Facultad de Ciencias Económicas y Empresariales.

Estrategias de Estudio en programación de grados no STEM

Laura García Arnau

5º E3-Analytics

Director: José Luis Arroyo Barriguete.

Madrid, 2024-25

Resumen

Este Trabajo de Fin de Grado analiza cómo se percibe y afronta la asignatura de programación en los grados no STEM, concretamente en programas híbridos como *Business Analytics.* A través de una encuesta ómnibus aplicada a estudiantes de la Universidad Pontificia Comillas, se estudian las estrategias de estudio, la dificultad percibida y el rendimiento académico en programación. Mediante un análisis de componentes principales, modelos de regresión (OLS) y árboles de regresión, se exploran los factores que inciden en el aprendizaje, destacando variables como el género, el grado cursado o la estrategia de memorización. Los resultados muestran que las mujeres tienden a percibir mayor dificultad, aunque no presentan un rendimiento inferior. Además, se concluye que las estrategias basadas en la memorización están asociadas con un menor desempeño académico y mayor dificultad percibida. El estudio proporciona recomendaciones prácticas para mejorar la enseñanza y el aprendizaje de programación en contextos no técnicos.

Palabras Clave

Grados no STEM, programación, dificultad percibida, rendimiento académico, estrategias de estudio

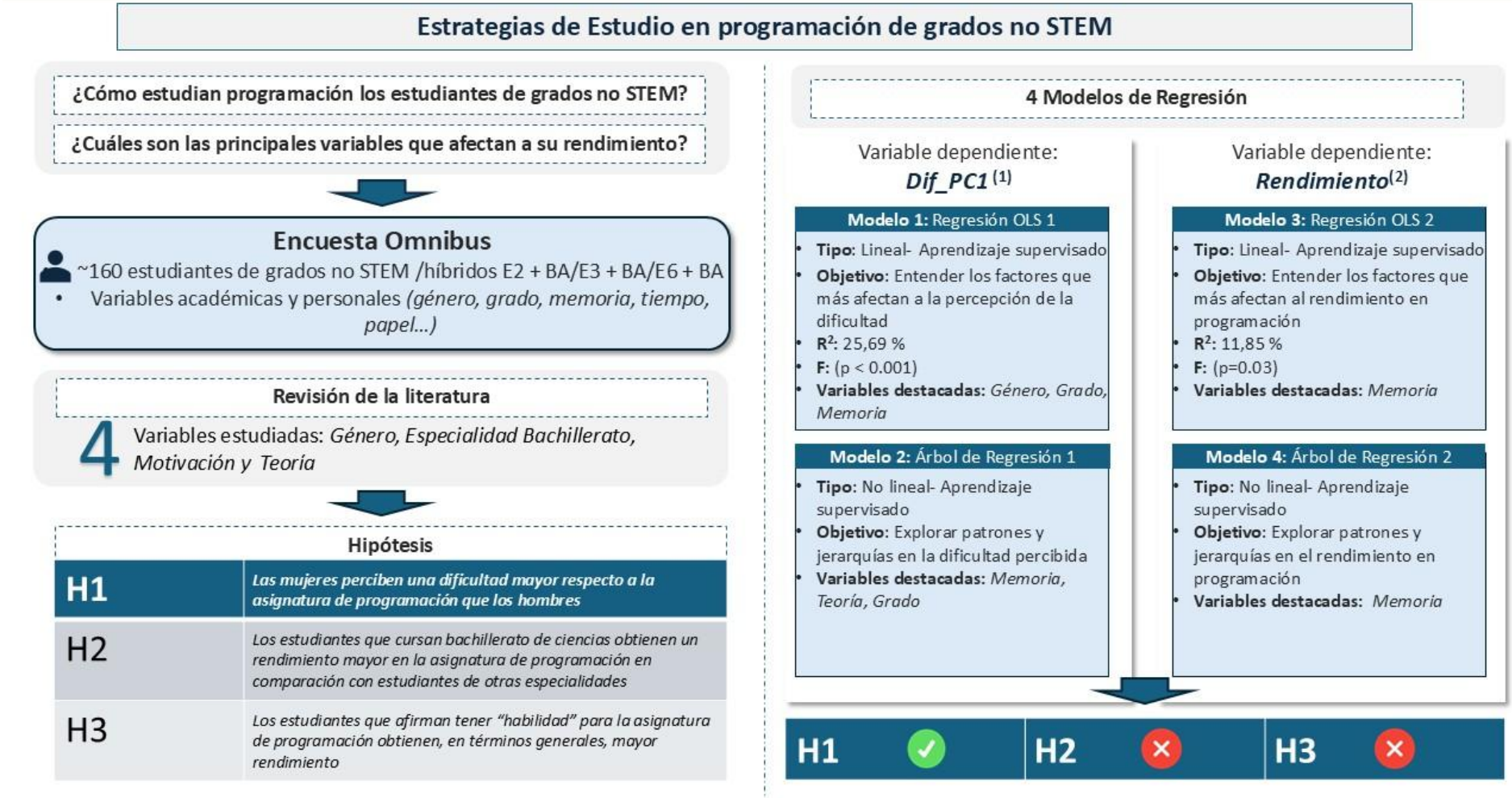
Abstract

This undergraduate thesis explores how programming is perceived and approached in non-STEM degrees, specifically in hybrid programs such as Business Analytics. Based on omnibus survey data collected from students at Universidad Pontificia Comillas, the research investigates study strategies, perceived difficulty, and academic performance in programming. A Principal Component Analysis (PCA) was conducted to synthesize perceived difficulty across technical subjects, and its main component was used as a key dependent variable. Through regression models (OLS) and regression trees, the study identifies key factors influencing learning outcomes, highlighting the role of gender, degree program, and memorization-based strategies. Results show that women report higher perceived difficulty, although their academic performance is not lower. Additionally, memorization as a learning method is associated with poorer results. The study offers practical recommendations to enhance programming education and learning in non-technical academic contexts.

Key Words

Non-STEM degrees, programming, perceived difficulty, academic performance, study strategies

