción, lo que puede llevar a una mayor persistencia y éxito en la programación.

Por otro lado, la motivación extrínseca involucra factores externos, como recompensas, reconocimiento o la presión para cumplir con ciertos estándares. Aunque esta forma de motivación puede ser eficaz para iniciar el proceso de aprendizaje, los estudios sugieren que una dependencia excesiva de recompensas externas puede limitar la profundidad del compromiso y la satisfacción personal con la programación [Ryan y Deci, 2000]. Sin embargo, un equilibrio entre la motivación intrínseca y extrínseca puede ser ideal para mantener el interés y el esfuerzo en la programación.

La investigación también muestra que la motivación tiene un impacto directo en la autoeficacia del estudiante, que es la creencia en su capacidad para realizar tareas específicas. Los estudiantes que están motivados tienden a establecer metas más altas y a perseverar ante las dificultades, lo que a su vez refuerza su autoeficacia y fomenta un ciclo positivo de aprendizaje y éxito [Bandura y Wessels, 1997].

2.4. Pensamiento Computacional

El término Pensamiento Computacional (Computational Thinking) fue propuesto por Jeanette Wing en 2006 como “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” [Wing, 2006]. Desde esa fecha, se han generalizado distintas acepciones para el término, desde diversas perspectivas y contemplando diferentes habilidades cognitivas. Tang et al. [Tang et al., 2020] presentan una taxonomía de estas definiciones.

En este trabajo, y en línea con la evaluación realizada en la Universidad Técnica Federico Santa María, se considerarán las cuatro habilidades cognitivas principales para las que fue intencionado el test de pensamiento computacional aplicado en la institución [Meza et al., 2024]. Estas habilidades son: Abstracción, Descomposición, Reconocimiento de Patrones y Generalización, y Pensamiento Algorítmico.

2.4.1. Abstracción

La capacidad de abstracción es uno de los pilares más importantes en el aprendizaje de la programación. A medida que los estudiantes se enfrentan a problemas complejos, la abstracción les permite descomponer estos problemas en partes más manejables y generar soluciones más eficientes. La programación, en su esencia, requiere que los individuos se distancien de los detalles específicos y se concentren en conceptos generales que puedan ser aplicados en múltiples contextos [Fonden Calzadilla, 2020].

Durante el proceso de aprendizaje, los estudiantes a menudo encuentran dificultades para comprender cómo aplicar la teoría a problemas prácticos. La abstracción les ayuda a superar este obstáculo al permitirles identificar patrones comunes y crear modelos mentales simplificados de cómo deben funcionar los programas. Por ejemplo, cuando un estudiante aprende sobre estructuras de datos, como listas o árboles, no se enfoca en los detalles específicos de la implementación, sino en cómo estas estructuras organizan y manipulan la información de manera eficiente. Esta habilidad de abstraer es lo que les permite comprender el concepto en un nivel más profundo y aplicar ese conocimiento en una variedad de situaciones [Fonden Calzadilla, 2020].

Uno de los desafíos más comunes para los principiantes en programación es manejar grandes cantidades de información sin perder de vista el propósito general del programa. La capacidad de abstracción permite a los estudiantes evitar el “atasco” en detalles irrelevantes y centrarse en las ideas clave que guiarán la solución del problema. Por ejemplo, en lugar de centrarse en cada línea de código por separado, un estudiante puede aprender a visualizar el propósito general de una función o un algoritmo, comprendiendo cómo cada componente contribuye al todo. Este enfoque no solo facilita el proceso de codificación, sino que también mejora la capacidad de depuración y optimización del código.

Además, la abstracción es crucial para el diseño de programas que sean modulares y escalables, dos conceptos fundamentales en el aprendizaje de la programación. Los estudiantes que desarrollan una fuerte capacidad de abstracción son más capaces de diseñar programas que pueden dividirse en pequeños componentes reutilizables, mejorando la eficiencia tanto en el desarrollo como en la comprensión de cómo funciona el software [Kramer, 2007].

La habilidad de abstraer también se relaciona con la resolución de problemas. Los estudiantes que dominan la abstracción tienden a ser más efectivos al enfrentarse a problemas nuevos y desafiantes, ya que son capaces de identificar patrones y aplicar soluciones que han aprendido en contextos anteriores. En este sentido, la abstracción no solo simplifica el proceso de aprendizaje, sino que también fomenta la capacidad de los estudiantes para transferir conocimientos a diferentes áreas de la programación [Edgar y Antonio, 2014].

2.4.2. Descomposición

La descomposición es la habilidad de dividir un problema complejo o un sistema grande en partes más pequeñas, manejables e interdependientes [Shute *et al.*, 2017]. Esta estrategia es fundamental en la programación, ya que los problemas del mundo real rara vez son lo suficientemente simples como para ser abordados de una sola vez. Al descomponer un problema, los programadores pueden concentrarse en resolver cada subproblema de forma aislada, lo que reduce la complejidad general y facilita el proceso de diseño y codificación.

En el contexto del aprendizaje de la programación, la descomposición es crucial para que los estudiantes no se sientan abrumados por la magnitud de una tarea. Por ejemplo, al desarrollar una aplicación, un estudiante puede descomponerla en módulos como la interfaz de usuario, la lógica de negocio y la gestión de datos. Cada uno de estos módulos puede ser desarrollado y probado de forma independiente antes de integrarse en la solución completa. Esta aproximación modular no solo simplifica la resolución, sino que también mejora la legibilidad y el mantenimiento del código, ya que cada parte tiene una responsabilidad clara y limitada.

2.4.3. Reconocimiento de Patrones y Generalización

El reconocimiento de patrones implica la capacidad de identificar similitudes, tendencias o regularidades dentro de problemas o conjuntos de datos. Una vez que se reconoce un patrón, la generalización es el proceso de abstraer ese patrón para aplicarlo a situaciones nuevas o más amplias [Shute *et al.*, 2017].

Para los estudiantes de programación, el reconocimiento de patrones se manifiesta al identificar tareas repetitivas que pueden ser automatizadas mediante bucles o funciones. Por ejemplo, si una serie de operaciones se realiza varias veces con diferentes datos de entrada, un estudiante con esta habilidad reconocerá el patrón y creará una función reutilizable en

lugar de duplicar el código. La generalización, por su parte, les permite diseñar algoritmos que no solo resuelven un caso específico, sino que pueden aplicarse a una clase más amplia de problemas. Esto es fundamental en el diseño de algoritmos eficientes y escalables.

La falta de reconocimiento de patrones a menudo lleva a código redundante y menos eficiente, lo que dificulta la depuración y el mantenimiento. Por el contrario, los estudiantes que desarrollan esta habilidad son capaces de escribir programas más concisos, elegantes y fáciles de entender. Además, esta competencia es crucial para comprender y aplicar paradigmas de programación como la programación orientada a objetos (POO), donde se identifican patrones de comportamiento y atributos comunes entre entidades para crear clases y objetos reutilizables.

2.4.4. Pensamiento Algorítmico

El pensamiento algorítmico es una forma lógica y organizada de pensar utilizada para descomponer un objetivo complicado en una serie de pasos (ordenados) usando las herramientas disponibles [Lockwood *et al.*, 2016]. Es el núcleo de la programación, ya que un programa es, en esencia, un algoritmo expresado en un lenguaje que una computadora puede entender. Esta habilidad implica no solo la capacidad de diseñar los pasos, sino también de comprender cómo se ejecutarán, prever posibles resultados y optimizar la eficiencia.

Para los estudiantes, desarrollar el pensamiento algorítmico significa aprender a traducir un problema del lenguaje natural a una serie de instrucciones precisas y sin ambigüedades que una máquina pueda seguir. Esto incluye la definición de entradas, procesos y salidas, así como la gestión de condiciones (decisiones) y repeticiones (iteraciones). Por ejemplo, al crear un programa para ordenar una lista de números, el pensamiento algorítmico implica definir cada paso: comparar elementos, intercambiarlos si es necesario, y repetir hasta que la lista esté ordenada.

Uno de los principales desafíos para los principiantes es la transición de un pensamiento intuitivo a un pensamiento estructurado y formal. Los errores lógicos en los programas a menudo se derivan de fallas en el diseño algorítmico, como la omisión de casos límite, el uso incorrecto de estructuras de control o la ineficiencia en la secuencia de operaciones. Fortalecer el pensamiento algorítmico permite a los estudiantes no solo escribir código funcional, sino también comprender su comportamiento, depurarlo eficazmente y mejorarlo. Esta habilidad es transferible y beneficia la resolución de problemas en cualquier disciplina, ya que enseña a abordar desafíos de manera sistemática y lógica [Shute *et al.*, 2017].

2.5. Errores conceptuales

Errores conceptuales en el aprendizaje de la programación se refiere a las interpretaciones incorrectas o incompletas que los estudiantes pueden tener sobre ciertos conceptos, lo que impacta negativamente su desempeño. Estas concepciones erróneas a menudo derivan de una comprensión limitada o incorrecta de principios fundamentales de la programación, como el control de flujo, el uso de variables y la lógica condicional.

Los errores conceptuales son ideas previas o suposiciones incorrectas que los estudiantes llevan consigo al enfrentarse a nuevos temas. En el caso de la programación, estas ideas erróneas pueden afectar la manera en que los estudiantes abordan problemas de programación, llevando a la creación de soluciones incorrectas o ineficientes. Como menciona Kwon [Kwon, 2017], estas equivocaciones suelen reflejarse en los planes de resolución de problemas, ya que los estudiantes aplican modelos mentales incorrectos para intentar resolver desafíos algorítmicos.

Diversos estudios han identificado que las misconceptions en programación pueden originarse de varios factores. Uno de ellos es el uso cotidiano de conceptos que, aunque parecen similares, funcionan de manera distinta en el contexto de la programación. Por ejemplo, la confusión sobre el funcionamiento de los bucles y las estructuras condicionales proviene de las expectativas que los estudiantes traen del lenguaje natural o del razonamiento lógico cotidiano [Sirkiä y Sorva, 2012].

2.6. Comprensión Lectora

En el ámbito de la programación, la comprensión lectora juega un papel crucial en la habilidad de los programadores para interpretar y manipular el código de manera efectiva. Según el estudio “The Search for Meaning: Inferential Strategic Reading Comprehension in Programming” [Kallia, 2023], la comprensión lectora en programación va más allá de la simple decodificación de sintaxis y estructuras; involucra una interpretación profunda de la lógica subyacente y la intención del código.

El estudio resalta que los programadores exitosos no solo leen el código de manera lineal, sino que emplean estrategias inferenciales para captar el significado implícito y las relaciones entre diferentes componentes del código. Esta capacidad para hacer inferencias y construir una representación mental coherente del funcionamiento del código es esencial para resolver problemas complejos y para la depuración efectiva.

Además, la comprensión estratégica en la programación implica la habilidad de identificar patrones y anticipar el comportamiento del código en función de las modificaciones realizadas. Este enfoque inferencial permite a los programadores adaptarse a nuevas situaciones y comprender cómo los cambios en el código afectan el sistema en su conjunto. El estudio sugiere que el desarrollo de estas habilidades de comprensión lectora inferencial puede ser

un factor determinante en la eficacia del aprendizaje de la programación, ya que facilita una comprensión más profunda y una resolución de problemas más eficiente.

La comprensión lectora en programación es un proceso multifacético que incluye la decodificación de sintaxis, la interpretación inferencial y la aplicación estratégica de conocimientos. Las habilidades de lectura inferencial y estratégica son fundamentales para el aprendizaje y la práctica efectiva de la programación, y su desarrollo puede ser clave para mejorar la competencia en este campo.

2.7. Prueba de Acceso a la Educación Superior (PAES)

Según la página de ChileAtiende [ChileAtiende, 2025], la Prueba de Acceso a la Educación Superior (PAES) es una evaluación estandarizada implementada en Chile, que tiene como objetivo principal medir habilidades y conocimientos esenciales para el desempeño académico en la educación superior. Esta prueba, que reemplaza a la antigua PSU (Prueba de Selección Universitaria), se orienta hacia un modelo más inclusivo y equitativo, adaptado a las necesidades actuales de los estudiantes y al contexto educativo del país.

2.7.1. Estructura de la PAES

La PAES está compuesta por diversas pruebas que evalúan competencias clave en áreas fundamentales para el aprendizaje y el desarrollo profesional. Las pruebas obligatorias incluyen:

- Competencia Matemática (M1): Evalúa habilidades básicas en matemáticas, como razonamiento numérico y resolución de problemas cotidianos.
- Comprensión Lectora: Mide la capacidad para interpretar y analizar textos de diversas tipologías y complejidades.

Además, los postulantes pueden rendir pruebas electivas que profundizan en áreas específicas, como:

- Competencia Matemática (M2): Dirigida a estudiantes que aspiran a carreras con mayor requerimiento matemático.
- Historia y Ciencias Sociales.
- Ciencias: Ofrece opciones en biología, física y química.

2.7.2. Relevancia de la PAES

La PAES busca medir competencias transversales y específicas que son fundamentales para el aprendizaje en la educación superior. Al enfocarse en habilidades como el razonamiento lógico, la comprensión lectora y el análisis crítico, esta prueba no solo sirve como un mecanismo de admisión a las universidades chilenas, sino también como un indicador del nivel de preparación de los estudiantes para enfrentar desafíos académicos más avanzados.

2.8. El Test de Lawson en el Contexto de la Programación

El **Test de Lawson de Habilidades de Razonamiento Científico** (LCTSR, por sus siglas en inglés) es una herramienta diseñada para evaluar el **razonamiento científico** de los individuos [Bao *et al.*, 2018]. Si bien la referencia se enfoca en el uso de este test en la educación en física, sus principios subyacentes son altamente relevantes para el ámbito de la programación, dada la naturaleza intrínsecamente lógica y abstracta de esta disciplina.

El LCTSR evalúa habilidades de razonamiento que son fundamentales para el éxito en la programación, tales como:

- **Razonamiento proporcional:** Es crucial para comprender la escala y las relaciones entre diferentes componentes de un programa, por ejemplo, cómo una variable afecta el resultado en proporción a otra.
- **Control de variables:** Al igual que en un experimento científico, en programación es vital aislar y manipular variables para entender su impacto, depurar errores y optimizar el código. Esta habilidad se relaciona directamente con la identificación y corrección de **errores conceptuales**, que a menudo surgen de una comprensión inadecuada de cómo las variables interactúan.
- **Razonamiento probabilístico:** Aunque menos directo que los anteriores, es útil para entender el comportamiento de algoritmos en escenarios inciertos o para evaluar la eficiencia de diferentes enfoques.
- **Pensamiento hipotético-deductivo:** Esta es una habilidad central en la programación, ya que implica formular hipótesis sobre el comportamiento del código, probarlas y deducir conclusiones para corregir o mejorar el programa.

2.9. Trabajos previos

Un trabajo previo corresponde al de Francisco Vásquez Pinto [Pinto, 2020], en el cual realiza un estudio comparativo sobre el impacto de utilizar un lenguaje de programación basado

en bloques, como Snap!, frente a un lenguaje de programación textual, como Python, en el desarrollo de habilidades de programación en estudiantes de primer año de ingeniería. La investigación fue llevada a cabo en la Universidad Técnica Federico Santa María (UTFSM) y abarcó un análisis tanto del desempeño académico como de la motivación de los estudiantes. Los resultados mostraron que los estudiantes que usaron Snap! exhibieron un mayor nivel de motivación y disfrute con el curso de programación en comparación con el grupo que utilizó Python. Sin embargo, en términos de rendimiento académico, ambos grupos obtuvieron resultados similares. El estudio también aborda las “misconceptions” o errores conceptuales comunes entre los estudiantes al aprender a programar, identificando que el uso de un lenguaje visual podría facilitar la superación de ciertos errores, proporcionando una alternativa pedagógica efectiva para los cursos introductorios de programación a nivel universitario. [Pinto, 2020]

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. Metodología

La presente investigación emplea una **metodología cuantitativa, de corte transversal y diseño no experimental**, con el objetivo de identificar y analizar los factores que inciden en el aprendizaje de programación. Se busca establecer relaciones y asociaciones entre diversas variables académicas, cognitivas y psicológicas, y el rendimiento de los estudiantes en un curso introductorio de programación.

3.1.1. Participantes y Contexto del Estudio

El estudio se llevó a cabo con estudiantes de primer año de diversas carreras universitarias de la Universidad Técnica Federico Santa María en el año 2025. Los datos provienen de distintas fuentes institucionales y de encuestas diseñadas específicamente para esta investigación, garantizando una muestra heterogénea que permite explorar la influencia de múltiples factores en el rendimiento académico en programación. Se realizó un análisis descriptivo del rendimiento en el primer certamen de programación según carrera y campus para contextualizar el desempeño de los estudiantes.

3.2. Uso de datos

La utilización de los datos recopilados se llevó a cabo bajo una estricta política de seguridad. Solo el profesor guía (Federico Meza) y el memorista (Ernesto Barría) tuvieron acceso a estos datos. Antes de su entrega, se aplicó un proceso de anonimización, mediante el cual la información sensible fue reemplazada por un identificador autoincremental, con el fin de proteger la privacidad de los estudiantes. Para su uso en el código, se empleó la función `gs4_auth()`, la cual requiere autenticarse con una cuenta que tenga permisos de acceso a los datos almacenados en Google Sheets. Cabe destacar que los datos nunca fueron almacenados localmente, manteniéndose siempre en la nube para resguardar su confidencialidad.

3.3. Recopilación de Datos

La recolección de datos se realizó a partir de múltiples fuentes para obtener una visión integral de los factores de interés. Estos datos incluyen información académica previa, medicio-

nes de pensamiento computacional, variables psicológicas como la motivación y la autoeficacia, y datos cualitativos como son el campus y la carrera.

3.3.1. Datos de Ingreso Académico Previo

Se obtuvieron datos académicos de ingreso correspondientes a la generación 2025, proporcionados por la universidad. Estos incluyen:

Puntaje NEM

El **NEM** corresponde al puntaje obtenido a partir del **promedio de las calificaciones de enseñanza media** del estudiante, considerando los años desde primero a cuarto medio. Este puntaje es estandarizado por el Departamento de Evaluación, Medición y Registro Educativo (DEMRE) y puede tomar valores entre 100 y 1000 puntos. Su cálculo depende de una tabla de conversión oficial que varía según el tipo de establecimiento educacional, permitiendo una comparación equitativa entre estudiantes de distintos contextos educativos.

Estadística	Valor
Total de Datos (N)	4406
Promedio	805.52
Desviación Estándar	124.5455
Varianza	15511.58

Tabla 1: Estadísticas descriptivas de los puntajes NEM

Puntaje Ranking

El **Ranking de Notas** es un puntaje ponderado que evalúa el rendimiento de un estudiante en comparación con el **promedio histórico de las notas de su propio colegio**. Es decir, no solo considera las calificaciones individuales del postulante, sino también cómo esas notas se posicionan respecto al desempeño de las generaciones anteriores de su mismo establecimiento. Este puntaje va de 100 a 1000 puntos y busca contextualizar el rendimiento académico del estudiante dentro de su entorno educativo.

Estadística	Valor
Total de Datos (N)	4406
Promedio	824.51
Desviación Estándar	134.1685
Varianza	18001.19