Graphical abstract



Notas: (1) Variable resultado de un PCA realizado sobre las variables percepción de dificultad en Matemáticas, Estadística, Machine Learning y Programación. (2): ¿Cuál consideras que ha sido tu rendimiento (nota obtenida) en las asignaturas de programación (Programación)?

Contenido

1. Introducción……1

1.1 Motivación del trabajo……1

1.2 Objetivos de la investigación……2

2. Revisión de la Literatura……3

2.1 La brecha de género en el rendimiento de la asignatura de programación……3

2.2 Especialidad cursada en bachillerato……7

2.3 Relación Motivación-Rendimiento……7

2.4 Relación Teoría-Práctica……8

2.5 Resumen de las hipótesis……9

3. Material y Métodos……10

3.1 Descripción de la muestra……10

3.2 Análisis de las variables independientes del modelo……11

3.3 Formulación de los modelos……14

3.3.1 Análisis de la variable dependiente del modelo - Dif_PC1……14

3.3.1.1 Explicación Análisis de Componentes Principales (PCA)……14

3.3.2 Análisis de la variable dependiente del modelo – Rendimiento……17

4. Resultados y Discusión……19

4.1 Variable Dependiente Dificultad Percibida en Programación (Dif_PC1)……19

4.1.1 MODELO 1- Regresión OLS 1……19

4.1.2 MODELO 2- Árbol de Regresión 1……20

4.2 Variable Dependiente Rendimiento en Programación (Rendimiento)……21

4.2.1 MODELO 3- Regresión OLS 2……21

4.2.2 MODELO 4- Árbol de Regresión 2……22

5. Conclusiones……23

5.1 Sobre Dificultad Percibida……23

5.2 Sobre Rendimiento……24

5.3 Conclusiones finales y aplicación práctica……25

5.4 Limitaciones del estudio y futuras líneas de investigación……25

6. Declaración respecto al uso de CHAT GPT u otras herramientas de IAG……27

7. Referencias……28

8. Anexo 1: Encuesta……30

8.1 Anexo 1: Encuesta……30

1. Introducción

1.1 Motivación del trabajo

En las últimas décadas, la programación y sus distintas ramas han ido adquiriendo relevancia creciente, sobrepasando su tradicional aplicación en los grados STEM y penetrando otros grados orientados a las ciencias sociales o grados híbridos como los de *business analytics*. Esta tendencia refleja las demandas del mercado actual, y como el manejo de habilidades tecnológicas, como el “Big Data” o la inteligencia artificial son cada vez más valoradas en el mundo laboral. Dentro del ámbito académico, disciplinas como la psicología o la economía están incluyendo asignaturas de programación en sus estudios y en el ámbito profesional, cada vez es más frecuente la demanda de habilidades tecnológicas por parte de las empresas.