Tabla 2: Estadísticas descriptivas de los puntajes ranking

Puntaje Matemática M1

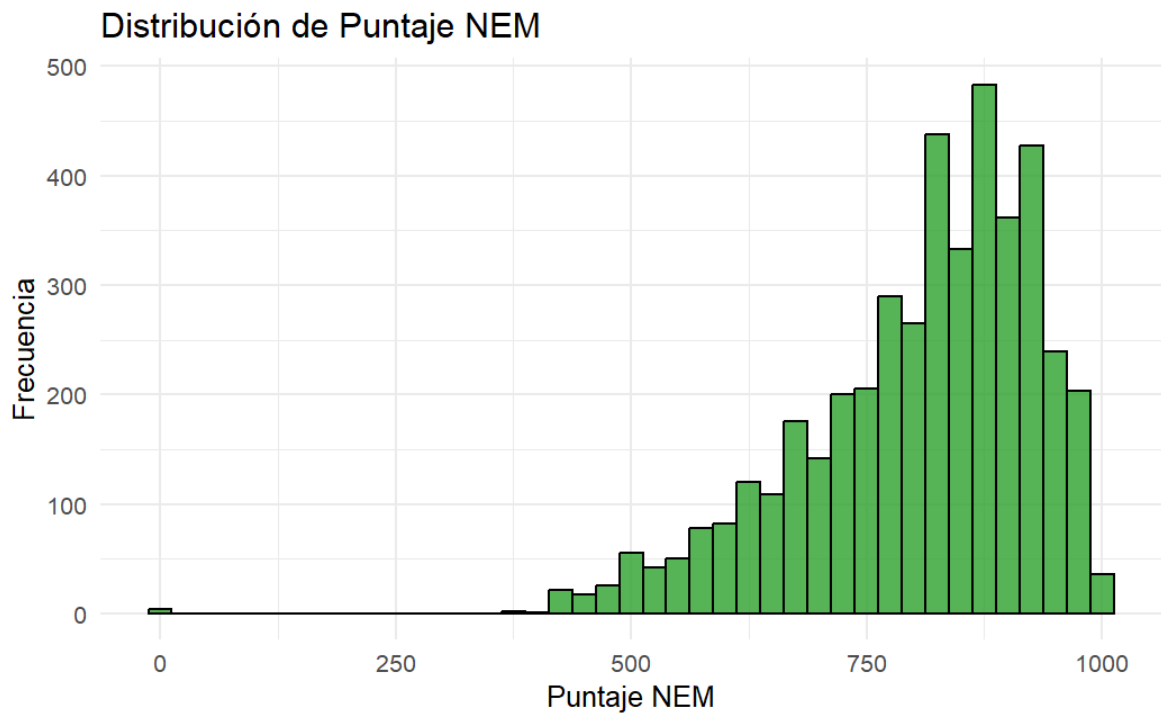


Figura 4: Distribución de puntajes NEM

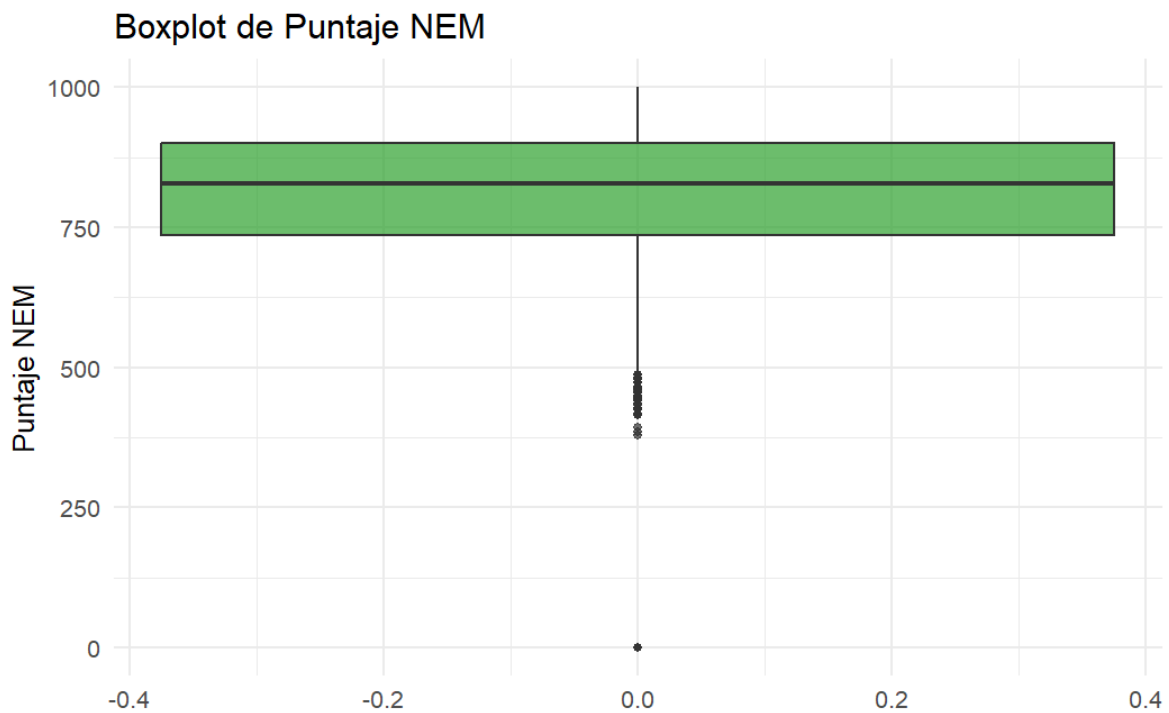


Figura 5: Boxplot de puntajes NEM

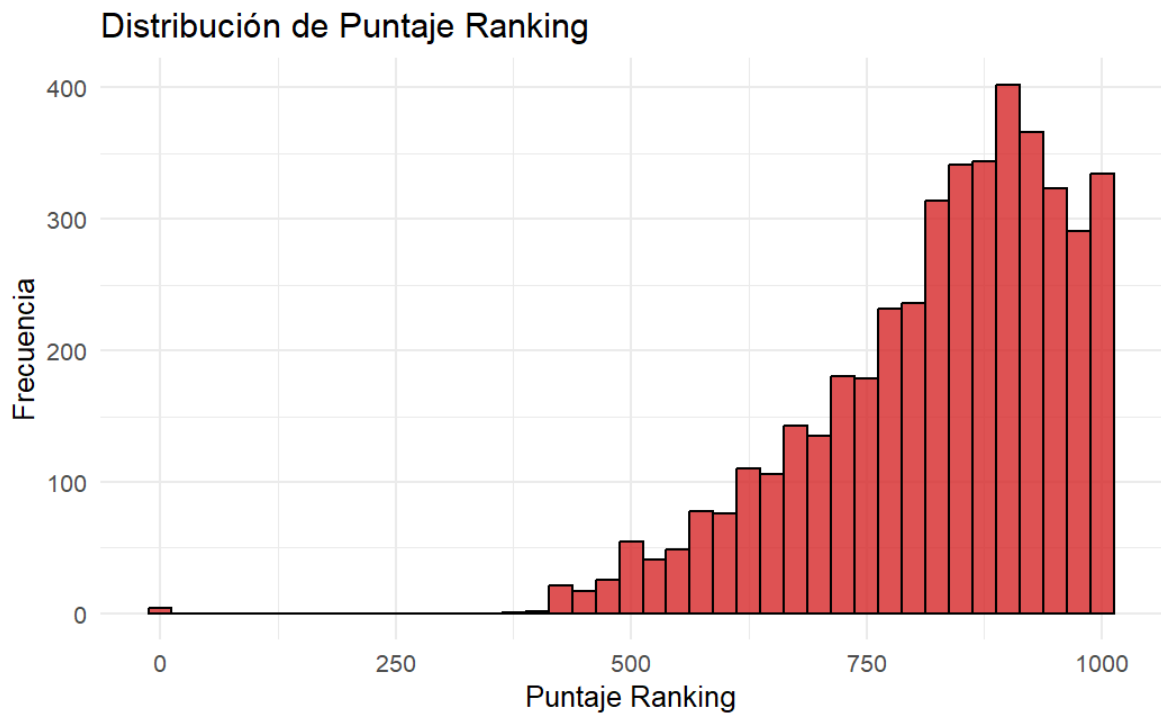


Figura 6: Histograma de puntajes ranking

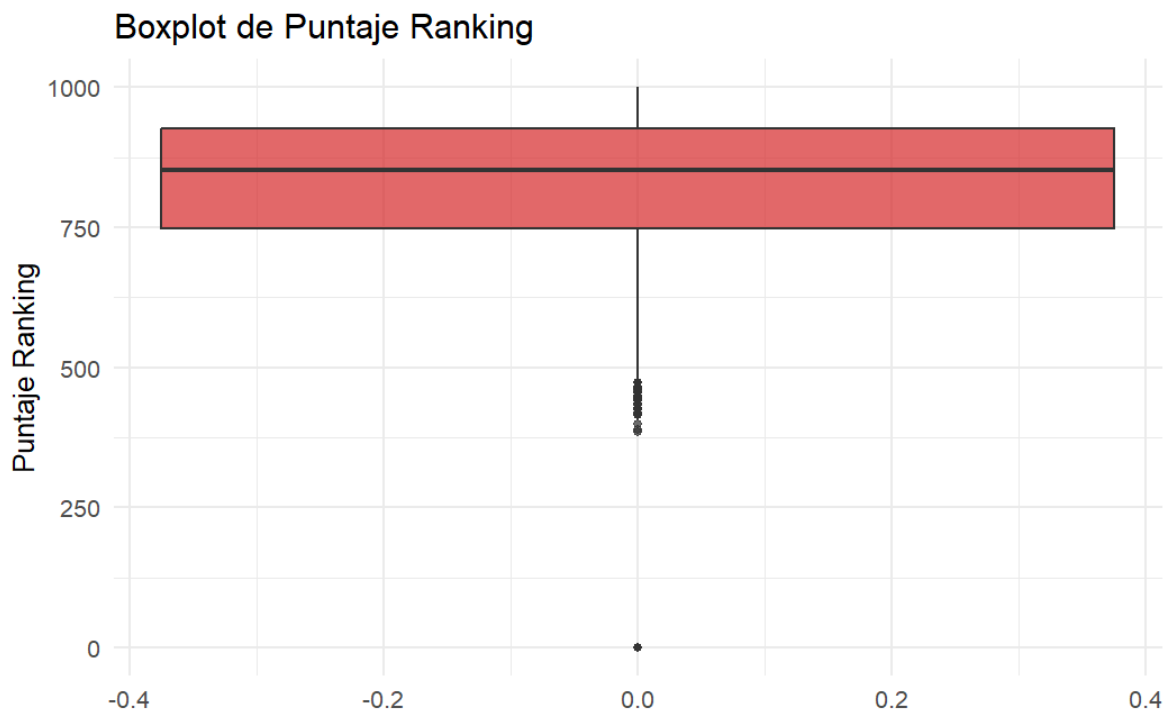


Figura 7: Boxplot de puntajes ranking

El **Puntaje Matemática M1** corresponde a los resultados de la prueba de matemáticas que evalúa competencias y habilidades matemáticas fundamentales. Esta prueba busca medir la capacidad de los estudiantes para aplicar conceptos matemáticos básicos en situaciones de la vida cotidiana y para abordar estudios universitarios que requieran una base matemática estándar. Esta prueba, al igual que NEM y Ranking, puede tomar valores de 100 a 1000 puntos.

Estadística	Valor
Total de Datos (N)	4406
Promedio	769.44
Desviación Estándar	197.5468
Varianza	39024.74

Tabla 3: Estadísticas descriptivas de los puntajes de la prueba de matemáticas M1

Puntaje Matemática M2

El **Puntaje Matemática M2** corresponde a los resultados de una prueba de matemáticas más avanzada, diseñada para evaluar competencias necesarias en carreras con una fuerte base matemática. Esta prueba es particularmente relevante para disciplinas del área de tecnología y ciencias básicas, donde se requiere un nivel de razonamiento y aplicación matemática superior. Esta prueba puede tomar valores de 100 a 1000 puntos.

Estadística	Valor
Total de Datos (N)	4406
Promedio	440.29
Desviación Estándar	206.7897
Varianza	42762

Tabla 4: Estadísticas descriptivas de los puntajes de la prueba de matemáticas M2

Puntaje Lenguaje

El **Puntaje Lenguaje** es la calificación obtenida en la prueba de lenguaje, la cual mide la capacidad del estudiante para interpretar, analizar y comprender textos. Esta prueba evalúa habilidades de comprensión lectora, vocabulario, y la capacidad de extraer información relevante y realizar inferencias a partir de diversos tipos de escritos. Esta prueba puede tomar valores de 100 a 1000 puntos.

Puntaje Ciencias

El **Puntaje Ciencias** es la puntuación obtenida en la prueba de ciencias. Esta evaluación mide el conocimiento y la comprensión de los estudiantes en diversas áreas científicas, como biología, física y química, así como su capacidad para aplicar principios científicos y razonamiento en la resolución de problemas. Esta prueba puede tomar valores de 100 a 1000 puntos.