En un mundo donde se prevé que la inteligencia artificial irrumpa en todos los ámbitos de la vida profesional y personal, la formación en programación hace a los alumnos más competitivos en un entorno laboral cada vez más digitalizado. Figueiredo y García-Peñalvo (2018) afirman que "aprender a programar está de moda." (p.1). Además, señalan que "son innumerables las iniciativas de varias entidades y organizaciones con el objetivo de promover el estudio del pensamiento computacional y, consecuentemente, la programación." (p. 1). En este sentido, la habilidad para abordar problemas de forma estructurada y lógica se convierte en una competencia esencial, donde “lo importante ya no es sólo saber ciencias, matemáticas y tecnología, sino saber resolver problemas en contextos reales “pensando como” matemáticas/os, científicas/os e ingenieras/os” (Domènech-Casal, 2018, p.3).

Más allá de las necesidades actuales, el aprendizaje de la programación ha sido históricamente una de las asignaturas más desafiantes dentro del programa académico, tanto para los estudiantes como para los docentes. Figueiredo y García-Peñalvo (2018) destacan que las dificultades en la enseñanza y el aprendizaje de la programación son una preocupación constante para estudiantes y docentes. Este reto es especialmente significativo en las primeras etapas del aprendizaje, cuando los programadores novatos se enfrentan a una serie de dificultades y carencias. Existe un amplio consenso entre los estudiantes de que los cursos de programación son percibidos como particularmente difíciles y, a menudo, presentan altas tasas de abandono (Ala-Mutka y Järvinen, 2005).

La complejidad de la disciplina es tal que se considera que una persona promedio necesita una cantidad considerable de tiempo para adquirir un conocimiento profundo. Según Winslow (1996), un novato necesitaría aproximadamente 10 años de experiencia para convertirse en un programador experto.

De acuerdo con Figueiredo y García-Peñalvo (2018), "el fracaso en el aprendizaje de la programación es un problema que afecta a todos los niveles educativos, pero es especialmente preocupante en la educación superior" (p.1), lo que subraya la necesidad urgente de abordar las barreras que enfrentan los estudiantes en este ámbito.

En el contexto no STEM, la introducción de programación plantea dificultades pedagógicas añadidas, pues los alumnos no suelen contar con una formación previa en fundamentos de programación. Esto hace que se creen barreras cognitivas y emocionales a la hora de aprender, de hecho, sus estudios con enfoque cualitativo orientados al campo de las ciencias sociales o las letras más puras pueden afectar a su percepción de los números y estrategias de estudio. Otros problemas pueden surgir como consecuencia de una “estigmatización de la programación” o

falta de motivación, de forma que algunos estudiantes sientan que la programación está únicamente destinada para aquellos con una formación técnica desarrollada.

En este contexto, surge la oportunidad y la necesidad de tratar esta realidad, analizando cómo los alumnos de estos programas no STEM abordan el estudio de la asignatura de programación, las dificultades que enfrentan, y cómo hacer para optimizar su aprendizaje y futura aplicación. Este trabajo, por tanto, nace como motivación de contribuir al diseño de estrategias de estudio más adaptadas a las necesidades de alumnos de grados no STEM.

1.2 Objetivos de la investigación

Este TFG forma parte de un proyecto de investigación y consta de 4 TFGs diferentes, cada uno evaluando un tipo de materias en grados no STEM. Se engloba en el marco del proyecto de innovación docente "Non-Binary Approaches to Higher Education: Quantitative Analysis and Fact-Based Research" (Ref. 221, U. Sevilla) y la Cátedra Santalucía de Analytics for Educación de la Universidad Pontificia Comillas.

El presente TFG se centra en el estudio de la asignatura de programación, y cómo ésta se comprende dentro de grados no STEM, y más concretamente en los denominados grados híbridos. El estudio se basa en el análisis de datos recogidos a través de una encuesta ómnibus diseñada para captar información detallada sobre las estrategias de estudio empleadas por estudiantes de grados híbridos (*Business Analytics*) al abordar la asignatura de programación. La encuesta abarca una amplia variedad de aspectos, desde preguntas generales relacionadas con el género y el grado académico que cursan los estudiantes, hasta cuestiones más específicas que indagan en sus percepciones sobre la programación, las estrategias de estudio empleadas y el rendimiento académico presentado en la asignatura.

Desde un punto de vista práctico, este estudio pretende generar un impacto en distintos ámbitos:

1. A nivel académico: se pretende lograr un entendimiento más amplio de cómo los estudiantes afrontan el estudio de la programación.
2. Mejora de los métodos de enseñanza y entrenamiento de los docentes: además de alcanzar mejores resultados, los hallazgos de este estudio pueden utilizarse para diseñar materiales didácticos adaptados o guías para los docentes, enfocadas en métodos pedagógicos que fomenten el aprendizaje.
3. Impacto en el futuro profesional de los estudiantes: al impulsar el aprendizaje de programación, los estudiantes de grados no STEM estarán mejor preparados para enfrentarse a las demandas del mercado laboral actual, que valora cada vez más las competencias digitales y tecnológicas. Según la Comisión Europea (2013), en las décadas próximas, la necesidad de perfiles STEM superará la provisión de estos, tanto en términos de cantidad como de calidad. Por ello, el impulso de estas habilidades más tecnológicas representa un elemento crucial para el avance de la sociedad, ya que no solo responde a las necesidades del mercado laboral, sino que también favorece el avance de la innovación y la competitividad global.

En definitiva, este estudio no se limita al plano teórico, sino que aspira a trascenderlo, proporcionando una contribución práctica como parte de una investigación más amplia. Así, con toda la información recogida, el estudio no solo contribuirá al éxito académico de los estudiantes, sino que también tendrá un impacto positivo en su desarrollo profesional y en su capacidad para adaptarse a un entorno laboral en constante evolución.

2. Revisión de la literatura

El aprendizaje en el ámbito de la programación ha sido siempre objeto de interés en la educación superior, dado su impacto en el rendimiento académico de los estudiantes y en sus altas tasas de abandono. A medida que las tecnologías y las herramientas para la enseñanza evolucionan, la comprensión de los factores que influyen en el dominio de la programación se vuelve crucial para optimizar las estrategias pedagógicas y reducir las tasas de fracaso. “We believe that learning to program is problematic, and that the results achieved by students in programming courses do not correlate well with their other academic results” (Fincher et al., 2006, p.2).

Muchos autores han abordado este problema en sus estudios y, además de identificar posibles explicaciones, han llegado a la conclusión de que el aprendizaje de la programación no depende únicamente de la habilidad técnica del estudiante, sino también de una combinación de factores psicológicos y metodológicos. Aspectos como la motivación, la autoeficacia y la disposición para aprender influyen significativamente en el proceso. Asimismo, las estrategias de estudio adoptadas, ya sea el aprendizaje basado en la memorización o el aprendizaje activo, desempeñan un papel clave en el rendimiento académico.

Así, las variables más consideradas como predictoras del rendimiento de los estudiantes en programación incluyen la capacidad analítica, las preferencias de estilo de trabajo, el género y el nivel de comodidad de los estudiantes con la asignatura. El estudio de Fincher et al. (2006) concluyó que el nivel de comodidad fue el mejor predictor del éxito, seguido por el trasfondo matemático como el segundo predictor más relevante, y la atribución del éxito a la suerte (con una correlación negativa) ocupando el tercer lugar.

Aún a pesar de que muchos autores han tratado este tema, parece no haber un consenso al respecto. Aunque el aprendizaje de la programación se reconoce como un fenómeno multidimensional, sigue siendo necesario un estudio más profundo para comprender la interacción entre los distintos factores que influyen en el rendimiento estudiantil. La divergencia de los hallazgos sugiere que variables como el género, la motivación, las estrategias de estudio o el contexto pedagógico sean analizadas de manera más integral para desarrollar enfoques de enseñanza más precisos.

2.1 La brecha de género en el rendimiento de la asignatura de programación

La brecha de género en el ámbito de la informática y la programación ha sido ampliamente estudiada en el contexto de las carreras STEM. Sin embargo, su impacto en grados no STEM, donde la programación se comprende como una asignatura complementaria, ha recibido menos atención. Para la revisión de la literatura, la gran parte de las ocasiones utilizaremos fuentes académicas que abordan esta temática dentro del contexto STEM, ya que la mayoría de los factores que influyen en nuestras variables son universales, independientemente de si se trata de una carrera directamente relacionada con la informática o no. Así, factores como la brecha de género en el contexto tecnológico, el rendimiento de los estudiantes o el método utilizado para el estudio de la programación son igualmente relevantes en ambos contextos.

Además, el enfoque pedagógico y las estrategias didácticas empleadas para enseñar programación, aunque puedan variar en su grado de especialización, también presentan implicaciones importantes en ambos ámbitos. Por lo tanto, aunque los grados no STEM no se enfoquen exclusivamente en la informática, los estudios provenientes del contexto STEM proporcionan una base sólida para abordar los factores que influyen en la enseñanza y el

aprendizaje de la programación en estas carreras no tecnológicas, ayudando a identificar posibles estrategias y mejorar la experiencia de aprendizaje.