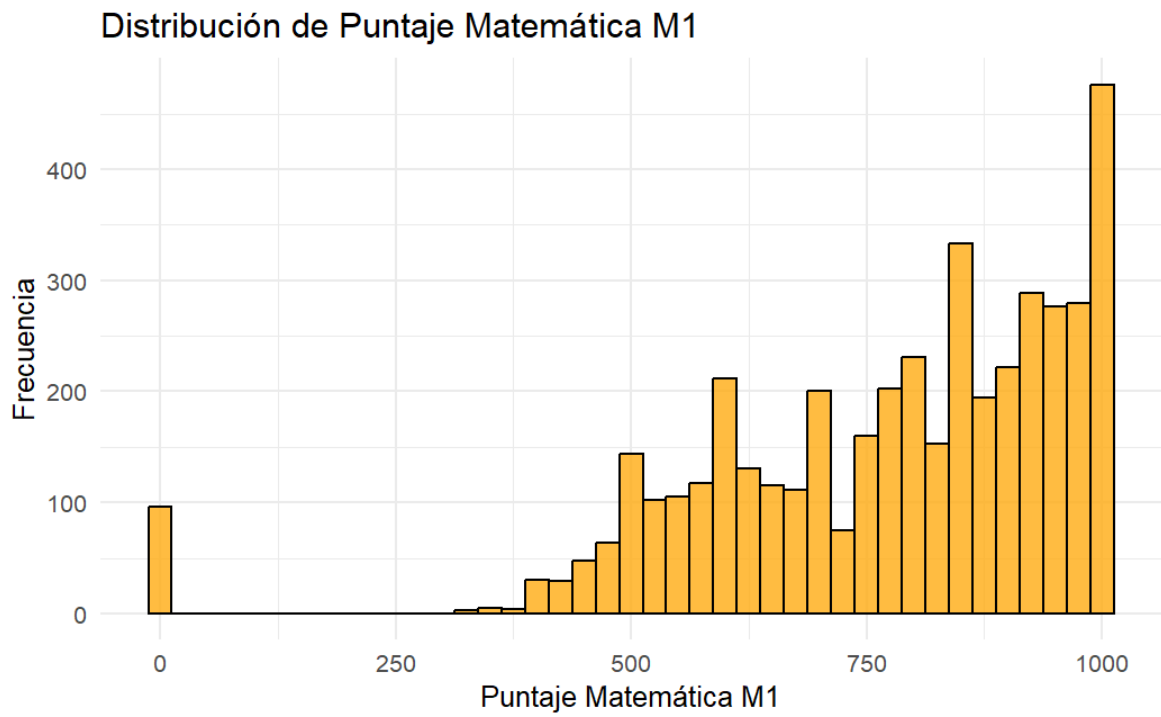


Figura 8: Distribución de puntajes Matemática M1

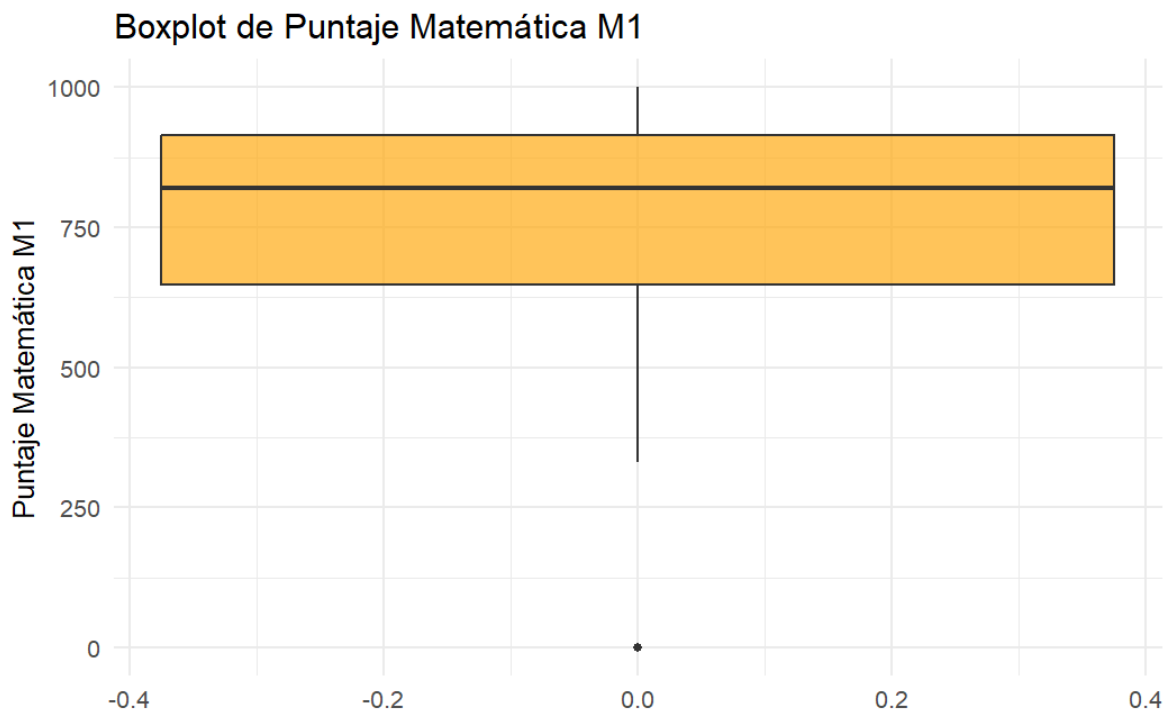


Figura 9: Boxplot de puntajes Matemática M1

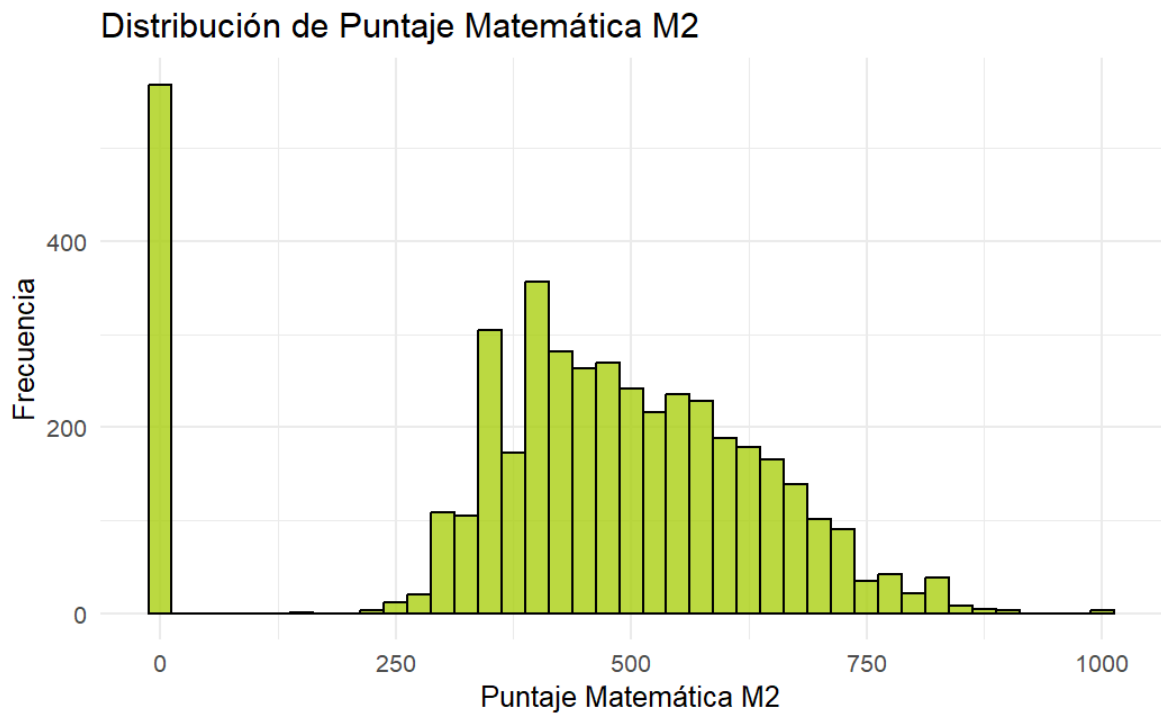


Figura 10: Distribución de puntajes Matemática M2

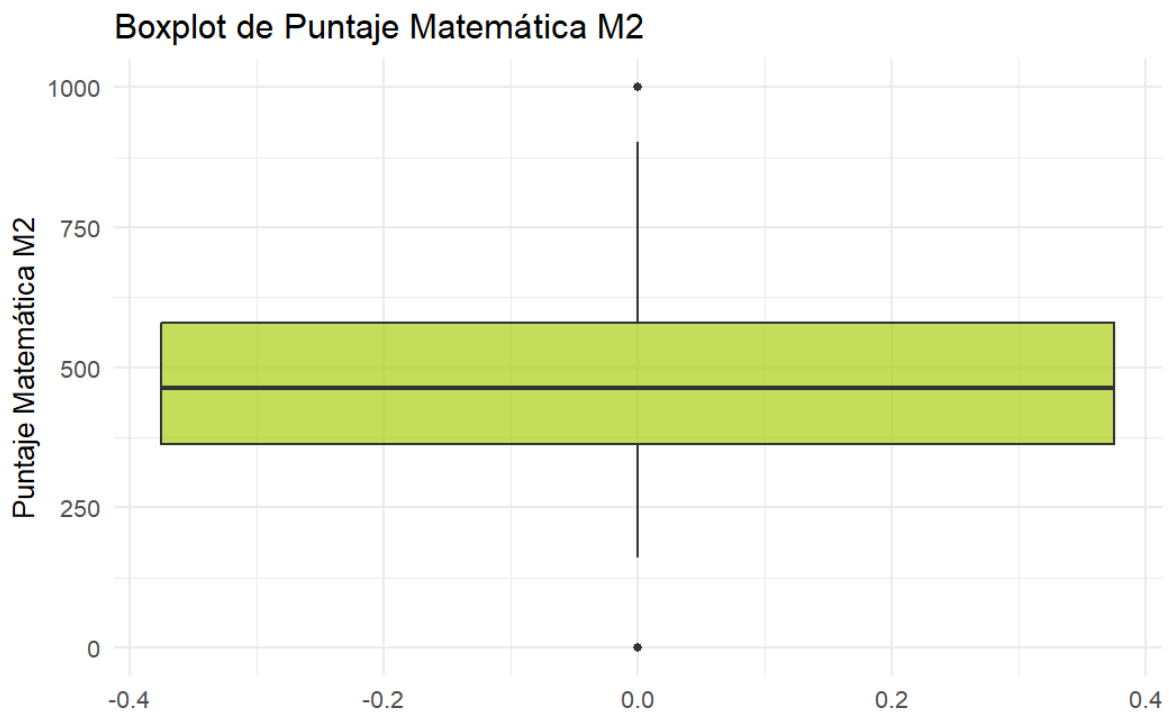


Figura 11: Boxplot de puntajes Matemática M2

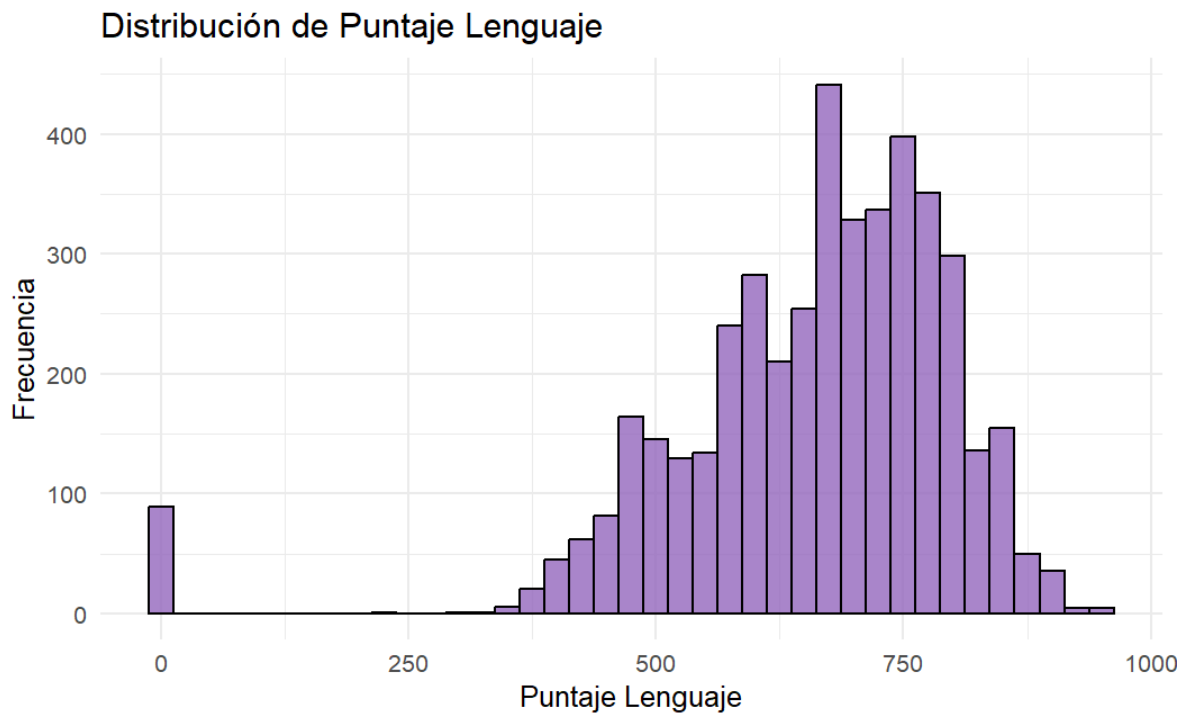


Figura 12: Distribución de puntajes Lenguaje

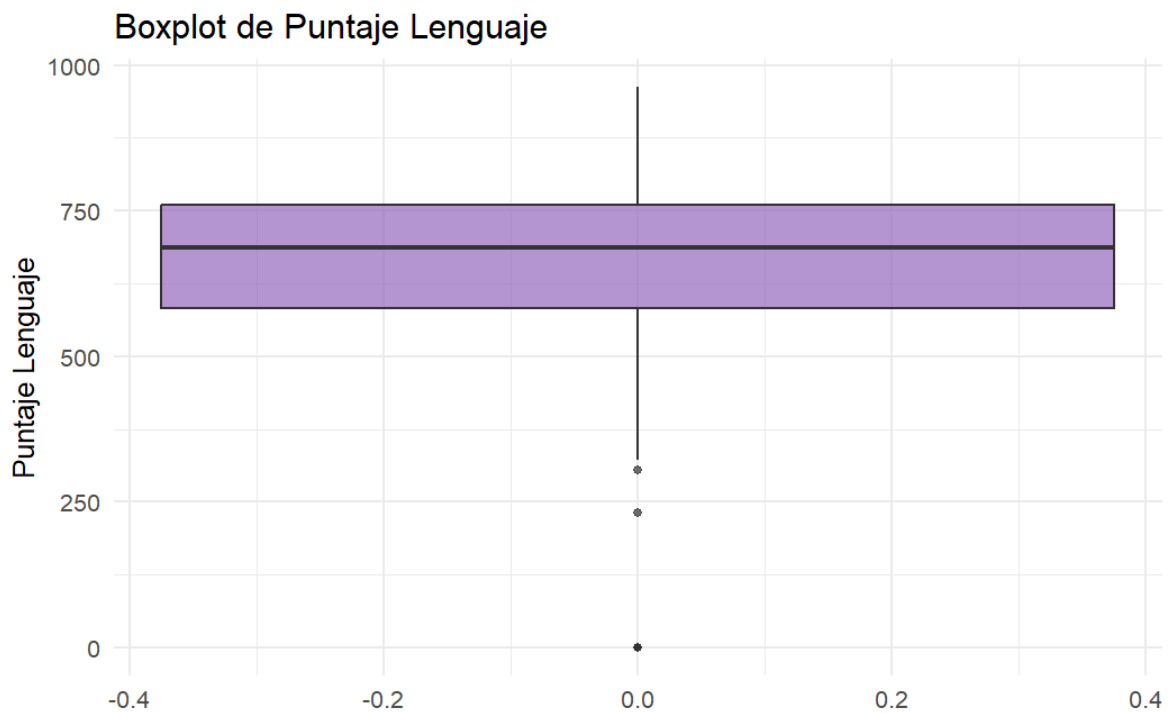


Figura 13: Boxplot de puntajes Lenguaje

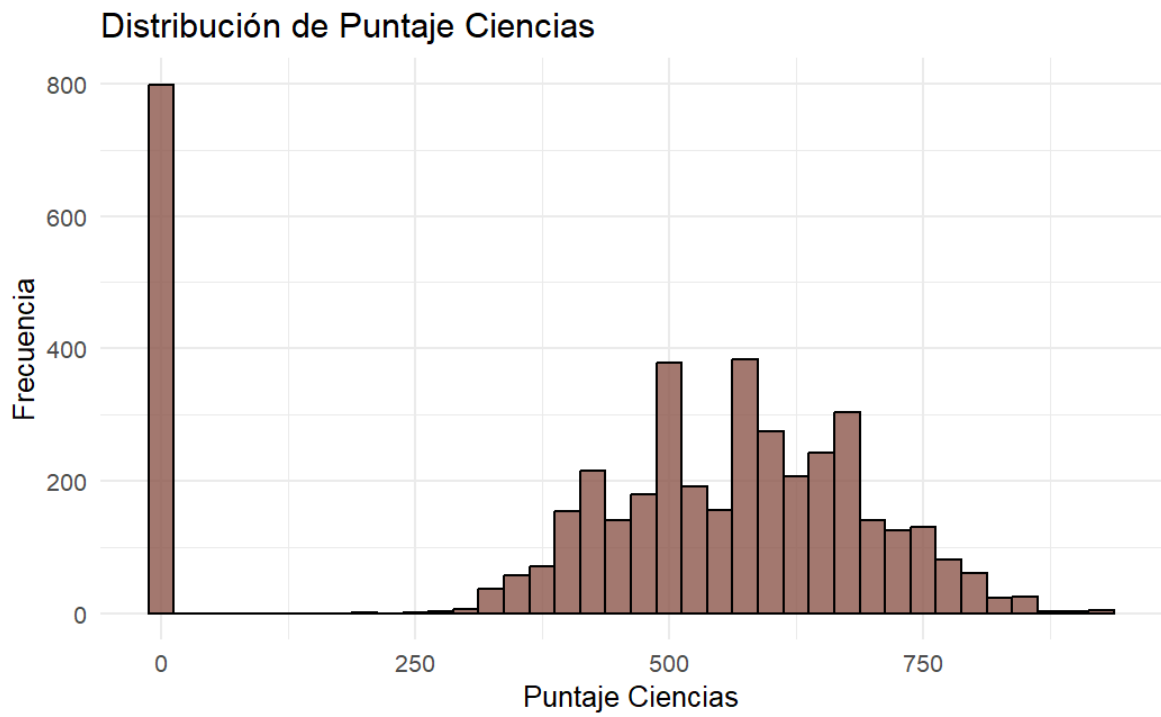


Figura 14: Distribución de puntajes Ciencias

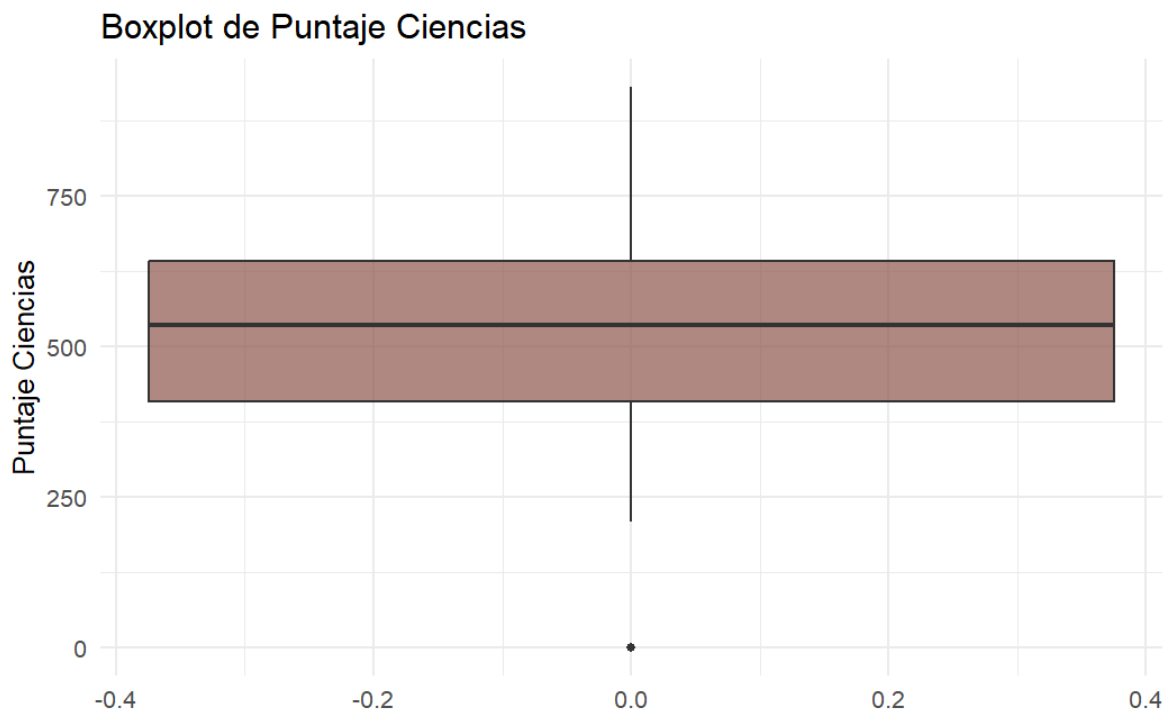


Figura 15: Boxplot de puntajes Ciencias

Estadística	Valor
Total de Datos (N)	4406
Promedio	658.21
Desviación Estándar	149.5254
Varianza	22357.83

Tabla 5: Estadísticas descriptivas de 'Puntaje Lenguaje'

Estadística	Valor
Total de Datos (N)	4406
Promedio	469.78
Desviación Estándar	245.6002
Varianza	60319.46

Tabla 6: Estadísticas descriptivas de los puntajes de la prueba de Ciencias

Puntaje Historia

El **Puntaje Historia** corresponde a la calificación obtenida en la prueba de historia y ciencias sociales. Esta prueba evalúa el conocimiento de los estudiantes sobre eventos históricos, procesos sociales, geografía y educación ciudadana, así como su capacidad para analizar y contextualizar información de estas disciplinas. Esta prueba puede tomar valores de 100 a 1000 puntos.

Estadística	Valor
Total de Datos (N)	4406
Promedio	254.61
Desviación Estándar	291.9115
Varianza	85212.31

Tabla 7: Estadísticas descriptivas de los puntajes de la prueba de Historia

3.3.2. Test de Lawson

El **Test de Lawson** es una prueba estandarizada diseñada específicamente para medir las habilidades de **razonamiento científico** de los individuos. Evalúa la capacidad para comprender y aplicar conceptos lógicos y principios de la investigación científica, como la formulación de hipótesis, el control de variables y la interpretación de resultados. Esta prueba puede tomar valores de 0 % a 100 %. Los resultados de este test, administrado a los estudiantes del campus Casa Central, se incorporaron al conjunto de datos para evaluar su relación con el rendimiento en programación, bajo la siguiente premisa entregada por el departamento de física del campus Casa Central:

“Los estudiantes con puntaje inferior al 33 % cuentan solo con habilidades de razonamiento

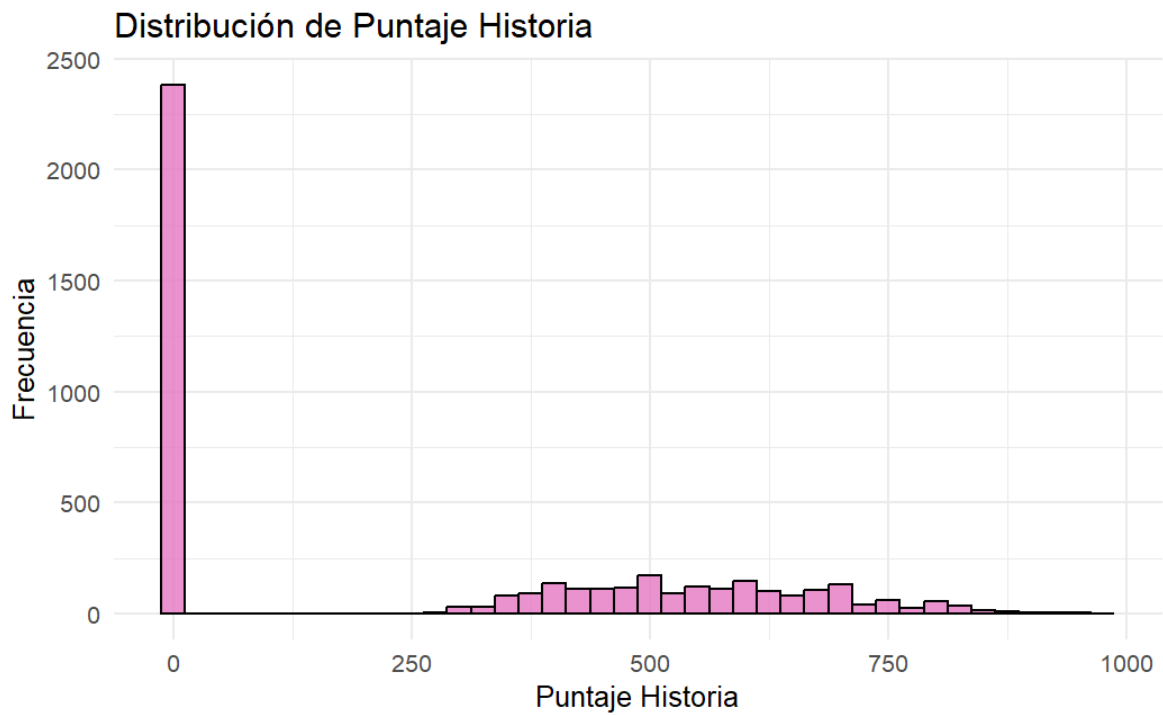


Figura 16: Distribución de puntajes Historia

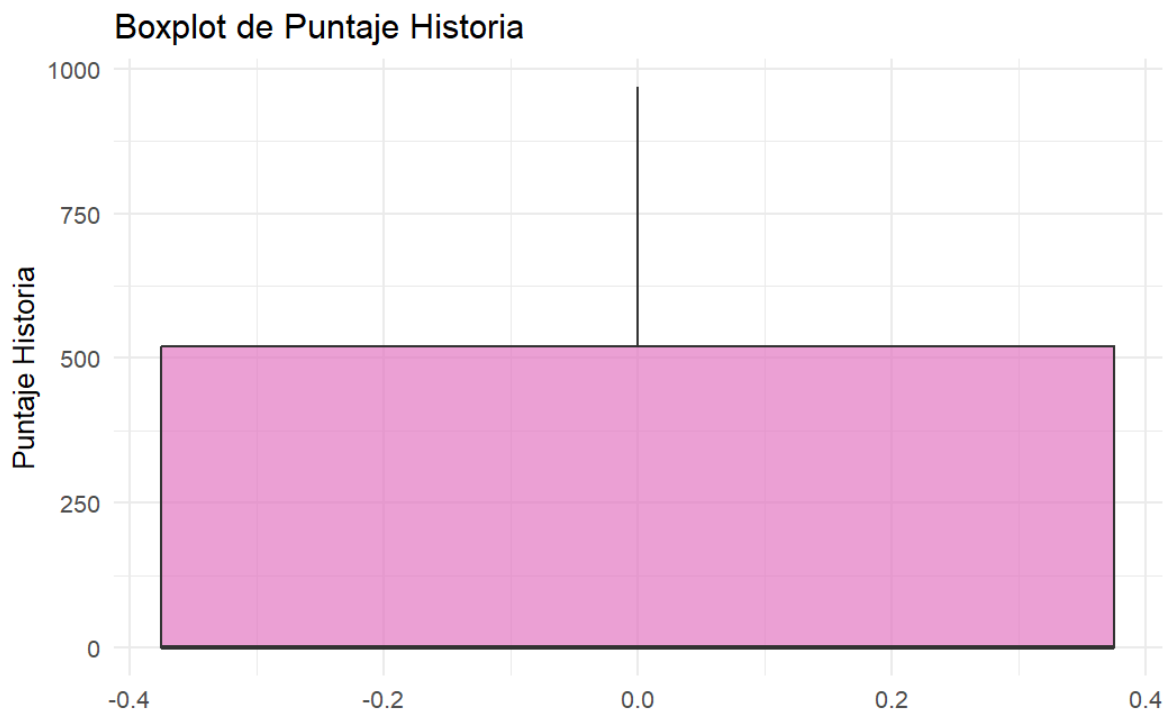


Figura 17: Boxplot de puntajes Historia

operacional concreto, lo que se les dificulta trabajar con abstracciones, y hay que ver cómo apoyarlos, en particular en asignaturas con alto nivel de abstracción.”

Estadística	Valor
Total de Datos (N)	810
Promedio	51.98
Desviación Estándar	20.3539
Varianza	414.2813

Tabla 8: Estadísticas descriptivas de los puntajes del test de Lawson

3.3.3. Test de Diagnóstico de Matemática

El **Test de Diagnóstico de Matemática** es una evaluación administrada por el Departamento de Matemática de la universidad. Su propósito es evaluar los **conocimientos matemáticos fundamentales** que los estudiantes poseen al inicio de su vida universitaria. Los resultados se consideran un indicador adicional de la base matemática de los participantes. Puede tomar valores de 0 a 18 puntos.

Estadística	Valor
Total de Datos (N)	2743
Promedio	10.14
Desviación Estándar	5.865104
Varianza	34.39944

Tabla 9: Estadísticas descriptivas de los puntajes de la prueba de diagnóstico de matemáticas

3.3.4. Test de Pensamiento Computacional

El **Test de Pensamiento Computacional** es una prueba que se aplica a todos los alumnos de primer año como parte del Plan de Inducción a la Vida Universitaria (PINVU). Consta de 12 preguntas distribuidas equitativamente en cuatro dimensiones clave del pensamiento computacional [Meza et al., 2024]:

- **Abstracción** (3 preguntas)
- **Pensamiento Algorítmico** (3 preguntas)
- **Descomposición** (3 preguntas)
- **Reconocimiento de Patrones y Generalización** (3 preguntas)

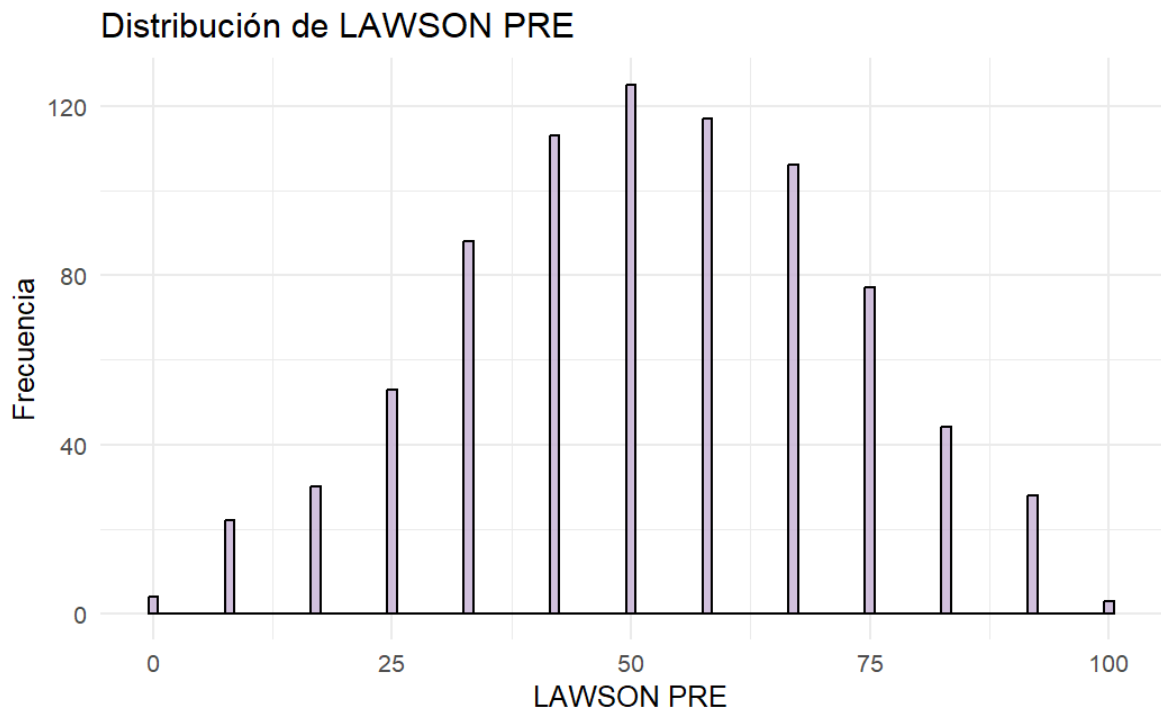


Figura 18: Distribución de puntajes Lawson

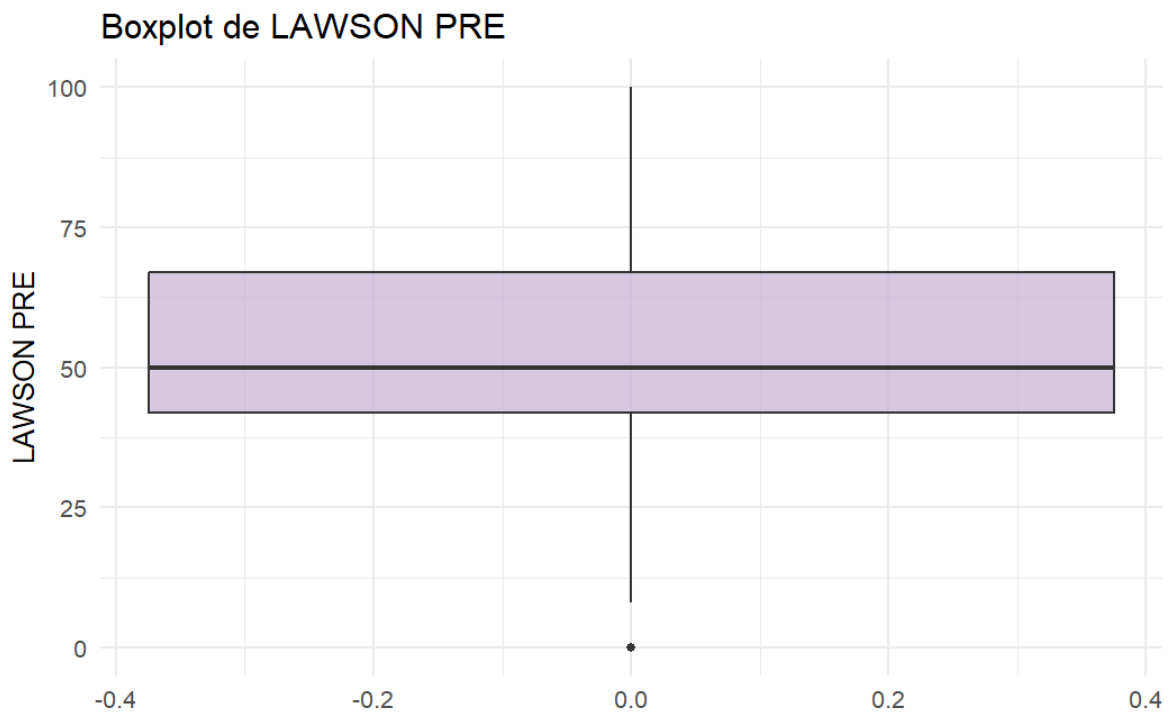


Figura 19: Boxplot de puntajes Lawson

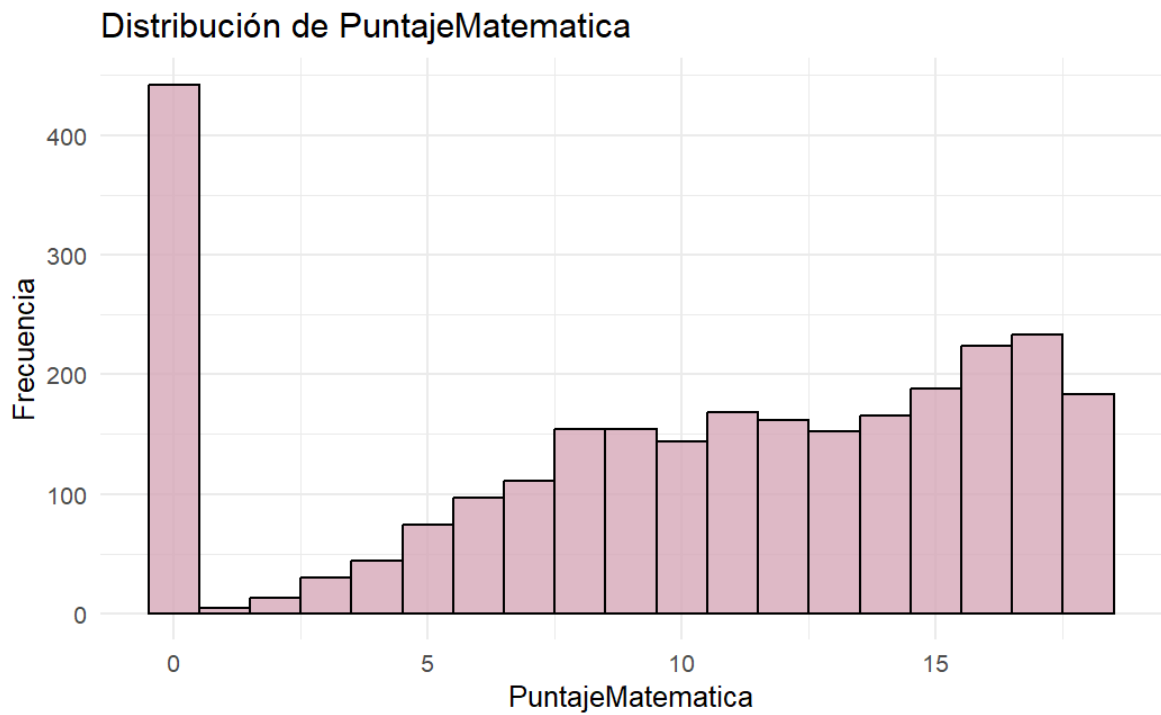


Figura 20: Distribución de puntajes Matemática

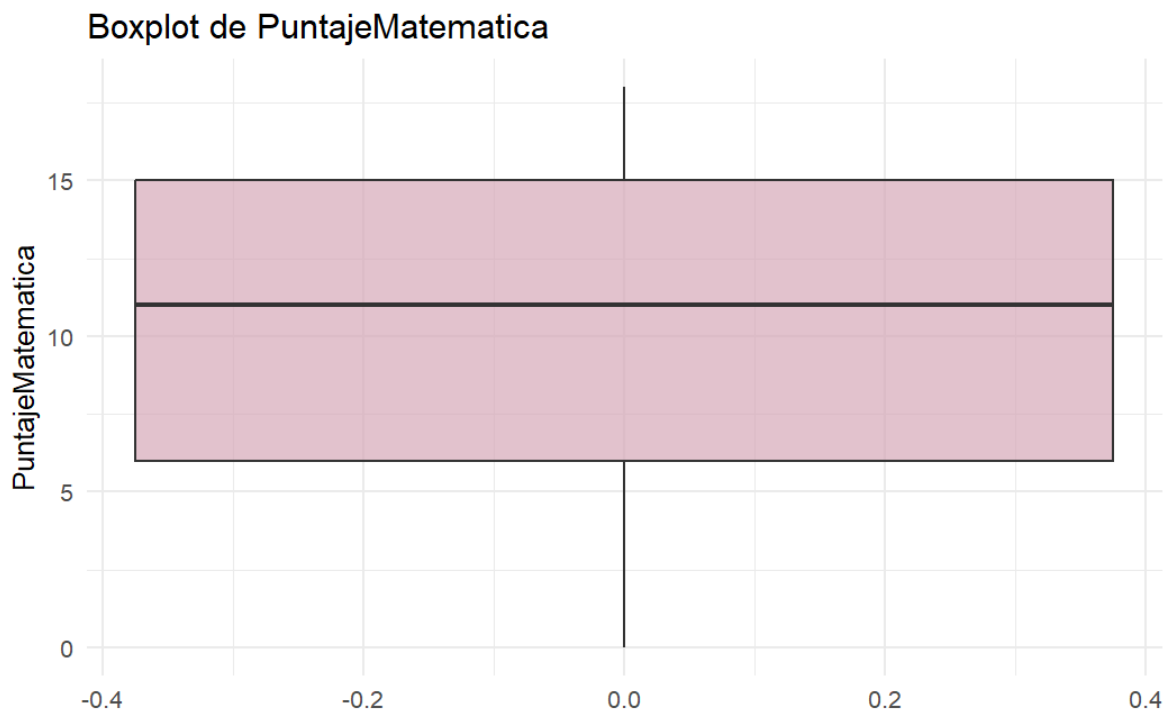


Figura 21: Boxplot de puntajes Matemática

La puntuación total y por cada dimensión se utiliza como indicador de las habilidades de pensamiento computacional de los estudiantes. Cada dimensión puede tomar un valor de 0 a 3 puntos y el total corresponde a la suma de todas las dimensiones, tomando valores de 0 a 12 puntos.

Categoría	Estadística	Valor
Total Pensamiento Computacional	Total de Datos (N)	2568
	Promedio	7.94 (66.17 %)
	Desviación Estándar	2.393897
	Varianza	5.730741
Abstracción	Total de Datos (N)	2568
	Promedio	1.91 (63.67 %)
	Desviación Estándar	0.9380999
	Varianza	0.8800314
Pensamiento Algorítmico	Total de Datos (N)	2568
	Promedio	1.83 (60.94 %)
	Desviación Estándar	0.8806085
	Varianza	0.7754714
Reconocimiento de Patrones y Generalización	Total de Datos (N)	2568
	Promedio	2.05 (68.33 %)
	Desviación Estándar	0.7884216
	Varianza	0.6216086
Descomposición	Total de Datos (N)	2568
	Promedio	2.16 (72.00 %)
	Desviación Estándar	0.8495043
	Varianza	0.7216576

Tabla 10: Estadísticas descriptivas consolidadas de Pensamiento Computacional y sus dimensiones

3.3.5. Encuestas de Factores Psicológicos

Para evaluar la motivación y la autoeficacia, se diseñaron dos encuestas específicas, basadas en escalas Likert de 5 puntos (Muy en desacuerdo, En desacuerdo, Ni de acuerdo ni en desacuerdo, De acuerdo, Muy de acuerdo). **Estas encuestas fueron adaptadas** de la “Academic Motivation Scale” y la “General Self-Efficacy Scale” con el fin de reducir la cantidad de preguntas. Es importante señalar que, debido a esta adaptación y a la naturaleza del estudio, **las encuestas no se encuentran validadas** en este contexto. Ambas encuestas fueron de carácter opcional.

Instrucciones Generales para las Encuestas: Lea cada afirmación cuidadosamente y marque en qué medida se identifica o está de acuerdo con cada una de ellas, basándose en su experiencia personal. *Escala de Respuesta:* *Muy en desacuerdo / En desacuerdo / Ni de acuerdo*

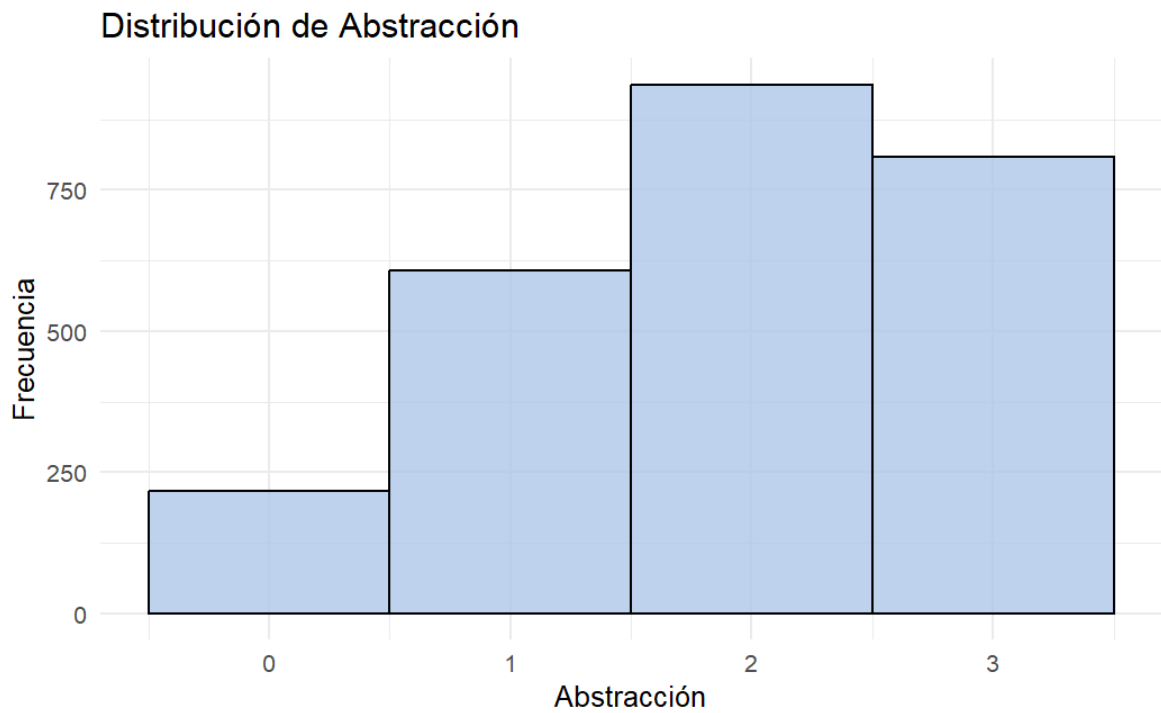


Figura 22: Distribución de puntajes Abstracción

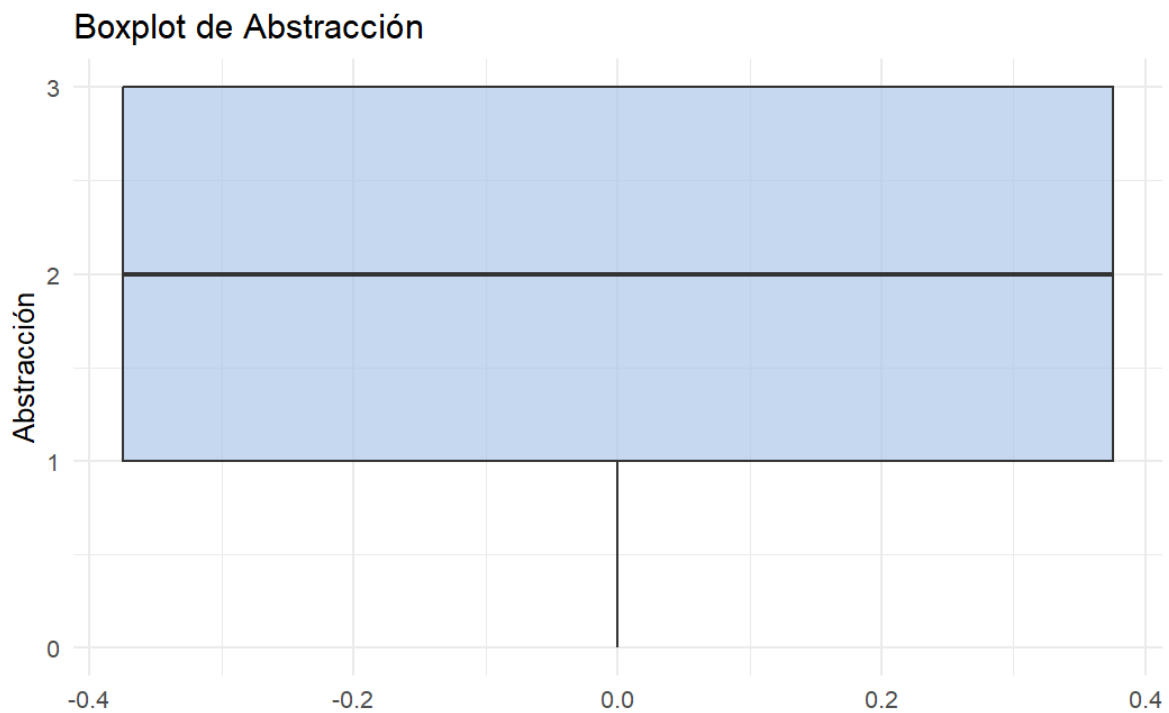


Figura 23: Boxplot de puntajes Abstracción

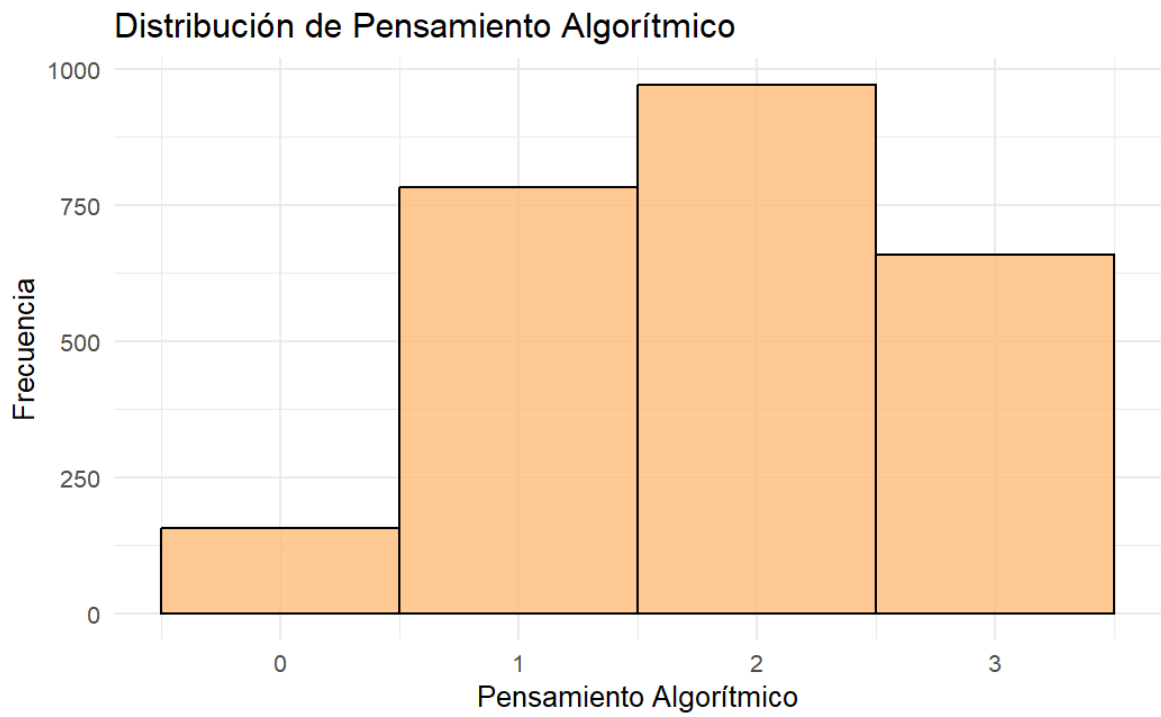


Figura 24: Distribución de puntajes Pensamiento Algorítmico

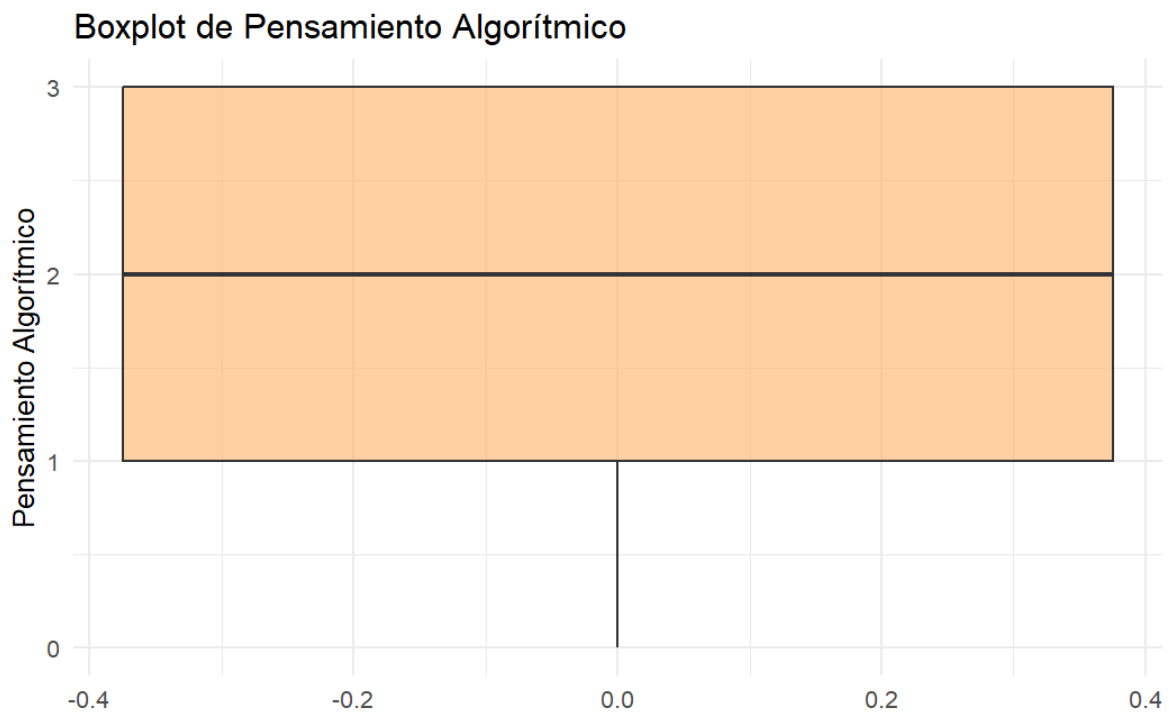


Figura 25: Boxplot de puntajes Pensamiento Algorítmico

A box plot showing the distribution of 'Reconocimiento de Patrones y Generalización' for the 'Modelo' variable. The y-axis ranges from 0 to 3. The box is green, spanning from approximately 2.0 to 3.0. The median is at 2.0. The whiskers extend from approximately -0.35 to 0.35. There is a single outlier at 0.0.

Statistic	Value (approx.)
Minimum (whisker)	-0.35
First Quartile (Q1)	2.0
Median	2.0
Third Quartile (Q3)	3.0
Maximum (whisker)	0.35
Outlier	0.0