Gran parte de la literatura aborda la brecha de género en el ámbito digital, destacando cómo el mundo tecnológico y científico continúa estando predominantemente liderado por hombres. Esta brecha digital de género se manifiesta en diversos aspectos, como la baja representación femenina en la industria tecnológica y la escasa matriculación de mujeres en carreras relacionadas con las ciencias o la informática.

Hay algunos que, además de estos factores, también atribuyen la brecha de género digital a la diferencia en el dominio de las TIC entre hombres y mujeres. “Ante la escasez de mujeres dedicadas a este ámbito del conocimiento, ¿se debe asumir que los varones tienen más capacidad?” (Jiménez y Fernández, 2016, p. 2)

La investigación *"Desempeño académico en la enseñanza de la programación: consideraciones desde la perspectiva de género"* se llevó a cabo en la Universidad de Cienfuegos, analizando el rendimiento de estudiantes en cinco asignaturas de programación. El objetivo de la investigación se centró en describir el desempeño académico en las asignaturas relacionadas con la programación, desde una perspectiva de género.

Para ello, se empleó una metodología cuantitativa-descriptiva y el estudio fue de tipo longitudinal. En cuanto a la población y muestra, el estudio se centró en los estudiantes de la Universidad de Cienfuegos "Carlos Rafael Rodríguez", siendo la muestra inicial de 28 estudiantes, que se redujo a 12 al final del estudio.

Para la recolección de datos, se recogieron registros académicos de los estudiantes, siendo las asignaturas analizadas: a) Introducción a la programación, b) Diseño y Programación Orientada a Objetos, c) Estructura de Datos I, d) Estructura de Datos II, e) Programación Web. Las variables se conformaron en dos: la variable estudiante nominal y dicotómica (masculino/ femenino), y las variables ordinales, siendo las calificaciones en las 5 asignaturas relacionadas con programación y categorizadas en cuatro niveles: Mal (2), Regular (3), Bien (4), Excelente (5). Finalmente, el estudio utilizó métodos cuantitativos y la prueba U de Mann-Whitney para comparar si había diferencias significativas o no entre las calificaciones de hombres y mujeres. Esta prueba se eligió ya que las variables son ordinales y no se asumió una distribución normal de los datos.

Los resultados de la investigación revelaron que, si bien la programación presenta gran dificultad para la mayoría del estudiantado, las mujeres reflejan un menor desempeño que los hombres (demostrado mediante la prueba de U de Mann-Whitney). En ninguna de las 5 asignaturas, las mujeres lograron obtener como promedio una calificación superior a la de los hombres.

Las potenciales explicaciones de estos resultados son difusas y variadas. En el estudio se menciona la posibilidad de que las mujeres entren a las carreras de informática con menos experiencia previa en programación que los hombres, lo que puede influir en su rendimiento académico. Como señalan Mora, Coto y Villalobos (2017, p. 5), "las mujeres ingresan a las carreras universitarias de informática con menos experiencia previa en programación que los hombres". Además, se hace referencia a casos de discriminación en el aula, donde algunas estudiantes entrevistadas reportaron actitudes misóginas y machistas de los docentes, quienes expresaron que "las mujeres no piensan con lógica para programar" (Razo Godínez, 2008, p. 85).

Finalmente, se destaca la necesidad de implementar medidas que no solo aumenten la matrícula de mujeres en carreras STEM, sino que también favorezcan su permanencia en estos campos, ya que, como se indica en García, Vázquez, García y González (2020), el problema no se limita a aumentar la matrícula, sino también a evitar el abandono de los estudios y fomentar la incorporación y permanencia en la profesión. Según Amores (2020, p. 7), "casi el doble de las mujeres abandona estos campos científicos más que los hombres".

Con el fin de seguir explorando las percepciones de la asignatura de programación desde un punto de vista de género, se llevó a cabo un análisis a partir de un estudio realizado por Gunasekara (2021) en el *International Journal of Research and Innovation in Social Science (IJRISS).* En él, se trataron cuestiones como las opiniones de los estudiantes sobre la asignatura de programación, la dificultad que percibían y si estas percepciones y desafíos son específicos de género o no.

La población objetivo del estudio estuvo compuesta por estudiantes de primer año del programa HNDIT en SLIATE, con un total de 1,735 estudiantes en 2018, de los cuales 317 fueron seleccionados como muestra utilizando la tabla de Krejcie y Morgan. La muestra incluyó principalmente mujeres (n = 169) y hombres (n = 94). Se utilizó un muestreo por conglomerados, seleccionando al azar siete Institutos Tecnológicos Avanzados. Los datos se recopilaron a través de un cuestionario autoadministrado dividido en tres secciones: experiencia y motivación en programación, problemas y enfoques de resolución, e información demográfica.