Página 35 de 69

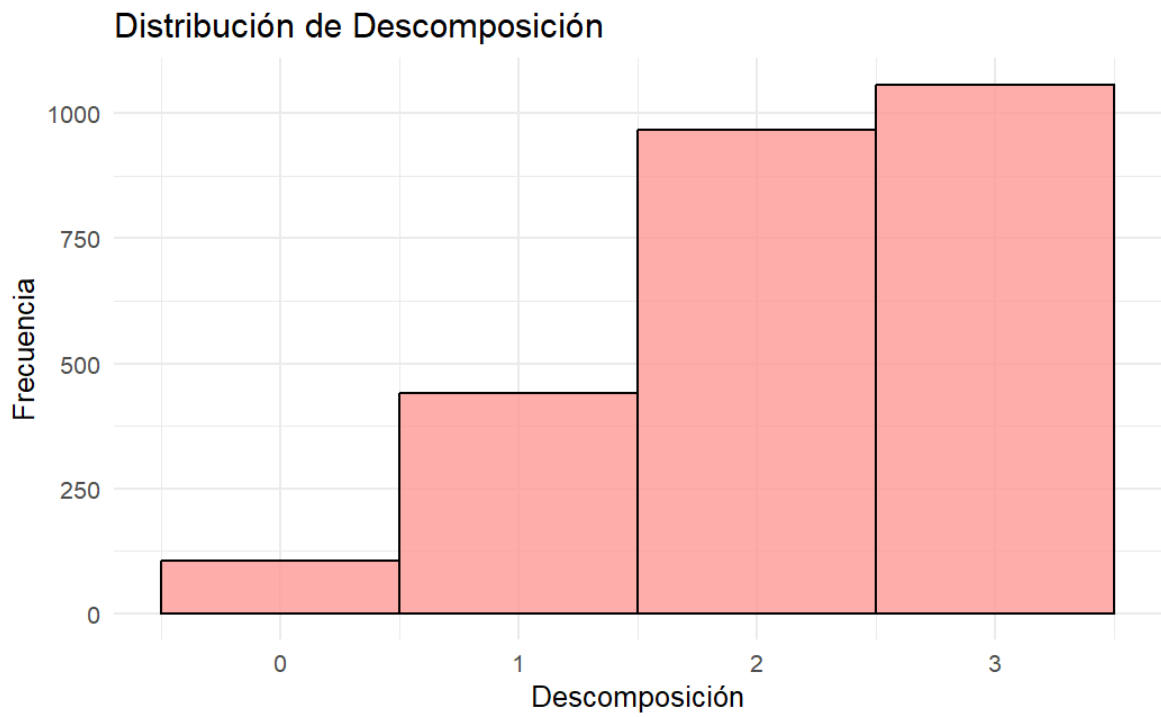


Figura 28: Distribución de puntajes Descomposición

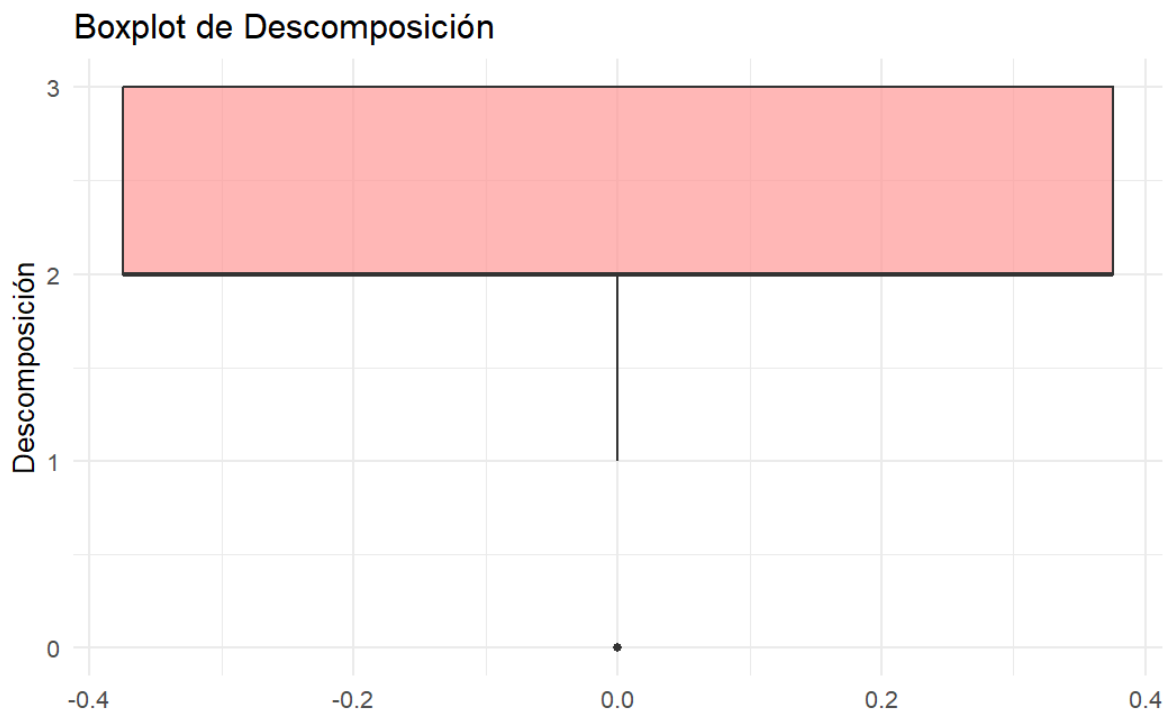


Figura 29: Boxplot de puntajes Descomposición

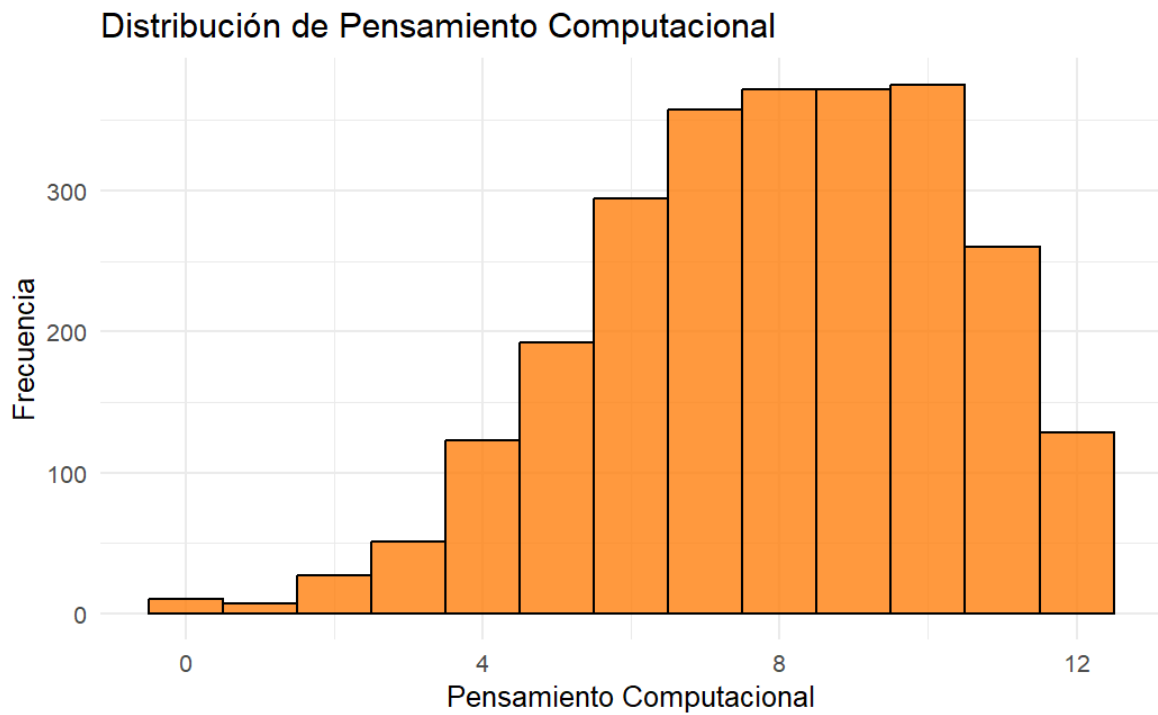


Figura 30: Distribución de puntajes Pensamiento Computacional

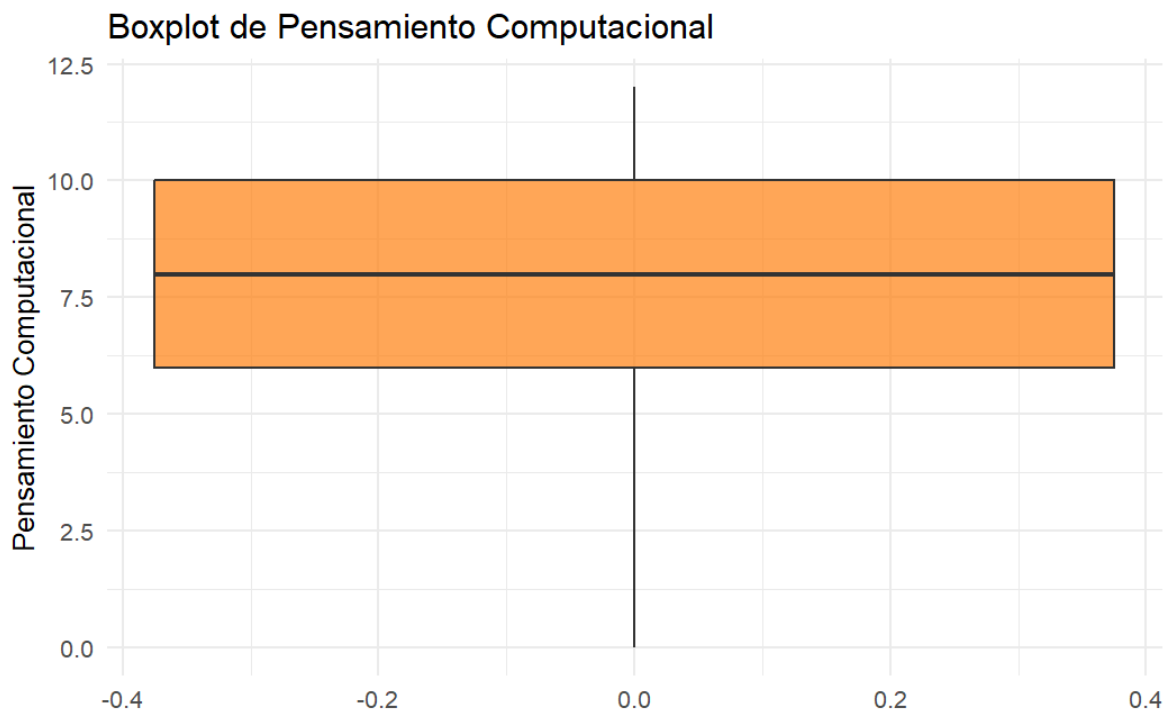


Figura 31: Boxplot de puntajes Pensamiento Computacional

ni en desacuerdo / De acuerdo / Muy de acuerdo

Encuesta de Motivación: Esta encuesta busca medir el nivel de motivación de los estudiantes en relación con su vida universitaria. Incluye 10 ítems que exploran diferentes aspectos de la motivación, como la motivación intrínseca (placer por aprender, superación personal, interés en temas) y extrínseca (recompensa económica, reconocimiento, “buena vida”).

Tabla 11: Ítems de la Encuesta de Motivación

Nº	Pregunta: ¿Por qué quiero ingresar a la universidad?
1	Para encontrar un trabajo bien remunerado más adelante.
2	Siento placer y satisfacción al aprender cosas nuevas.
3	Disfruto superarme a mí mismo/a en mis estudios.
4	Estudiar me permitirá obtener un trabajo de mayor reconocimiento.
5	Me permitirá trabajar en un área que me gusta.
6	Tener éxito en la universidad me hace sentir importante.
7	Estudiar me ayudará a tener “una buena vida” más adelante.
8	Siento satisfacción cuando logro realizar actividades académicas difíciles.
9	Me permite aprender sobre muchos temas que me interesan.
10	Quiero demostrarme a mí mismo/a que puedo tener éxito en mis estudios.

Estadística	Valor
Total de Datos (N)	341
Promedio	4.32
Desviación Estándar	0.5952565
Varianza	0.3543303

Tabla 12: Estadísticas descriptivas del promedio de motivación

Encuesta de Autoeficacia: Esta encuesta evalúa la percepción de los estudiantes sobre su capacidad para enfrentar y manejar diversas situaciones. Consta de 10 ítems diseñados para medir la confianza en las habilidades personales para abordar desafíos, persistir ante dificultades y alcanzar metas.

Tabla 13: Ítems de la Encuesta de Autoeficacia

N°	Pregunta
1	A pesar de los obstáculos yo puedo encontrar las maneras de obtener lo que quiero.
2	Yo enfrento problemas graves si me esfuerzo lo suficiente.
3	Es fácil mantenerme en mis metas hasta alcanzarlas.
4	Me siento seguro de poder enfrentar eficazmente situaciones inesperadas.
5	Gracias a mis cualidades personales yo puedo enfrentar situaciones inesperadas.
6	Yo puedo mantener la calma cuando estoy en problemas porque confío en mis habilidades para enfrentarlos.
7	Venga lo que venga, por lo general soy capaz de manejarlo.
8	Yo puedo resolver la mayoría de los problemas si me esfuerzo lo necesario.
9	En una situación difícil, generalmente se me ocurre lo que debo hacer.
10	Cuando me enfrento a un problema, generalmente encuentro varias soluciones.

Estadística	Valor
Total de Datos (N)	341
Promedio	3.95
Desviación Estándar	0.7390201
Varianza	0.5461508

Tabla 14: Estadísticas descriptivas del promedio de autoeficacia

3.3.6. Datos cualitativos

Carrera

La variable **Carrera** identifica el programa académico específico en el que está matriculado un estudiante. Este dato cualitativo es relevante para contextualizar el rendimiento en programación, ya que diferentes carreras pueden tener distintos niveles de exigencia o enfoque en habilidades computacionales.

Se filtraron por aquellos alumnos que rindieron el ramo de programación en el primer semestre del año 2025, debido a la gran cantidad de registros en el dataframe (68 carreras)

Campus

La variable **Campus** se refiere a la sede universitaria específica (e.g., Casa Central, Vitacura, San Joaquín) donde el estudiante cursa sus estudios. Este dato cualitativo puede ser un factor influyente, ya que los diferentes campus podrían tener variaciones en el perfil y cantidad de estudiantes, como se puede ver en la figura 33.

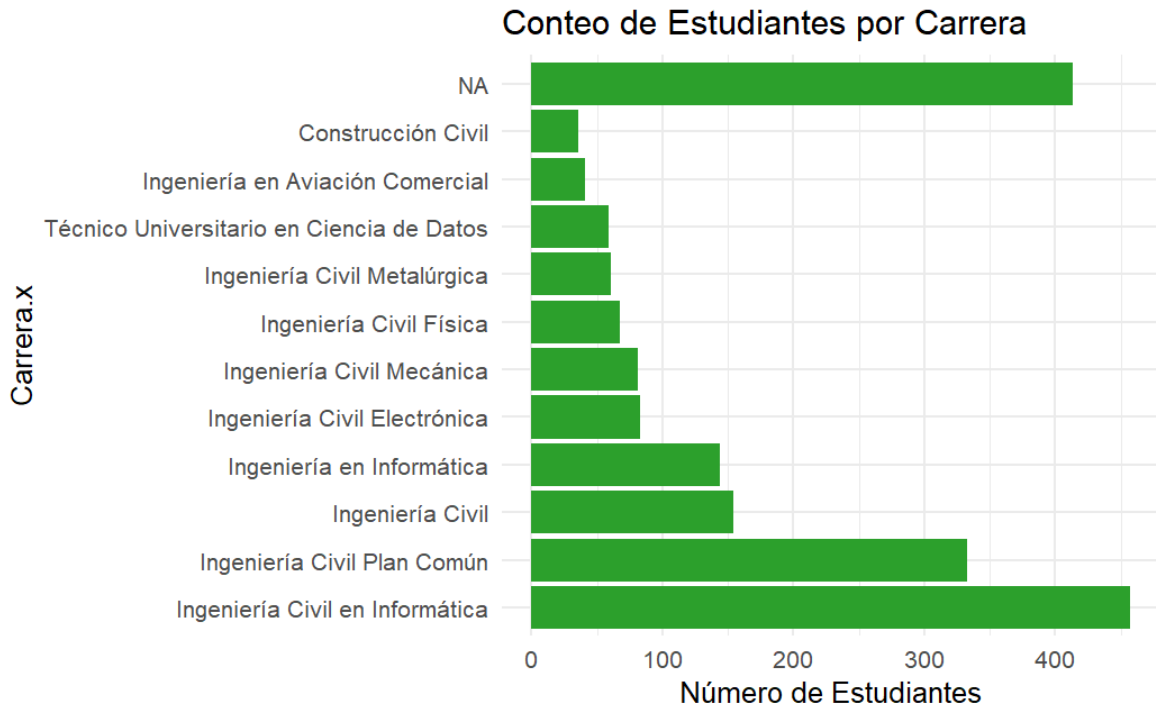


Figura 32: Cantidad de alumnos por carrera

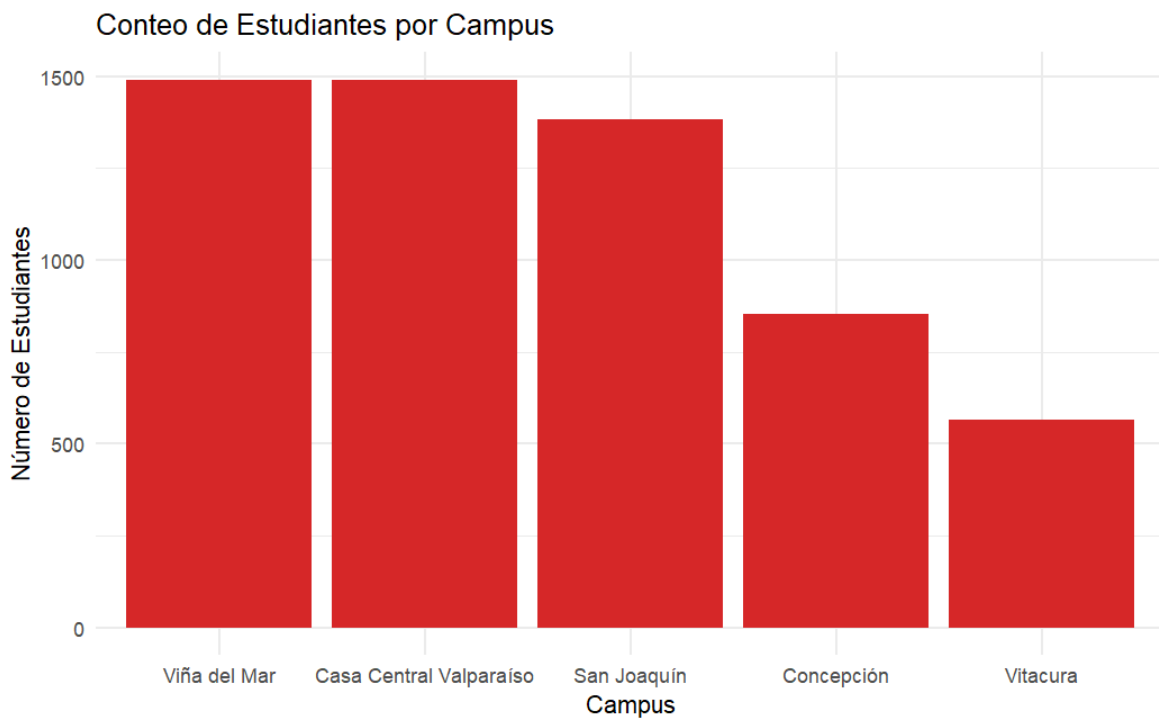


Figura 33: Cantidad de alumnos por campus

3.3.7. Datos de Rendimiento en Programación

Primer Certamen

El **Primer Certamen** es la nota obtenida por los estudiantes en la primera prueba escrita del curso introductorio de programación. Este puntaje se utiliza como el principal indicador del **rendimiento inicial** de los estudiantes en la materia, siendo crucial para el análisis de los factores que inciden en el aprendizaje de programación.

En la tabla 15, se presentan las estadísticas descriptivas para las notas de este certamen.

Estadística	Valor
Total de Datos (N)	1930
Promedio	47.28
Desviación Estándar	33.857
Varianza	1146.294

Tabla 15: Estadísticas descriptivas de las notas del primer certamen.

Segundo Certamen

El **Segundo Certamen** es la calificación obtenida por los estudiantes en la segunda prueba escrita del curso introductorio de programación. A diferencia del Primer Certamen, que se utiliza como el indicador principal de rendimiento inicial, la nota del Segundo Certamen cumple una función crucial de **validación**.

Este puntaje nos permite validar las predicciones de los modelos desarrollados. Al comparar los resultados predichos por el modelo con los puntajes reales del Segundo Certamen, podemos evaluar la precisión y la robustez del modelo para generalizar su capacidad predictiva más allá del conjunto de datos inicial utilizado para su entrenamiento. En esencia, actúa para confirmar si los factores identificados realmente inciden en el rendimiento continuado del estudiante en el curso.

Estadística	Valor
Total de Datos (N)	1510
Promedio	50.96
Desviación Estándar	34.15346
Varianza	1166.459

Tabla 16: Estadísticas descriptivas de las notas del Certamen 2

Nota Final

La **Nota Final** es el promedio ponderado de todas las evaluaciones del curso, incluyendo certámenes, laboratorios, tareas y cualquier otra calificación que componga el 100 % de la nota de aprobación del curso.

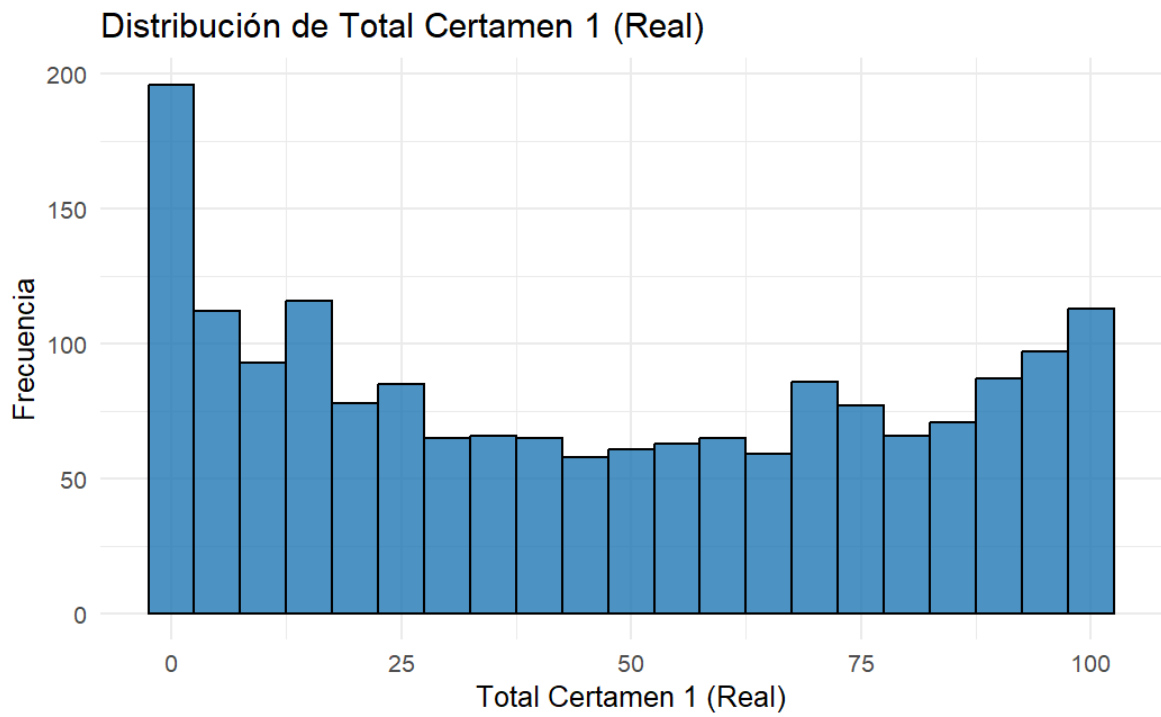


Figura 34: Distribución de notas Certamen 1

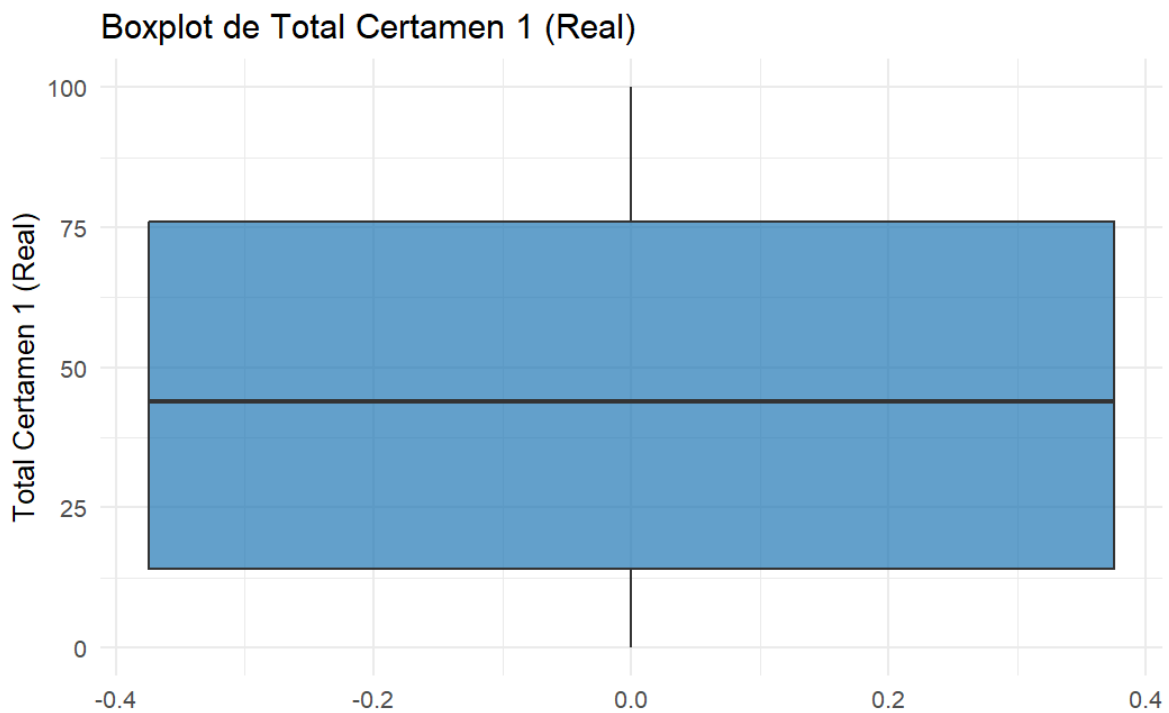


Figura 35: Boxplot de notas Certamen 1

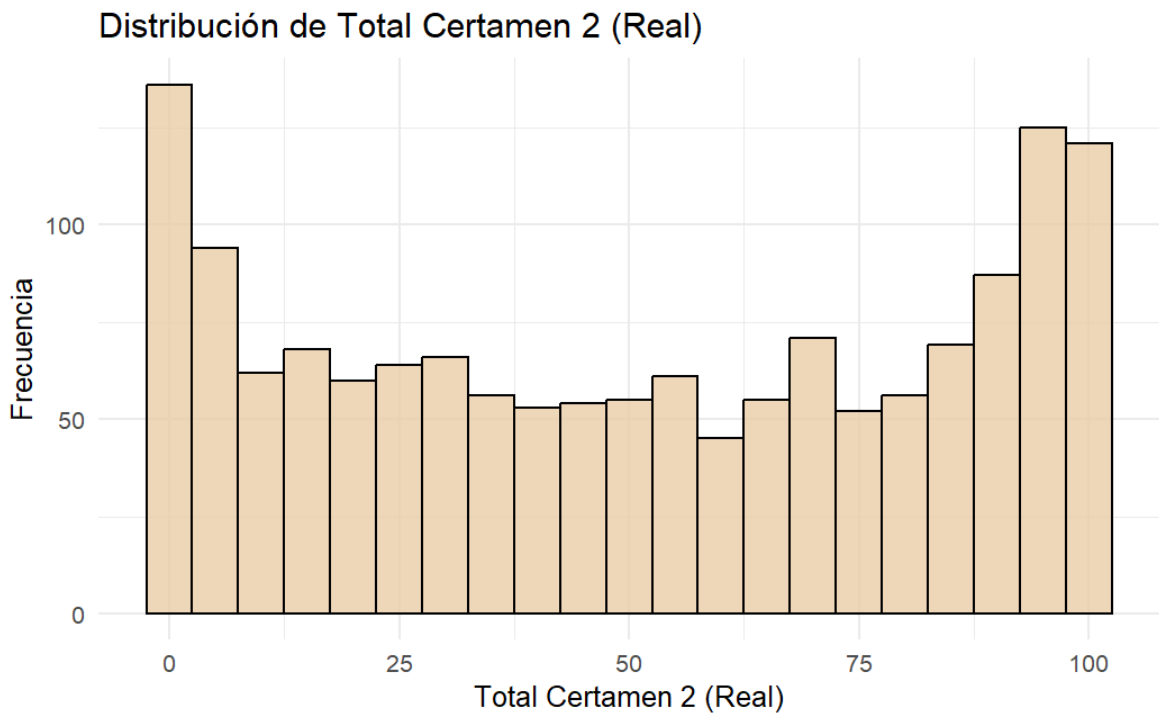


Figura 36: Distribución de notas Certamen 2

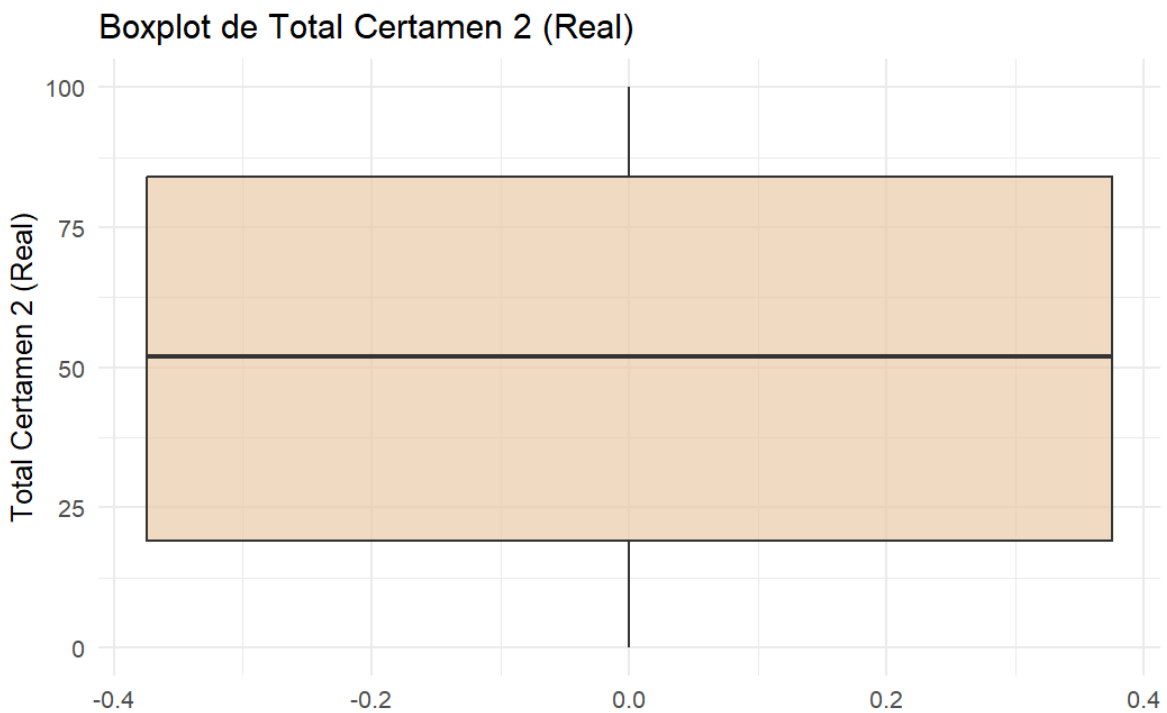


Figura 37: Boxplot de notas Certamen 2

Esta nota representa el **rendimiento global y definitivo** del estudiante en la asignatura de programación. Se utiliza como un indicador crucial para evaluar si los factores analizados y los modelos predictivos son capaces de predecir no solo el desempeño en evaluaciones parciales como el Certamen 1, sino también el **éxito académico a largo plazo** en el curso. Al considerar la Nota Final, podemos validar de manera más exhaustiva si el Certamen 1 es un buen predictor del rendimiento general en el curso, proporcionando una perspectiva completa sobre el impacto de las variables estudiadas en el aprendizaje de programación.

Estadística	Valor
Total de Datos (N)	1887
Promedio	54.85
Desviación Estándar	27.47432
Varianza	754.838

Tabla 17: Estadísticas descriptivas de las Notas Finales del curso

3.4. Transformación de Datos

Una vez recolectados, los datos fueron sometidos a un riguroso proceso de limpieza y transformación utilizando el software **R (versión 4.5.0)** y el entorno de desarrollo integrado **RStudio (versión 2025.05.1)**. Los pasos clave incluyeron:

- **Selección de Última Respuesta Válida:** En las encuestas de motivación y autoeficacia, cuando existían múltiples respuestas por estudiante, se seleccionó la más reciente utilizando el valor máximo del campo Respuesta.
- **Limpieza y Unificación de Datos por Campus:** Las bases de datos de matemática provenientes de los tres campus (Casa Central, Vitacura y San Joaquín) fueron limpiadas y estandarizadas individualmente, incluyendo el puntaje del estudiante. Posteriormente, fueron combinadas en una sola tabla mediante `bind_rows()`.
- **Unión de Bases de Datos:** Se realizó una unión (`left_join`) de las diferentes fuentes utilizando el identificador como clave, integrando las variables de interés en un único **dataframe base** (`df_base`).
- **Cálculo de Promedios:** Para las respuestas a las encuestas de motivación y autoeficacia, se calcularon los promedios por estudiante (`motivacion_prom` y `autoeficacia_prom`) utilizando `rowMeans()`, omitiendo valores faltantes.
- **Conversión de Tipos de Datos:** Se convirtieron variables relevantes como las notas de certámenes al tipo numérico (`as.numeric`) para permitir análisis estadísticos posteriores.

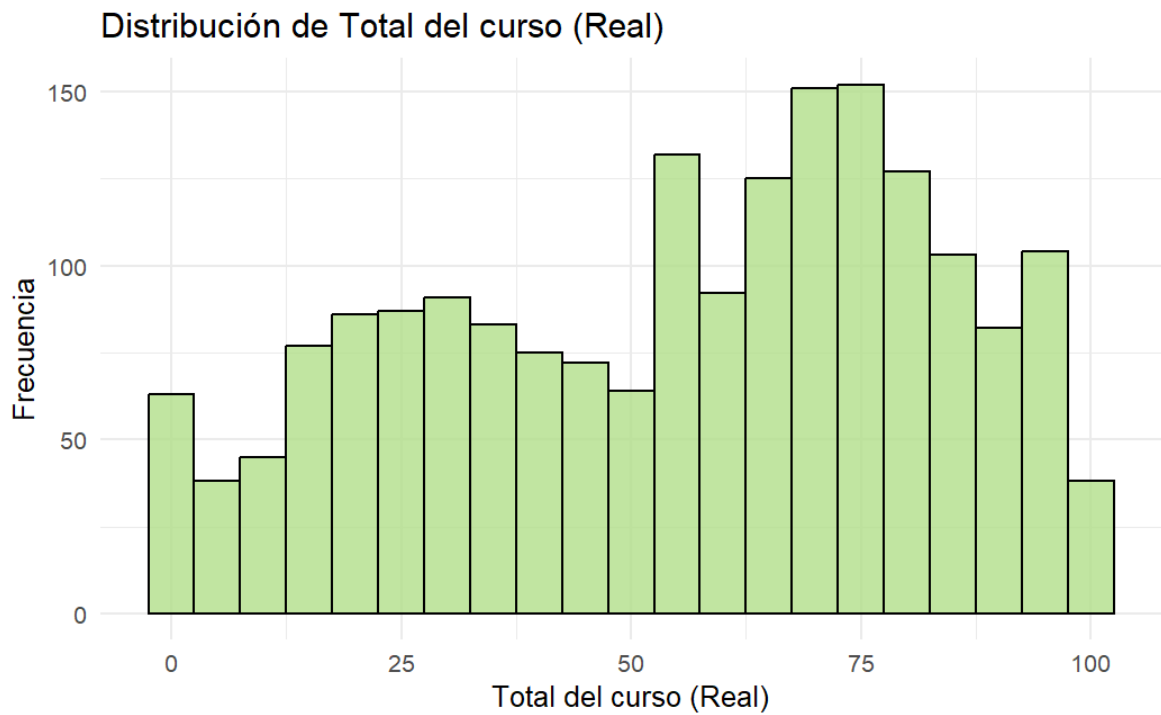


Figura 38: Distribución de notas finales

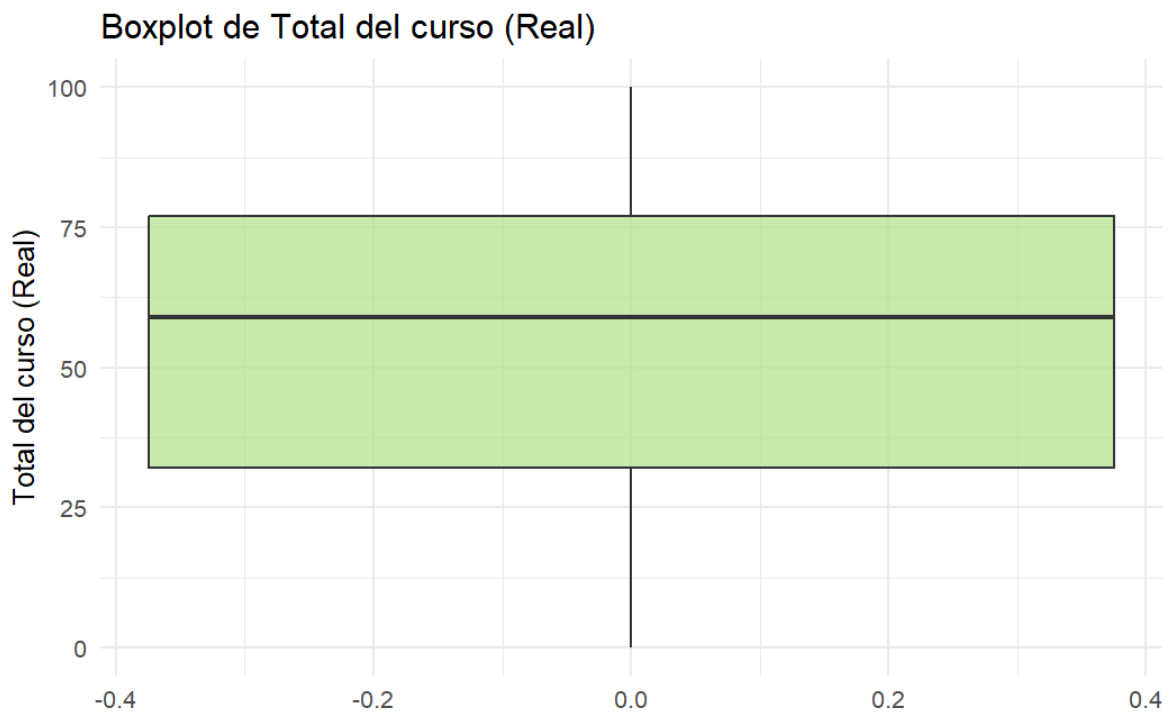


Figura 39: Boxplot de notas Finales

- **Incorporación de Certamen 2:** Se integraron los datos del Certamen 2 mediante una unión adicional con la base principal, filtrando previamente los registros válidos.
- **Incorporación de Nota Final:** Finalmente se integraron los datos de la nota final del curso mediante una unión adicional con la base principal, filtrando previamente los registros válidos.

3.5. Análisis de Datos

El análisis de los datos se llevó a cabo en R, empleando diversas técnicas estadísticas para explorar las relaciones entre las variables y construir modelos predictivos.

3.5.1. Estadísticos Descriptivos

Se calcularon estadísticos descriptivos (media, desviación estándar, varianza, y resúmenes de cinco números) para las variables clave, con el fin de comprender la distribución y variabilidad de los datos. Se utilizaron **diagramas de caja (boxplots)** e **histogramas** para visualizar la distribución de las variables.

3.5.2. Análisis de Correlaciones

Para evaluar las relaciones entre las variables predictoras y el rendimiento en el Certamen 1, se emplearon dos enfoques estadísticos de correlación descritos en Restrepo y González (2007) [Restrepo y González, 2007]:

- **Coefficiente de Pearson (r):** mide la fuerza y dirección de la relación lineal entre dos variables cuantitativas continuas. Este coeficiente se basa en la covarianza entre las dos variables estandarizada por sus desviaciones estándar, y su valor se encuentra entre -1 y $+1$. Pearson es sensible a la presencia de valores atípicos y asume que las variables presentan una distribución normal conjunta [Restrepo y González, 2007].
- **Coefficiente de Spearman (ρ):** es un estadístico no paramétrico basado en rangos, apropiado para variables ordinales o cuando las condiciones de normalidad no se cumplen. Spearman también varía entre -1 y $+1$, y mide la fuerza de una relación monótona (no necesariamente lineal) entre dos variables. Puede usarse como alternativa robusta a Pearson cuando los datos son asimétricos, presentan valores extremos, o cuando se trabaja con escalas de tipo Likert, como es el caso de las variables de motivación y autoeficacia [Restrepo y González, 2007].

Ambos coeficientes permiten evaluar si existe asociación entre dos variables, y se interpretan de forma similar en cuanto a su magnitud y signo.

La magnitud de los coeficientes se interpretó según los siguientes rangos de referencia:

Rango	Interpretación
$ r < 0,3$	Relación débil
$0,3 \leq r < 0,7$	Relación moderada
$0,7 \leq r \leq 1,0$	Relación fuerte

Tabla 18: Interpretación de la magnitud del coeficiente de correlación.

Este análisis fue clave para seleccionar las variables más relevantes para los modelos predictivos, diferenciando entre relaciones verdaderamente significativas y asociaciones espurias o débiles.

3.5.3. Modelado de Regresión Lineal Múltiple

Se desarrollaron **tres modelos de regresión lineal múltiple** para predecir el rendimiento en el Certamen 1 de Programación ('Total Certamen 1 (Real)') a partir de diferentes conjuntos de variables predictoras. Estos modelos fueron diseñados con el objetivo de evaluar la capacidad explicativa de distintos factores académicos y cognitivos sobre el desempeño inicial en programación.

- **Modelo 1:** Incluyó como predictores los puntajes de ingreso en Matemática M1 y Matemática M2. Este modelo se construyó priorizando la representatividad de la muestra, ya que estas variables presentaban baja cantidad de valores faltantes y alta correlación con la nota del Certamen 1.
- **Modelo 2:** Amplía el Modelo 1 incorporando los puntajes de Lenguaje y NEM, con el fin de capturar un panorama más amplio de las habilidades académicas previas de los estudiantes. Este modelo busca analizar si la adición de estas variables mejora el poder predictivo sin sacrificar significativamente el tamaño de la muestra.
- **Modelo 3:** Es el modelo más completo y conceptualmente robusto. Incluye todas las variables anteriores y agrega las dimensiones del pensamiento computacional ('Abstracción', 'Pensamiento Algorítmico', 'Descomposición' y 'Reconocimiento de Patrones y Generalización'), seleccionadas por su alta correlación teórica y empírica con el rendimiento en programación. Este modelo privilegia la riqueza del contenido explicativo, aunque implica una reducción en el número de observaciones válidas debido a datos faltantes en las encuestas.

Indicadores utilizados para la evaluación de los modelos:

- **R cuadrado ajustado (R_{adj}^2):** Mide la proporción de la varianza en la nota del Certamen 1 explicada por las variables del modelo, ajustado por el número de predictores. A mayor valor, mayor capacidad explicativa. El ajuste evita la sobrestimación del modelo al incluir más variables.
- **Error Estándar Residual (RSE):** Representa la desviación típica de los errores (residuos) del modelo. Cuanto menor sea el RSE, mayor es la precisión promedio de las predicciones en las unidades de la variable dependiente (nota del certamen).
- **Estadístico F:** Evalúa si el modelo, en su conjunto, explica una cantidad significativa de la variabilidad en la variable dependiente. Un valor alto de F y un p-valor asociado muy bajo indican que al menos una de las variables predictoras tiene un efecto significativo.
- **Valor-p del estadístico F:** Si el valor-p es menor que un umbral típico ($\alpha = 0,05$), se concluye que el modelo tiene capacidad predictiva global significativa.
- **Observaciones válidas:** Cantidad de registros disponibles con datos completos para todas las variables del modelo. Es relevante para evaluar la robustez de los resultados y la generalización.

3.5.4. Análisis de Interacción Motivación x Autoeficacia

Se ajustó un modelo de regresión lineal para explorar la **interacción entre motivación y autoeficacia** en la predicción del rendimiento. Este modelo incluyó los efectos principales de 'motivacion_prom' y 'autoeficacia_prom', así como su término de interacción ('motivacion_prom * autoeficacia_prom'). Se evaluó la significancia estadística de cada coeficiente y la bondad de ajuste global del modelo.

3.5.5. Análisis por Categoría

Finalmente, se realizó un **análisis descriptivo por carrera y por campus** para el promedio de la nota del Certamen 1 de Programación. Se calcularon el número de estudiantes, el promedio y la desviación estándar para cada categoría, y se visualizaron los resultados mediante **diagramas de caja**. Este análisis permite identificar posibles diferencias en el rendimiento en programación entre distintas poblaciones estudiantiles.

3.6. Validación del Modelo de Predicción

Para validar la capacidad predictiva del **Modelo 3 (el modelo más completo)**, el cual fue entrenado para predecir el rendimiento en el **Primer Certamen**, se utilizaron las notas reales

del **Segundo Certamen**. Este enfoque permite evaluar la **robustez** y **generalización** del modelo en un contexto de evaluación diferente pero relacionado dentro del mismo curso de programación.

3.6.1. Realización de Predicciones y Análisis de Errores

Se aplicó el modelo entrenado (**Modelo 3**) a los datos del Certamen 2 para generar predicciones. Posteriormente, se calculó el **error de predicción** como la diferencia entre la nota predicha y la nota real del Certamen 2, utilizando la fórmula $Error = Prediccin - Real$. Este cálculo es fundamental para identificar el **sesgo** del modelo: un error positivo indica que el modelo tiende a **sobreestimar** las notas, mientras que un error negativo sugiere que las **subestima**. Complementariamente, se calculó la **diferencia absoluta**, definida como $|Prediccin - Real|$. Esta métrica es crucial para evaluar la **magnitud del desajuste** del modelo, independientemente de si la predicción fue mayor o menor que la nota real. La diferencia absoluta es un componente esencial para el cálculo de métricas de precisión agregadas, como el **Error Cuadrático Medio (MSE)** y la **Raíz del Error Cuadrático Medio (RMSE)**, las cuales proporcionan una medida global de la exactitud del modelo.

3.7. Análisis de Correlación entre C1 y Nota Final

Con el objetivo de evaluar si el **rendimiento en el Certamen 1** constituye un buen indicador del **desempeño final en el curso**, se replicó el análisis de correlación realizado previamente entre variables académicas y el Certamen 1, pero esta vez utilizando como variable dependiente la **Nota Final del curso de Programación**.

Se calcularon los coeficientes de correlación de Pearson entre la nota del Certamen 1 y la Nota Final, así como con el resto de las variables predictoras previamente consideradas (puntajes de ingreso y dimensiones del pensamiento computacional). Este análisis permite determinar la **fuerza y dirección de la relación lineal** entre el desempeño inicial del estudiante y su rendimiento final en la asignatura.

Adicionalmente, se examinó si la nota del Certamen 1 puede servir como un **predictor temprano de éxito académico** en programación. Una correlación alta y positiva entre ambas notas sugeriría que el Certamen 1 refleja de manera efectiva las habilidades necesarias para tener éxito en el curso, y por lo tanto, puede ser utilizado como una herramienta para **identificar estudiantes en riesgo** y aplicar medidas de apoyo oportunas.

Este análisis proporciona una base empírica para respaldar o cuestionar el uso del Certamen 1 como criterio de predicción, y permite contextualizar su relevancia dentro de los modelos predictivos más amplios desarrollados en esta investigación.

CAPÍTULO 4

Análisis y Resultados

En este capítulo se presentan los resultados y análisis derivados del estudio, con énfasis en los factores que influyen en el rendimiento académico de los estudiantes en los cursos **IWI-131: Programación** e **INF-129: Introducción a la Programación**. Se validan los hallazgos mediante análisis estadístico, exploración visual, y la construcción de modelos predictivos.

4.1. Factores Asociados al Rendimiento en el Certamen 1

Se generó un **gráfico de barras horizontal** para visualizar las 12 variables con mayor correlación con el rendimiento en el Certamen 1, agrupándolas temáticamente para facilitar la interpretación.

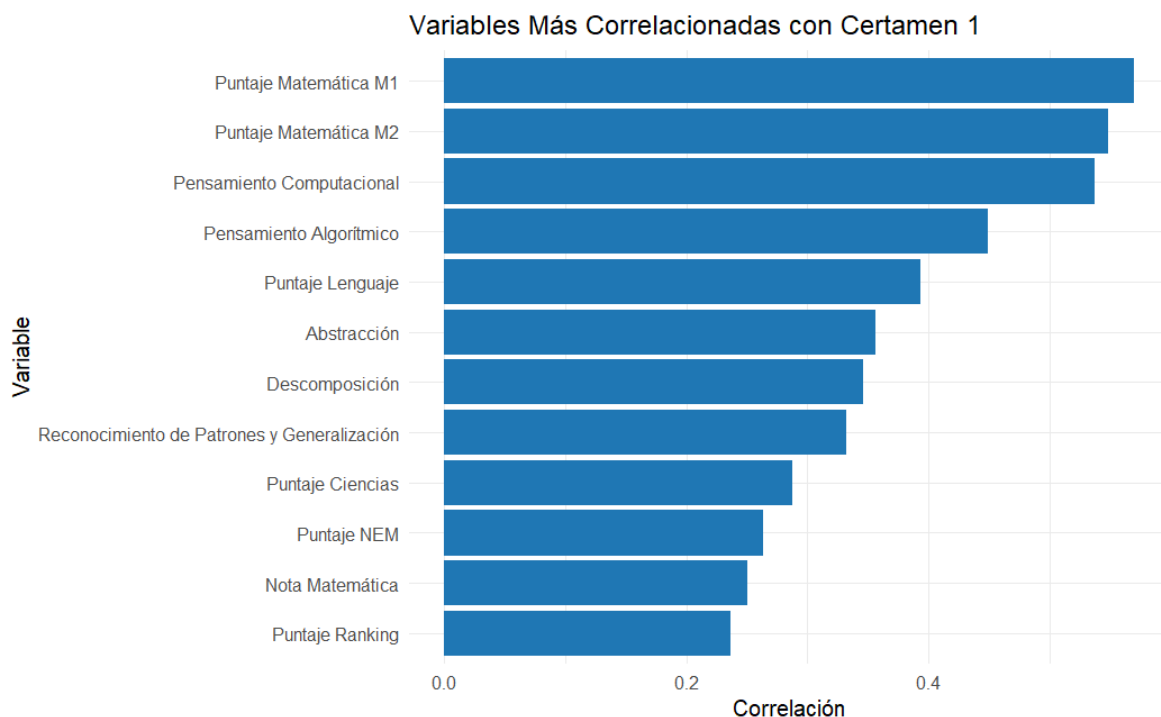


Figura 40: Variables más correlacionadas con certamen 1

El análisis de correlación mostró que las variables más asociadas al rendimiento en el Certamen 1 se agrupan en dos categorías principales: **habilidades matemáticas y académicas previas** y **habilidades de pensamiento computacional**.