Los resultados del estudio indicaron, en primer lugar, que no existen diferencias significativas entre hombres y mujeres en cuanto a su experiencia o conocimiento previo en programación, lo que contradice lo anteriormente mencionado por Mora, Coto y Villalobos (2017). “The percent of male students having previous experience in programming is nearly the same as of female students” (Gunasekara, 2021, p. 154).

En lo que respecta a la percepción de dificultad, según el estudio, los hombres consideran la programación el doble de interesante que las mujeres, quienes, además, la perciben como considerablemente más compleja. Como señala Gunasekara (2021), "The male proportion stating that programming is interesting is as double as female proportion" (p. 154). Por otro lado, "female students feel programming is difficult. As a percentage, it is 62.1% of all students. Thus, when they face programming-related problems, they tend to seek help from lecturers, demonstrators, or from peers" (p. 154).

Esta menor motivación e interés por parte de las mujeres podría estar impactando de forma negativa en su rendimiento académico. El sentimiento de dificultad hacia la programación, sumado a una percepción de que la materia es más compleja, puede generar un ciclo de desmotivación. Esta falta de autoconfianza podría llevarlas a evitar enfrentar desafíos más complejos por temor a fracasar, lo que limita su progreso académico en comparación con sus compañeros varones.

Otro artículo titulado “Attitude, Gender and Achievement in Computer Programming" investigó la actitud de los estudiantes, el género y su rendimiento en esta área, desarrollando una escala de 35 ítems tipo Likert de cinco puntos para medir las actitudes hacia la programación, la cual fue administrada a 179 estudiantes matriculados en un curso introductorio de programación informática. La fiabilidad global de la escala se estableció con un coeficiente de 0,96. Los resultados mostraron que la actitud de las mujeres hacia la programación era considerablemente más baja que la de los hombres. Según los hallazgos “…attitude of the male students towards

computer programming is meaningfully higher than that of attitudes toward programming" (Baser, 2013, p.252)

En conclusión, los hombres muestran una actitud significativamente más positiva hacia la programación, lo que se traduce directamente en un mejor desempeño en esta área. Su mayor confianza en sus habilidades y la percepción de la programación como algo interesante y accesible les permite afrontar los desafíos con mayor disposición y perseverancia.

Continuando en el marco del análisis sobre la posible brecha existente en el rendimiento académico cuando hablamos de programación, el estudio *"Learning Programming, Problem Solving and Gender: A Longitudinal Study"* de Yurdugül y Aşkar (2013) analiza las diferencias de género en el aprendizaje de la programación, centrándose en tres tipos de conocimiento: conceptual, sintáctico y estratégico.

El objetivo del estudio fue determinar el desarrollo del aprendizaje de programación en función del género y su relación con las habilidades de resolución de problemas. Para ello, los autores emplearon un modelo de crecimiento latente multivariante, utilizando datos recopilados de 86 estudiantes (48 hombres y 38 mujeres) que cursaban una asignatura de programación en un departamento de Tecnologías de la Educación y la Instrucción. Durante el estudio, se midieron sus conocimientos en los tres dominios de programación en tres momentos distintos del semestre, además de evaluar sus habilidades generales de resolución de problemas mediante el *Problem Solving Inventory, Form-A (PSI-A)* de Heppner y Petersen (1982).

Los resultados revelaron diferencias significativas entre los géneros. Se observó que los estudiantes varones mejoraron a mayor escala sus conocimientos conceptuales y estratégicos a lo largo del curso, mientras que las estudiantes femeninas demostraron un mejor desempeño en el conocimiento sintáctico y partieron de un nivel inicial más alto en esta área. Yurdugül y Aşkar (2013) afirmaron lo siguiente:

> The results demonstrated the significant differences in favor of male students in prior conceptual and strategic knowledge. Male students increased their conceptual knowledge scores (0.53; P≤0.05) and strategic knowledge scores (0.44; P≤0.05) more than female students during the programming course. Female students were more successful than males in their initial status and in development of syntactic programming knowledge. (p. 609)

Además, los autores encontraron una relación significativa entre la capacidad general de resolución de problemas y el género, indicando que los estudiantes varones poseían mayores habilidades en esta área en comparación con sus compañeras. En consecuencia, el desarrollo del conocimiento estratégico (definido como la capacidad de diseñar, codificar y probar un programa para resolver un problema) y del conocimiento conceptual a lo largo del tiempo, fue más favorable para los hombres.