- **Habilidades matemáticas y académicas previas:** Las variables como el Puntaje Matemática M1, Puntaje Matemática M2, Puntaje NEM, Puntaje Lenguaje, y la Nota Matemática mostraron correlaciones positivas y significativas con la nota del certamen. Esto es consistente con literatura previa [Owolabi *et al.*, 2014], que resalta la relación entre habilidades lógico-matemáticas y programación.
- **Pensamiento computacional:** Las subdimensiones de Pensamiento Algorítmico, Abstracción, Descomposición y Reconocimiento de Patrones también mostraron correlaciones importantes. En especial, Abstracción y Pensamiento Algorítmico presentaron significancia en los modelos predictivos. Estos resultados coinciden con estudios como [Kramer, 2007], que destacan la importancia de habilidades cognitivas específicas para el aprendizaje de programación.

En cambio, las variables psicológicas de **motivación** y **autoeficacia** no presentaron una correlación fuerte con el rendimiento en el Certamen 1, esto se puede deber a la naturaleza opcional de las encuestas, por lo que pudo haber un sesgo sobre quiénes decidieron responder. La Tabla 19 resume sus estadísticos descriptivos.

Tabla 19: Estadísticos descriptivos de motivación y autoeficacia promedio.

Estadístico	Motivación	Autoeficacia
Mínimo	1.000	1.000
1er Cuartil	4.100	3.500
Mediana	4.400	4.000
Media	4.334	3.952
3er Cuartil	4.800	4.500
Máximo	5.000	5.000
NA's	1699	1699
Varianza	0.343	0.555
Desviación Estándar	0.586	0.745

El modelo de regresión con interacción entre motivación y autoeficacia no resultó significativo (R^2 ajustado = $-0,0056$, $p = 0.8113$). Esto sugiere que, dentro del contexto del curso y los datos disponibles, estas variables no son predictoras directas del rendimiento.

4.2. Comparación de Modelos Predictivos

Se evaluaron tres modelos de regresión lineal múltiple:

- **Modelo 1:** Usó solo Puntaje Matemática M1 y M2.
- **Modelo 2:** Agregó Puntaje Lenguaje y NEM.

Tabla 20: Comparación de modelos predictivos del rendimiento en el Certamen 1.

Modelo	R ² Ajustado	RSE	Observaciones	Estadístico F	p-valor
Modelo 1	0.3468	27.98	1345	358.00	$< 2,2 \times 10^{-16}$
Modelo 2	0.3568	27.76	1345	187.60	$< 2,2 \times 10^{-16}$
Modelo 3	0.3971	26.20	972	80.95	$< 2,2 \times 10^{-16}$

- **Modelo 3:** Incorporó además subdimensiones del pensamiento computacional.

El Modelo 3 presentó el **mejor ajuste**, explicando el **39.71 %** de la varianza en la nota del certamen (un **R² ajustado** de 0.3971), con una **significancia estadística alta** ($F = 80,95$, $p < 2,2 \times 10^{-16}$). Aunque este modelo tuvo un **error estándar residual (RSE) de 26.20**, fue el más bajo de los tres modelos, indicando la mayor precisión. Sin embargo, se construyó con **972 observaciones**, lo que representa una cantidad menor de datos en comparación con los otros dos modelos debido a la disponibilidad de todas las variables incluidas.

4.3. Análisis del Modelo con Interacción: Motivación x Autoeficacia

Se construyó un modelo de regresión lineal para evaluar si existe una **interacción significativa entre la motivación y la autoeficacia** en la predicción del rendimiento en el **Primer Certamen de Programación**.

4.3.1. Evaluación Global del Modelo

Los resultados del modelo, detallados en la Tabla 22, indican lo siguiente:

- **R² Ajustado: -0.007311** — Este valor, cercano a cero e incluso negativo, sugiere que el modelo **no explica prácticamente nada** de la variabilidad observada en el rendimiento del Certamen 1. Un R^2 ajustado negativo indica que el modelo es peor que un modelo sin predictores (es decir, peor que simplemente usar el promedio de la variable dependiente).
- **R² Múltiple: 0.00171** — Solo un **0.17 %** de la varianza total de las notas del Certamen 1 es explicada por este modelo.
- **Estadístico F: 0.1895 (p-valor = 0.9035)** — El **p-valor** asociado al estadístico F (0.9035) es muy superior al nivel de significancia común (e.g., 0.05). Esto significa que el modelo en su conjunto **no es estadísticamente significativo**, lo que impide rechazar la hipótesis nula de que todos los coeficientes de los predictores son cero. En otras palabras, el modelo **no tiene valor predictivo útil**.

- **Error Estándar Residual (RSE): 34.32** — Este valor es relativamente alto en una escala de 0 a 100, indicando una dispersión considerable de los residuos y, por ende, una baja precisión en las predicciones individuales. Se basa en 332 grados de libertad, habiéndose excluido 1443 observaciones debido a valores faltantes en las variables de motivación y autoeficacia.

4.3.2. Interpretación de Coeficientes

La Tabla 21 muestra los coeficientes individuales:

- **motivacion_prom (p-valor = 0.980)**: El efecto de la motivación promedio no es estadísticamente significativo.
- **autoeficacia_prom (p-valor = 0.679)**: El efecto de la autoeficacia promedio tampoco es estadísticamente significativo.
- **motivacion_prom:autoeficacia_prom (p-valor = 0.746)**: El término de interacción entre motivación y autoeficacia no es estadísticamente significativo.

Los resultados de este análisis indican claramente que **ni la motivación promedio, ni la autoeficacia promedio, ni la interacción entre ambas tienen un efecto estadísticamente significativo** en la predicción del rendimiento en el Primer Certamen de Programación. El modelo es **no predictivo** y su desempeño es considerablemente peor en comparación con los modelos que incorporan variables académicas y de pensamiento computacional previo. Esto sugiere que estos factores psicológicos, **tal como fueron medidos en este estudio**, no actúan como predictores directos del rendimiento inicial en programación. La baja cantidad de observaciones utilizadas en este modelo (336) debido a la naturaleza opcional de las encuestas también podría haber afectado la capacidad de detectar efectos, incluso si existieran.

Tabla 21: Coeficientes del Modelo de Regresión Lineal: Interacción entre Motivación y Autoeficacia

Término	Estimación	Error Estándar	Valor t	p-valor
(Intercepto)	47.6113	36.5116	1.304	0.193
motivacion_prom	-0.2262	8.9463	-0.025	0.980
autoeficacia_prom	4.8892	11.8060	0.414	0.679
motivacion_prom:autoeficacia_prom	-0.8425	2.6024	-0.324	0.746

4.4. Análisis por Carrera y Campus

Se observó una variabilidad importante en el rendimiento según la carrera y el campus de los estudiantes:

Tabla 22: Métricas de Ajuste del Modelo de Regresión

Métrica del Modelo	Valor
R^2 Múltiple	0.00171
R^2 Ajustado	-0.00731
Error Estándar Residual (RSE)	34.32 (gl = 332)
Estadístico F	0.1895 (gl = 3, 332)
p-valor del estadístico F	0.9035
Observaciones utilizadas	336

- **Carreras con mejor rendimiento promedio:** Ingeniería Civil en Informática, Ingeniería Civil Electrónica.
- **Carreras con menor rendimiento promedio:** Ingeniería Civil Mecánica, Ingeniería Civil.
- **Campus con mayor dispersión:** San Joaquín presentó mayor variabilidad en notas de certamen.

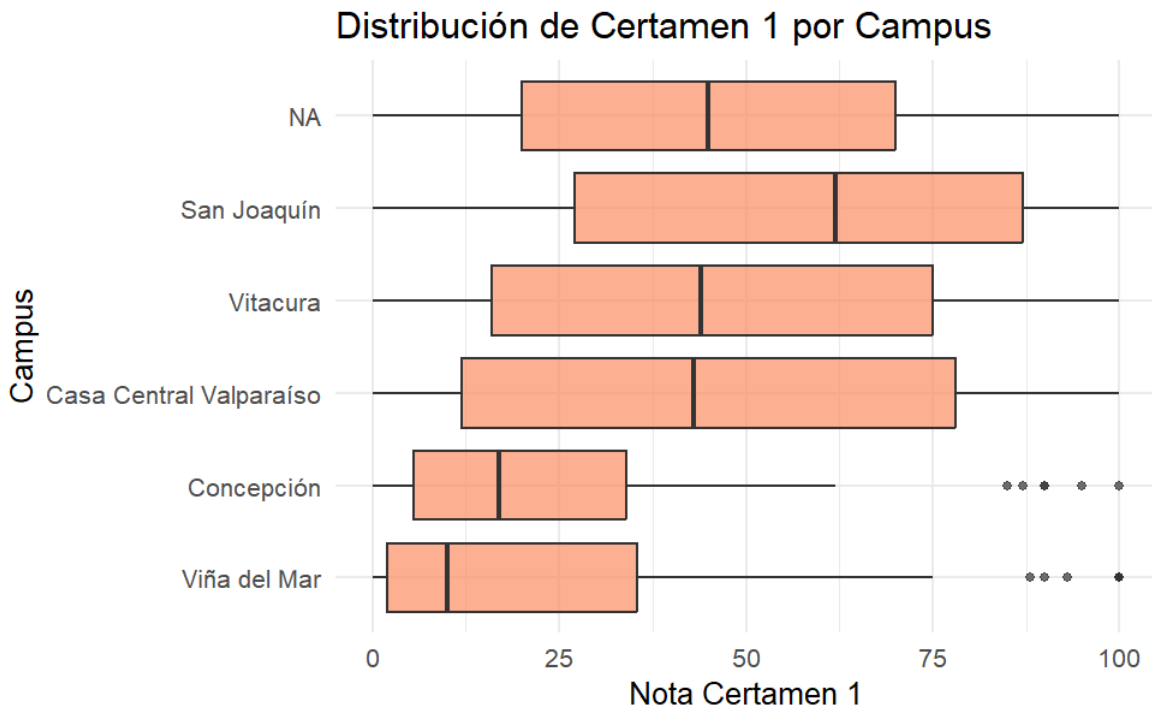


Figura 41: Distribución de notas de certamen 1 por campus

Estos hallazgos permiten identificar perfiles de riesgo según contexto académico, lo que puede orientar estrategias diferenciadas de apoyo.

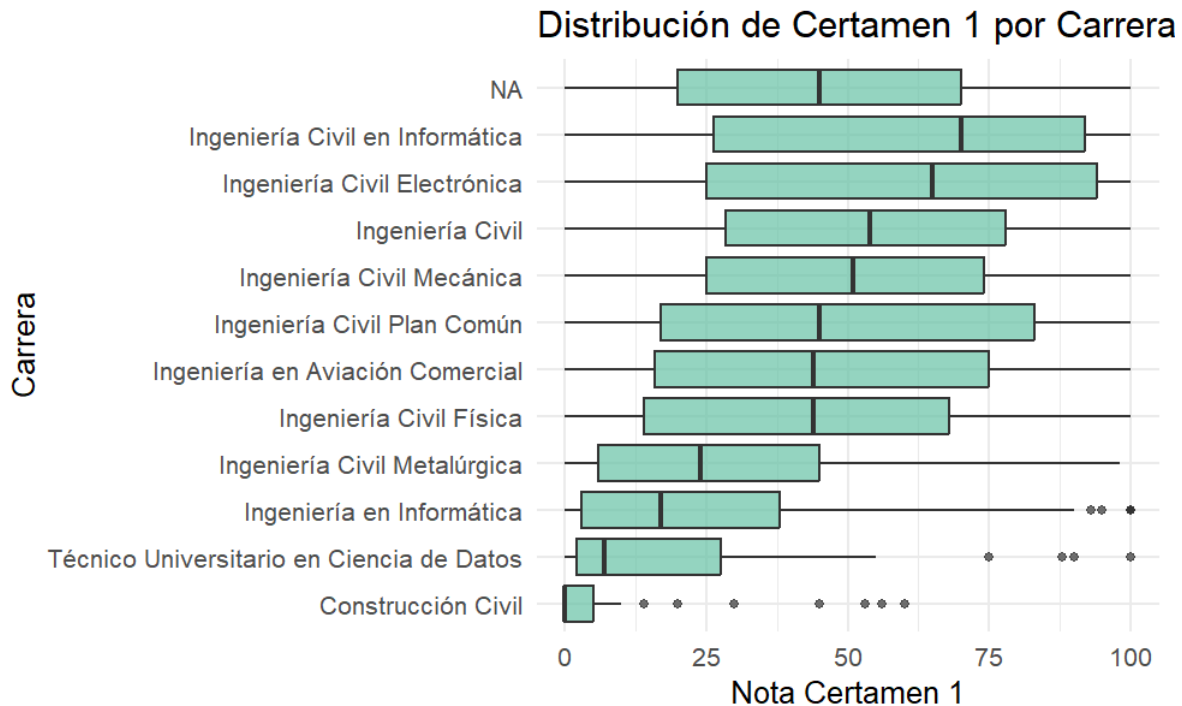


Figura 42: Distribución de notas de certamen 1 por carrera

4.5. Análisis del Test de Lawson

El Test de Razonamiento Científico de Lawson es una medida de habilidades de razonamiento formal, útil para predecir la capacidad de abstracción y el desarrollo de pensamiento lógico en estudiantes. En este estudio, se utilizó un punto de corte en 33 puntos para clasificar a los estudiantes en dos grupos: *bajo desempeño* ($LAWSON < 33$) y *desempeño adecuado o alto* ($LAWSON \geq 33$), con base en la premisa entregada por el Departamento de Física de la universidad.

Rendimiento en Certamen 1 según nivel de razonamiento lógico

La Tabla 23 muestra los estadísticos descriptivos del Certamen 1 según estos grupos. Los estudiantes con puntuaciones bajas en el test de Lawson (< 33) obtuvieron un promedio levemente mayor en el certamen (47.97 puntos) en comparación con quienes obtuvieron 33 o más (42.12 puntos). Sin embargo, esta diferencia debe analizarse con cautela, ya que:

- La mediana es mayor en el grupo con puntajes bajos (50 vs. 32), lo que sugiere una mayor concentración de notas altas en ese subconjunto.

- Las desviaciones estándar son altas en ambos casos, indicando una dispersión importante de las notas.

Tabla 23: Resumen estadístico del Certamen 1 según grupos de Puntajes Lawson

Grupo Lawson	n	Promedio	Desviación Estándar	Mediana
LAWSON < 33	69	47.97	35.73	50
LAWSON ≥ 33	385	42.12	34.31	32

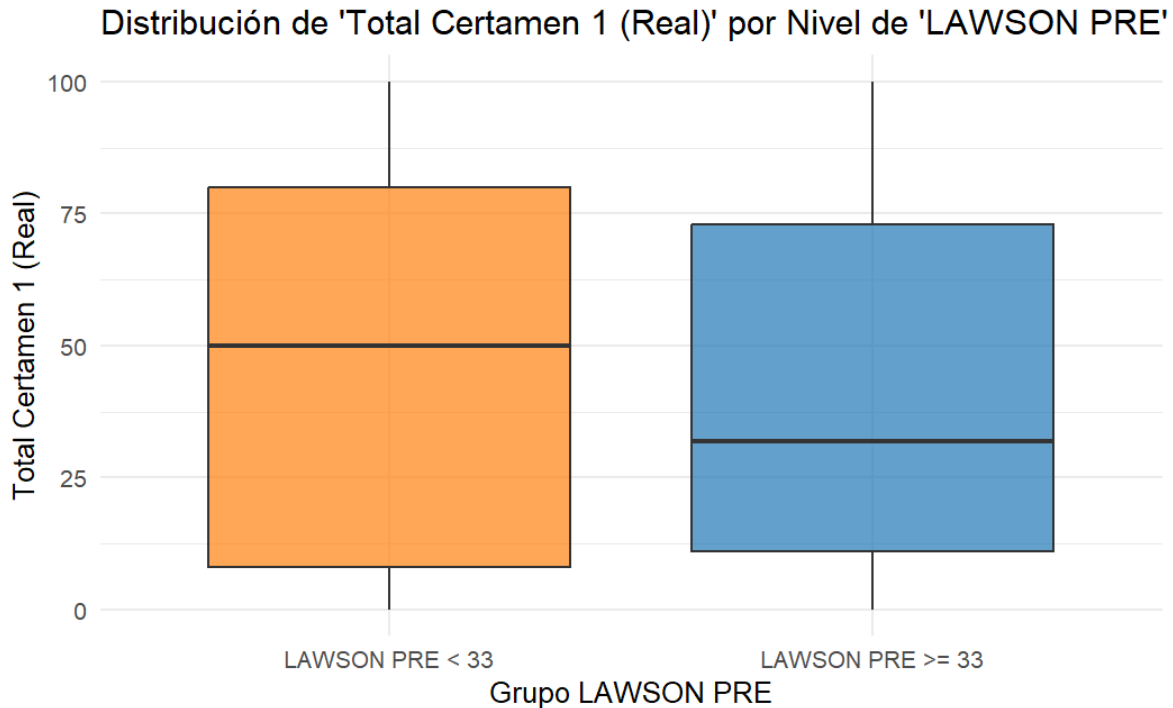


Figura 43: Distribución de Notas de Certamen 1 por nivel de test de Lawson

La Figura 43 refuerza esta observación, mostrando que la distribución es más simétrica en el grupo <33, mientras que el grupo ≥33 presenta una mayor concentración de puntajes bajos. Este resultado es contraintuitivo, ya que se esperaría que un mayor razonamiento lógico conlleve un mejor rendimiento académico. Una posible explicación radica en la heterogeneidad del grupo con puntaje alto en Lawson, donde podrían coexistir estudiantes con alto razonamiento abstracto pero sin suficiente preparación matemática o hábitos de estudio.

Relación entre razonamiento lógico y abstracción computacional

Dado que el test de Lawson mide habilidades de pensamiento formal, se exploró su relación con la dimensión de *Abstracción* del pensamiento computacional. La Tabla 24 muestra que

los estudiantes con puntajes bajos en Lawson (< 33) tienen, en promedio, una puntuación levemente mayor en la dimensión de abstracción (2.06 vs. 1.85).

Tabla 24: Resumen estadístico de la dimensión *Abstracción* según grupos de LAWSON PRE

Grupo LAWSON PRE	n	Promedio	Desviación Estándar	Mediana
LAWSON PRE < 33	49	2.06	0.92	2
LAWSON PRE ≥ 33	311	1.85	0.94	2

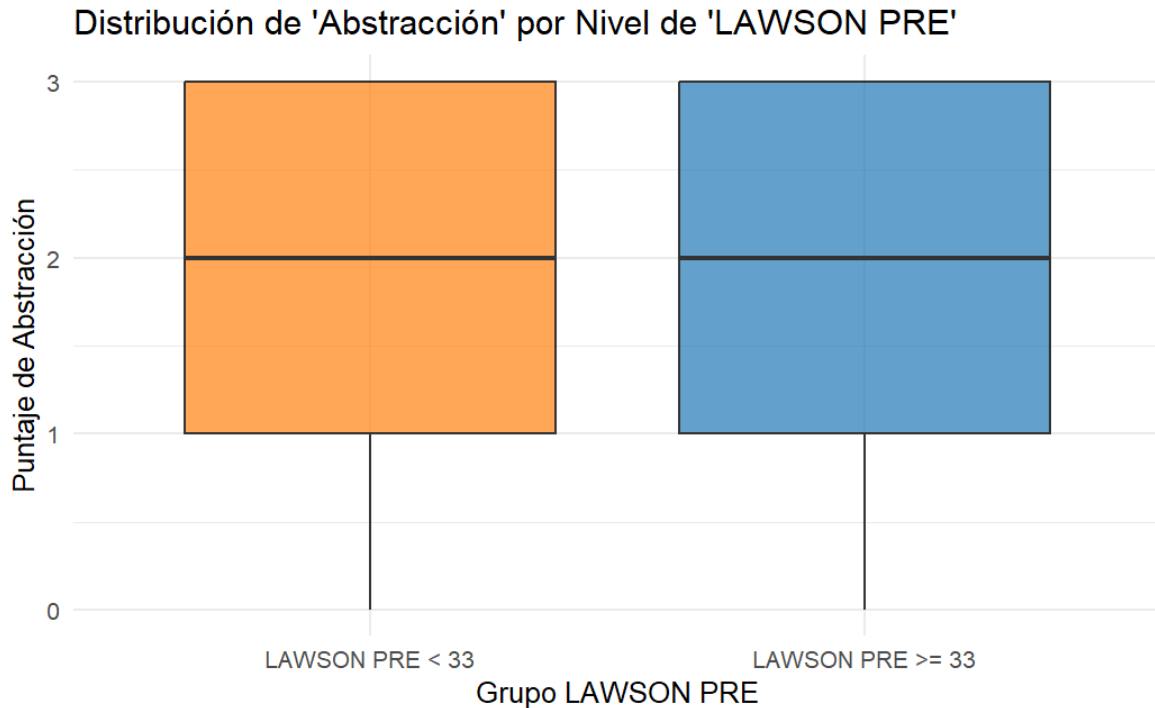


Figura 44: Distribución de puntaje de Abstracción por nivel de test de Lawson

La Figura 44 muestra distribuciones similares entre ambos grupos, con medianas idénticas y una dispersión comparable. No se observa una diferencia visual clara entre los grupos en términos de puntaje en la dimensión de *Abstracción*, lo que sugiere que el nivel de razonamiento formal medido por el test de Lawson no se asocia de manera evidente con esta dimensión del pensamiento computacional.

4.6. Evaluación del Modelo de Predicción para el Certamen 2

Se evaluó el rendimiento del modelo, entrenado con los datos del Certamen 1, al aplicarlo para predecir las notas del Certamen 2. A continuación, se presentan las estadísticas descriptivas de las predicciones, de las notas reales y del error de predicción.

Distribución de las Predicciones del Modelo

Tabla 25: Estadísticas Descriptivas de las Predicciones del Certamen 2

Estadístico	Valor
Mínimo	-20.93
1er Cuartil (Q1)	37.91
Mediana	52.73
Media	51.24
3er Cuartil (Q3)	67.06
Máximo	99.47
Desviación Estándar	21.40
Cantidad de valores perdidos (NA's)	807

Distribución de las Notas Reales del Certamen 2

Tabla 26: Estadísticas Descriptivas de las Notas Reales del Certamen 2

Estadístico	Valor
Mínimo	0.00
1er Cuartil (Q1)	20.00
Mediana	52.00
Media	51.34
3er Cuartil (Q3)	84.50
Máximo	100.00
Desviación Estándar	34.02
Cantidad de valores perdidos (NA's)	284

Análisis del Error de Predicción (Predicción - Real)

Tabla 27: Estadísticas del Error de Predicción para el Certamen 2

Estadístico	Valor
Mínimo	-89.32
1er Cuartil (Q1)	-25.45
Mediana	-6.84
Media	-4.34
3er Cuartil (Q3)	16.41
Máximo	77.36
Desviación Estándar	29.54
MSE (Error Cuadrático Medio)	890.62
RMSE (Raíz del MSE)	29.84
NA's	910

Diferencia Absoluta de la Predicción

Tabla 28: Estadísticas de la Diferencia Absoluta ($| \text{Predicción} - \text{Real} |$) para el Certamen 2

Estadístico	Valor
Mínimo	0.025
1er Cuartil (Q1)	10.74
Mediana	22.22
Media	24.57
3er Cuartil (Q3)	35.99
Máximo	89.32
Desviación Estándar	16.95
NA's	910

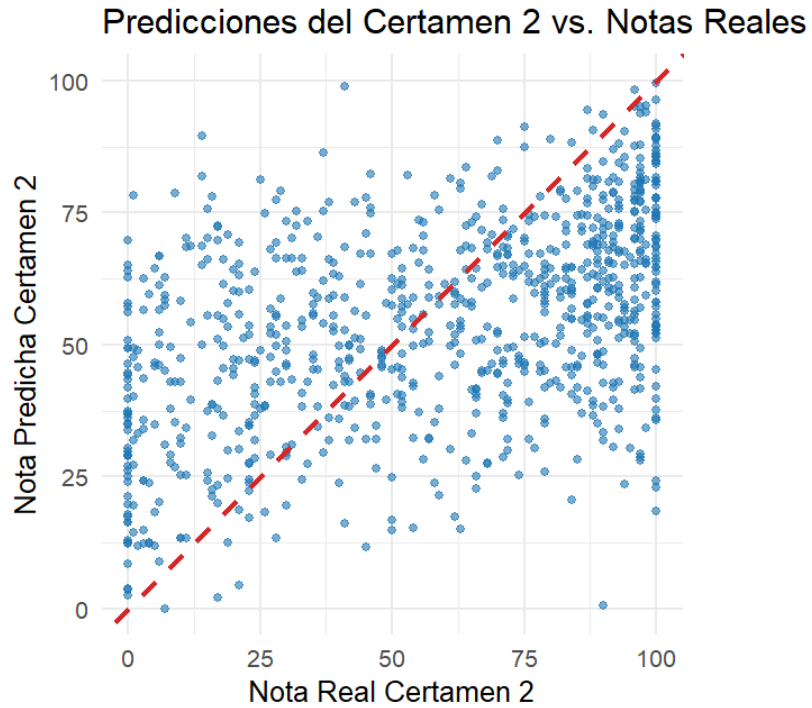


Figura 45: Gráfico de dispersión nota predicción vs nota real certamen 2

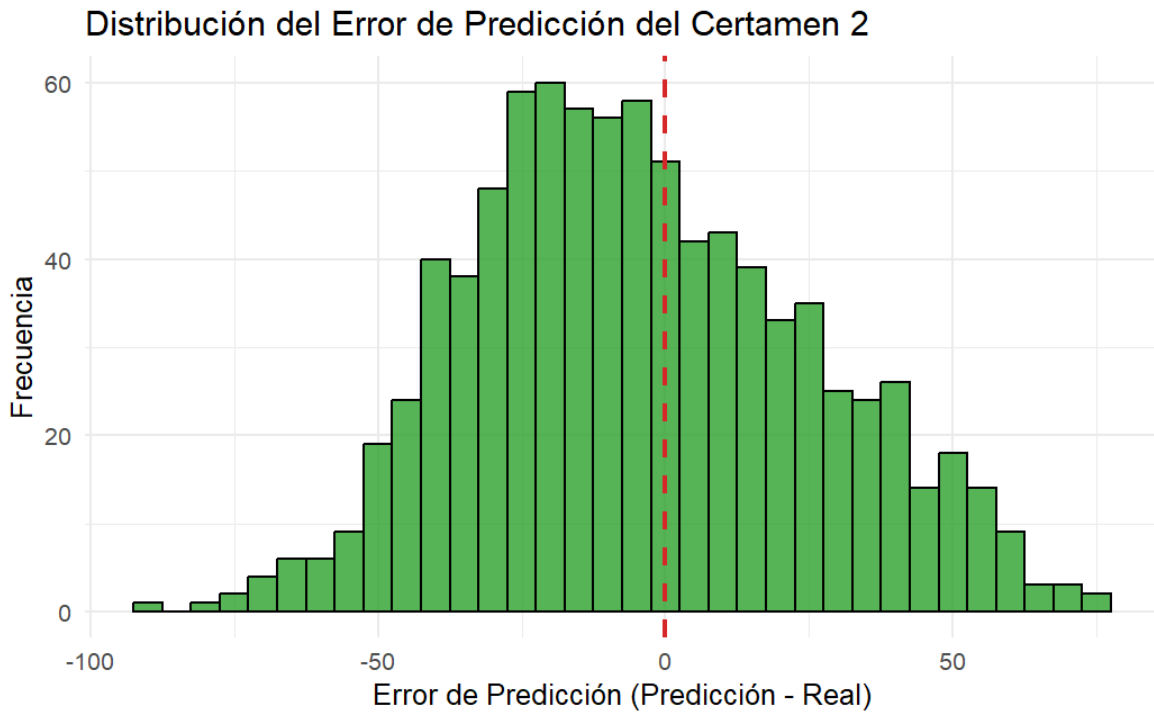


Figura 46: Distribución de error de predicción certamen 2

Aunque el modelo predice razonablemente el promedio general del Certamen 2 (Media de predicción: **51.24**, Media real: **51.34**), su **capacidad individual de predicción es limitada**, evidenciada por un RMSE elevado de **29.84** y una diferencia absoluta promedio de **24.57** puntos. Además, se observan valores extremos tanto en errores negativos como positivos, lo cual sugiere que el modelo tiende a errar considerablemente en ciertos casos individuales.

Este comportamiento puede deberse a varios factores:

- Cambios en la dificultad del Certamen 2 respecto al primero.
- Variables relevantes no incluidas en el modelo.
- Cambios en la motivación o nivel de esfuerzo de los estudiantes entre evaluaciones.

4.7. Correlación con la Nota Final del Curso

Se calcularon los coeficientes de correlación de Pearson entre diversas variables y la nota final del curso, con el objetivo de identificar los factores que más se asocian con el rendimiento académico global. A continuación, se presenta un resumen ordenado de mayor a menor correlación:

Tabla 29: Correlación de Variables con la Nota Final del Curso

Variable	Coefficiente de Correlación (r)
Total Certamen 1 (Real)	0.783
Puntaje Matemática M1	0.559
Puntaje Matemática M2	0.539
Pensamiento Computacional (Total)	0.465
Puntaje Lenguaje	0.423
Pensamiento Algorítmico	0.387
Puntaje NEM	0.374
Puntaje Ciencias	0.360
Puntaje Ranking	0.354
Nota Matemática	0.334
Reconocimiento de Patrones y Generalización	0.311
Descomposición	0.304
Abstracción	0.288
Autoeficacia Promedio	0.049
Motivación Promedio	0.041
Puntaje Historia	-0.032
LAWSON	0.007

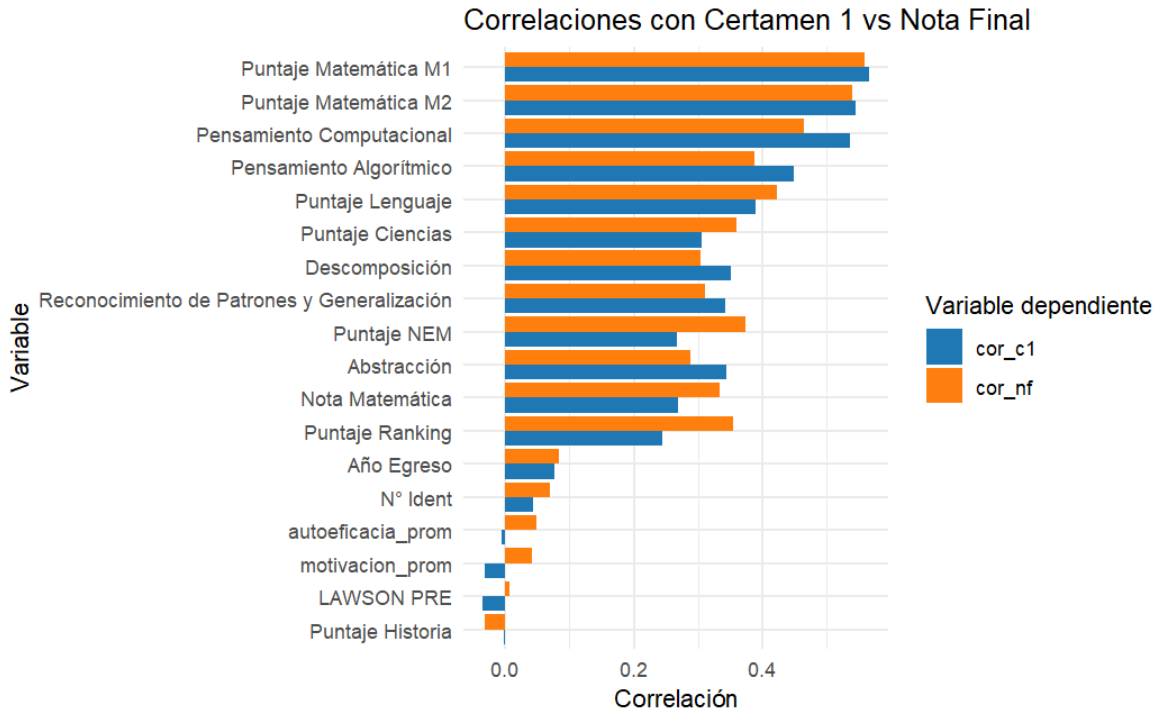


Figura 47: Gráfico de barras comparativo entre correlaciones con certamen 1 vs nota final

Como se observa en la Tabla 29, el resultado del **Certamen 1** muestra la correlación más alta con la nota final del curso ($r = 0,783$), lo que sugiere que este instrumento representa una medida muy representativa del desempeño global de los estudiantes.

Las **habilidades matemáticas** previas, medidas mediante los puntajes M1 y M2, también presentan correlaciones fuertes ($r \approx 0,54 - 0,56$), seguidas por variables de **pensamiento computacional**, con el total de esta dimensión alcanzando $r = 0,465$.

Factores como el **puntaje en lenguaje**, **NEM**, y **ranking** también muestran correlaciones moderadas, lo que respalda su utilidad como predictores del desempeño general.

En contraste, variables como **motivación**, **autoeficacia**, y el resultado del **test de Lawson** muestran correlaciones muy bajas ($r < 0,05$), lo que indica que, al menos en este contexto y muestra, tienen una relación débil con el rendimiento académico final.

Estos hallazgos refuerzan la idea de que el rendimiento en programación está más estrechamente asociado a habilidades cognitivas y conocimientos previos que a factores actitudinales o motivacionales, al menos cuando estos últimos son autorreportados mediante encuestas opcionales.

4.7.1. Modelo de Regresión Lineal Múltiple para la Nota Final del Curso

Con el objetivo de predecir el rendimiento final de los estudiantes en el curso, se ajustó un modelo de regresión lineal múltiple utilizando como variables predictoras los puntajes de ingreso (M1, M2, Lenguaje, NEM), las subdimensiones del pensamiento computacional y la nota obtenida en el Certamen 1.

La variable dependiente es la nota final del curso, y las variables independientes consideradas son:

- Puntaje Matemática M1
- Puntaje Matemática M2
- Puntaje Lenguaje
- Puntaje NEM
- Abstracción
- Pensamiento Algorítmico
- Descomposición
- Reconocimiento de Patrones y Generalización
- Total Certamen 1 (Real)

Tabla 30: Coeficientes del modelo de regresión lineal para la nota final del curso

Variable	Coef. Estimado	Error Estándar	p-valor
(Intercepto)	-28.42	6.04	< 0.001
Puntaje Matemática M1	0.0184	0.0068	0.0072
Puntaje Matemática M2	0.0165	0.0056	0.0031
Puntaje Lenguaje	0.0107	0.0058	0.0641
Puntaje NEM	0.0437	0.0051	< 0.001
Abstracción	-0.716	0.539	0.185
Pensamiento Algorítmico	0.555	0.579	0.338
Descomposición	-0.630	0.608	0.300
Reconocimiento de Patrones y Generalización	0.463	0.609	0.448
Total Certamen 1 (Real)	0.462	0.0163	< 0.001

- **R² Ajustado:** 0.6755
- **RSE (Error estándar residual):** 13.28

- **Estadístico F:** 224.9 (gl = 9 y 959), $p < 2,2 \times 10^{-16}$
- **Observaciones utilizadas:** 969

El modelo explica aproximadamente un **67.6 %** de la variabilidad de la nota final del curso, lo cual es un resultado sólido en el contexto educativo. La variable con mayor poder explicativo fue el **Certamen 1**, con un coeficiente altamente significativo ($p < 0,001$), indicando una fuerte relación positiva con el desempeño final.

También resultaron significativos los puntajes M1, M2 y NEM, lo que confirma la importancia de los antecedentes académicos. Aunque las dimensiones del pensamiento computacional no fueron significativas individualmente, su inclusión mejora el ajuste general del modelo. Esto sugiere que podrían estar aportando de forma conjunta o interactuar con otras variables no incluidas.

4.8. Propuestas de Clasificación

Con el objetivo de facilitar la interpretación de las predicciones del modelo y apoyar la toma de decisiones pedagógicas, se proponen dos estrategias para clasificar a los estudiantes en función de la nota predicha en el Certamen 1. Estas clasificaciones permiten agrupar a los alumnos según su **nivel de riesgo académico** y priorizar intervenciones tempranas.

4.8.1. Criterio 1: Nota de aprobación como punto de corte central

En esta propuesta, se utiliza la **nota de aprobación (55)** como punto central para definir los tramos de riesgo, empleando el **Error Estándar Residual (RSE)** del modelo para calcular las cotas superior e inferior. Específicamente:

- **Tramo rojo (riesgo alto):** Predicción $\leq 55 - \text{RSE}$
- **Tramo amarillo (riesgo medio):** $55 - \text{RSE} < \text{Predicción} < 55 + \text{RSE}$
- **Tramo verde (riesgo bajo):** Predicción $\geq 55 + \text{RSE}$

Este enfoque refleja la **incertidumbre estadística** del modelo en torno al umbral de aprobación, permitiendo una clasificación basada en márgenes de error esperados. Sin embargo, una limitación es que el **tramo amarillo tiende a ser amplio**, debido a que el RSE (en este caso, ≈ 26 puntos) es considerable en comparación con el rango de notas posibles (0–100). Esto genera que una proporción importante de estudiantes caiga en la categoría intermedia, dificultando la priorización clara de casos críticos.

4.8.2. Criterio 2: Clasificación basada en tramos equitativos de riesgo

Una segunda propuesta busca distribuir más equitativamente a los estudiantes en tres tramos de riesgo, utilizando la **distribución de las notas predichas** y su variabilidad para definir los cortes. En este enfoque no se busca una predicción precisa, sino una **clasificación orientada a la acción**, donde:

- El **tramo rojo** agrupa a los estudiantes con predicciones en el tercil inferior.
- El **tramo amarillo** incluye a aquellos en el tercil medio.
- El **tramo verde** contiene al tercil superior de estudiantes según nota predicha.

Esta estrategia presenta varias ventajas:

- Permite una distribución más uniforme de estudiantes entre los grupos, facilitando la planificación de apoyos diferenciados.
- No depende del umbral arbitrario de aprobación, sino que se adapta a la distribución real de las predicciones.
- Mejora la detección de estudiantes en riesgo relativo, incluso si su nota predicha es superior a 55 pero baja en comparación con sus pares.

Reclasificación posterior al Certamen 1

Independientemente del criterio adoptado, se recomienda realizar una **reclasificación posterior al Certamen 1**, una vez que se dispone de la nota real obtenida. Esta nota puede integrarse como una nueva variable predictora en modelos futuros (como en el caso del modelo de nota final del curso), lo que mejora significativamente el poder explicativo. Además, permite una clasificación más precisa del riesgo académico para la segunda parte del semestre, permitiendo así intervenciones más efectivas y focalizadas.

CONCLUSIONES

Este trabajo tuvo como finalidad explorar los factores que influyen en el rendimiento de estudiantes de primer año en un curso de programación, utilizando herramientas de análisis estadístico y modelado predictivo. A través del uso de datos provenientes de distintas fuentes — como resultados de la prueba PAES, encuestas de motivación y autoeficacia, el test de razonamiento de Lawson, y las calificaciones de certámenes — se construyó una base de datos integrada para analizar correlaciones, generar modelos de predicción, y proponer formas de clasificar a los alumnos según su nivel de riesgo de reprobación.

Los modelos desarrollados permitieron identificar variables con alta capacidad predictiva, como el éxito en el Certamen 1 y ciertos puntajes PAES, y otras que, pese a su respaldo teórico, no mostraron una relación tan fuerte con el desempeño real, como el test de Lawson. Estos hallazgos permiten apoyar la toma de decisiones docentes, generar planes de acompañamiento más efectivos y avanzar hacia una educación más personalizada y basada en datos.

5.1. Alcances y limitaciones

Uno de los principales aportes de este trabajo es la posibilidad de generar insumos tempranos para la intervención pedagógica en base a predicciones razonables del rendimiento de los estudiantes. Aunque los modelos desarrollados no alcanzan una precisión exacta, logran categorizar a los estudiantes en tramos de riesgo que permiten tomar decisiones informadas. En este sentido, la utilidad del modelo no está en predecir con certeza absoluta, sino en detectar tendencias y apoyar la acción docente.

El test de razonamiento de Lawson, si bien teóricamente debería mostrar alta correlación con el desempeño en programación, presentó una relación débil con los resultados reales. Esto puede deberse a que evalúa habilidades cognitivas generales más que aspectos específicos del pensamiento computacional. Además, se observó que el grupo de estudiantes con Lawson alto no tuvo mejor desempeño que sus pares, y la mediana de notas de ambos grupos fue similar.

Otro limitante importante fue la cantidad de datos faltantes. Muchos estudiantes no tenían registros PAES (posiblemente por ingreso vía programas especiales) o no respondieron las encuestas. Esto redujo el tamaño muestral efectivo y posiblemente afectó la representatividad del modelo.

En cuanto a las encuestas de motivación y autoeficacia, su carácter voluntario puede haber introducido sesgos de selección, además de que su diseño podría mejorarse mediante escalas estandarizadas y validación interna.

5.2. Perspectiva futura

A partir del trabajo realizado, se proponen varias líneas de acción y extensión para proyectos futuros:

- **Reclasificación posterior al Certamen 1:** Incorporar la nota real del Certamen 1 en los modelos subsiguientes mejora considerablemente la capacidad predictiva. Esto permite detectar a tiempo a quienes requieren apoyo adicional antes de instancias finales.
- **Mejora de las encuestas:** Se recomienda rediseñar los instrumentos de motivación y autoeficacia, incorporando preguntas de control para validar respuestas y empleando escalas estandarizadas ampliamente utilizadas en psicometría.
- **Modelado más avanzado:** Explorar técnicas como clustering para segmentar a los estudiantes por perfiles similares, o bien aplicar modelos de machine learning como redes neuronales o random forest, que podrían mejorar el poder predictivo en contextos con alta no linealidad.
- **Tratamiento de datos faltantes:** Diseñar métodos para imputar valores perdidos o generar predicciones intermedias que permitan incorporar a estudiantes excluidos por falta de datos iniciales.
- **Inclusión de nuevas variables:** Analizar otras posibles fuentes de información relevantes, como indicadores de participación en clases, uso de plataformas, asistencia, u otros elementos cualitativos observados por los docentes.

5.3. Reflexión Final

Este trabajo demuestra el potencial del análisis de datos educativos como herramienta para mejorar los procesos de enseñanza y aprendizaje. Si bien el modelo desarrollado no sustituye la evaluación docente ni el acompañamiento personalizado, constituye un valioso complemento para tomar decisiones más informadas y justas. Además, abre la puerta a una cultura institucional basada en evidencias, donde los datos no se almacenan sino que se analizan y se transforman en acciones concretas.

Finalmente, este trabajo deja abierta una invitación a quienes deseen profundizar en el tema, tanto desde la investigación educativa como desde la ingeniería de datos, para seguir desarrollando modelos predictivos, mejorando instrumentos de medición, y explorando nuevas formas de acompañar el aprendizaje de manera significativa y contextualizada.

REFERENCIAS BIBLIOGRÁFICAS

- [Ambrosio et al., 2011] Ambrosio, A., Costa, F., Almeida, L., Franco, A., y Macedo, J. (2011). Identifying cognitive abilities to improve cs1 outcome. *Proceedings - Frontiers in Education Conference*.
- [Bandura y Wessels, 1997] Bandura, A. y Wessels, S. (1997). *Self-efficacy*. Cambridge University Press Cambridge.
- [Bao et al., 2018] Bao, L., Xiao, Y., Koenig, K., y Han, J. (2018). Validity evaluation of the lawson classroom test of scientific reasoning. *Physical Review Physics Education Research*, 14(2):020106.
- [Bergin y Reilly, 2005] Bergin, S. y Reilly, R. (2005). Programming: factors that influence success. En *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pp. 411–415.
- [ChileAtiende, 2025] ChileAtiende (2025). Prueba de acceso a la educación superior (paes). <https://www.chileatiende.gob.cl/fichas/81332-prueba-de-acceso-a-la-educacion-superior-paes>. Última modificación: 30 de mayo de 2025.
- [Edgar y Antonio, 2014] Edgar, S. M. y Antonio, P. J. (2014). Lógica y abstracción en la formación de ingenieros: una relación necesaria. *Ingeniería, investigación y tecnología*, 15(2):299–310.
- [Fonden Calzadilla, 2020] Fonden Calzadilla, J. C. (2020). Importancia del pensamiento abstracto. su formación en el aprendizaje de la programación. *EduSol*, 20(72):122–135.
- [Guzdial, 2010] Guzdial, M. (2010). Why is it so hard to learn to program. *Making Software: What Really Works, and Why We Believe It*. O'Reilly Media, pp. 111–124.
- [Kallia, 2023] Kallia, M. (2023). The search for meaning: Inferential strategic reading comprehension in programming. En *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*, pp. 1–14.
- [Kramer, 2007] Kramer, J. (2007). Is abstraction the key to computing? *Commun. ACM*, 50:36–42.
- [Kwon, 2017] Kwon, K. (2017). Novice programmer's misconception of programming reflected on problem-solving plans. *International Journal of Computer Science Education in Schools*, 1:14.
- [Lockwood et al., 2016] Lockwood, E., DeJarnette, A. F., Asay, A., y Thomas, M. (2016). Algorithmic thinking: An initial characterization of computational thinking in mathematics. *North American Chapter of the International Group for the Psychology of Mathematics Education*.

- [Meza *et al.*, 2024] Meza, F., Vásquez, A., y San Martín, D. (2024). Validation of a bebras-based test to assess computational thinking abilities in first-year college students. En *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*, SIGCSE 2024, p. 1750–1751, New York, NY, USA. Association for Computing Machinery.
- [Owolabi *et al.*, 2014] Owolabi, J., Olanipekun, P., e Iwerima, J. (2014). Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming. *GSTF Journal on Computing (JoC)*, 3:1–6.
- [Pinto, 2020] Pinto, F. J. V. (2020). Impacto del uso de un lenguaje basado en bloques en el desarrollo de habilidades de programación. Trabajo para optar al título de Ingeniero Civil Informático.
- [Restrepo y González, 2007] Restrepo, L. F. y González, J. (2007). De pearson a spearman. *Revista Colombiana de Ciencias Pecuarias*, 20(2):183–192.
- [Ryan y Deci, 2000] Ryan, R. M. y Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67.
- [Shute *et al.*, 2017] Shute, V. J., Sun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22:142–158.
- [Sirkiä y Sorva, 2012] Sirkiä, T. y Sorva, J. (2012). Exploring programming misconceptions: an analysis of student mistakes in visual program simulation exercises. En *Proceedings of the 12th Koli calling international conference on computing education research*, pp. 19–28.
- [Tang *et al.*, 2020] Tang, X., Yin, Y., Lin, Q., Hadad, R., y Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148:103798.
- [Wing, 2006] Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.