Como posibles conclusiones, extraemos que, la brecha de género en programación no solo radica en el rendimiento, sino también en cómo se desarrolla el aprendizaje a lo largo del tiempo. Según estos resultados, sería necesario fomentar estrategias de enseñanza que ayuden a las mujeres a fortalecer su conocimiento conceptual y estratégico, así como impulsar su capacidad general de resolución de problemas.

2.2 Especialidad cursada en bachillerato

En varios estudios, la "especialidad cursada en bachillerato" de los encuestados es tratada como una variable muy relevante a la hora de determinar y proyectar el éxito que tendrán estos a lo largo de su carrera universitaria. Desde una perspectiva lógica, aquellos que optan por una especialidad en ciencias tienden a estar mejor preparados para carreras en áreas STEM, mientras que los que los estudiantes ciencias sociales se orientan hacía grados dentro del ámbito ADE. Si aplicamos esta línea de pensamiento a los grados híbridos, el análisis podría llegar a ser más incierto.

Arroyo-Barriguete et al (2023) llevaron a cabo un estudio comparativo en los grados de Administración de Empresas, en el que se analizó el efecto de asignaturas específicas, prestando especial atención a la asignatura de matemáticas, en grados híbridos o no STEM. Las muestras utilizadas provenían de dos universidades diferentes, con grupos que oscilaban entre 800 y 900 alumnos. Metodológicamente, el estudio combinó modelos de regresión con redes neuronales interpretables para garantizar la robustez de los resultados.

Los hallazgos revelaron un comportamiento idéntico en ambos grupos, indicando que los alumnos que cursaron el bachillerato de ciencias obtuvieron un rendimiento superior en matemáticas en comparación con sus compañeros de ciencias sociales. Aunque la asignatura de programación no fue objeto de análisis en el estudio, se puede argumentar que ésta está estrechamente relacionada con el razonamiento técnico y cuantitativo necesario para resolver problemas matemáticos, por lo que posiblemente los resultados en programación muestren el mismo patrón.

2.3 Relación Motivación-Rendimiento

En el contexto de la dificultad de la programación, diversos estudios han señalado que la automotivación de los estudiantes puede influir significativamente en su rendimiento y eficacia. La investigación realizada por Martínez-Mejía (2021) exploró los factores determinantes en el aprendizaje de la programación, con un enfoque particular en variables internas como el nivel de motivación y la autoeficacia percibida. El estudio se llevó a cabo con estudiantes de programación de la Facultad de Ciencias de Puerto Rico, empleando un diseño de investigación basado en encuestas.

Para el análisis de los datos, se utilizó el software SPSS, aplicando estadísticas descriptivas y un análisis de correlación de Spearman para evaluar la relación entre la motivación de los estudiantes y su rendimiento académico en programación. Los resultados evidenciaron que una actitud positiva hacia la asignatura puede tener un impacto favorable en el desempeño académico. En particular, se encontró una correlación de Spearman positiva y significativa entre los ítems de las subescalas de autoeficacia y motivación con el rendimiento en la asignatura de Fundamentos de Programación.

Estos hallazgos coinciden con lo señalado por Ruiz Dodobara (2005), quien sostiene que "un buen desempeño académico no puede ser garantizado únicamente por los conocimientos y habilidades del individuo. Las creencias de autoeficacia pueden marcar la diferencia en el rendimiento de dos personas con el mismo grado de habilidad" (p.10).

Esta teoría de la correlación entre la autoeficacia y el rendimiento académico de los alumnos es apoyada por suficientes autores, reconociendo la motivación de estos hasta como uno de los factores más importantes para ser exitoso: "Attitude towards the subject matter and SE are

among the most important factors in determining one's success in a particular field (Basak & Govender, 2015)."

El estudio titulado "Effect of Self-efficacy and Emotional Engagement on Introductory Programming Students", realizado por Kanaparan, Cullen y Mason en 2017, investigó cómo la autoeficacia y el compromiso emocional influyen en el rendimiento de estudiantes en cursos introductorios de programación.

La investigación se llevó a cabo en una universidad australiana y contó con la participación de 433 estudiantes. Se utilizó un enfoque cuantitativo para analizar la relación entre la autoeficacia, el compromiso emocional y el rendimiento académico en programación. Los hallazgos revelaron que existe una fuerte correlación entre la autoeficacia en programación y la satisfacción e interés de los estudiantes en el módulo, lo cual también se vinculó con su desempeño en el curso introductorio de programación. Específicamente, los estudiantes con mayor autoeficacia y compromiso emocional tendieron a obtener mejores resultados académicos.

Además, Govender et al. (2014) confirmó que "There is a strong link between SE[1] in problem solving and SE in computer programming, suggesting that students who have confidence in their problem-solving ability tend to perform better in computer programming tasks.

Por el contrario, algunos estudios cuestionan la existencia de una correlación entre la actitud y el rendimiento académico. Erodogan, Aydin y Kabaca sostienen que no hay una relación directa entre ambos factores, aunque reconocen que un alto nivel de autoconfianza (considerado una subescala de la motivación) sí contribuye a una mejora en los resultados de los exámenes. De manera similar, el estudio de Hongwarittorrn y Krairit (2015) no encontró una correlación estadísticamente significativa entre la actitud de los estudiantes hacia la informática y sus calificaciones. (Baser, 2013)

2.4 Relación Teoría-Práctica

Por otro lado, en el ámbito de la didáctica, la relación entre teoría y práctica ha sido un tema de debate constante. El aprendizaje en las aulas ha estado marcado por la dicotomía entre ambos componentes, un enfrentamiento que se remonta a las posturas opuestas de autores como Bacón y Descartes. A lo largo de los años, y a través de diversas teorías mediacionales, esta oposición entre la teoría y la práctica no ha encontrado una resolución definitiva (Perrupato, 2020). Esto genera dudas sobre la función y la integración de estas dos dimensiones dentro del entorno educativo.

Numerosos estudios defienden la necesidad de establecer una conexión más estrecha entre teoría y práctica; sin embargo, es evidente que persiste una brecha entre ambas. Surge la pregunta: ¿sería un mejor equilibrio entre estos dos componentes la clave para mejorar el rendimiento de los estudiantes? En particular, en el contexto de las ciencias puras y las asignaturas STEM, siempre ha prevalecido la convicción de que la práctica tiene un papel fundamental para el éxito en la evaluación de estas disciplinas.

Algunos estudios consideran los conceptos de *deep learning* y *surface learning* como nuevos enfoques de estudio y que influyen en el rendimiento académico y la adquisición de conocimientos. Un estudiante adopta un enfoque de *deep learning* cuando busca comprender los conceptos en profundidad, establecer conexiones entre diferentes ideas y aplicarlas en diversos escenarios. Este enfoque fomenta el pensamiento crítico y la resolución de problemas

[1] *Sef-efficacy*

desde una perspectiva práctica, lo que facilita la retención a largo plazo del conocimiento y su aplicación en situaciones novedosas (Marton & Säljö, 1976). Por otro lado, el *surface learning* se caracteriza por un aprendizaje basado en la memorización mecánica sin una comprensión profunda del contenido. Los estudiantes que adoptan este enfoque tienden a centrarse en la repetición de información para superar exámenes o cumplir con requisitos académicos, en lugar de desarrollar una comprensión significativa. Como resultado, su capacidad para aplicar el conocimiento en contextos diferentes suele ser limitada (Spada & Moneta, 2011).

El estudio de Fincher et al. (2006) entrevistó a 177 estudiantes de 11 universidades distintas con el objetivo de explorar la posibilidad de una correlación entre el rendimiento de los estudiantes en el primer curso de programación y los métodos que estos utilizaron para estudiar la asignatura. Se encontró que los estudiantes que obtuvieron calificaciones altas en el atributo de *deep learning* también lograron un alto rendimiento, lo que hace que el método de estudio sea un factor determinante a la hora de predecir el rendimiento. “We found a positive correlation between deep learning approaches and mark, and a negative correlation between surface learning approaches and mark” (Fincher et al., 2006, p.5).

Además, se encontró otra relación positiva entre la autoeficacia y la metodología de aprendizaje empleada por los estudiantes. En su estudio, Lindblom-Ylänne et al. (2018) afirmaron que los estudiantes con baja autoeficacia (SE), baja motivación para estudiar y creencias negativas sobre su capacidad para aprender tendían a utilizar el enfoque de *surface learning* como técnica de estudio, centrándose más en memorizar el código que en comprenderlo.

No obstante, existen diversas perspectivas sobre el enfoque más adecuado para abordar la asignatura de programación, especialmente en los primeros cursos. Varios autores coinciden en la necesidad de combinar ambas técnicas de estudio, ya que ni el aprendizaje memorístico ni el práctico por sí solos son suficientes para lograr un buen rendimiento. Konecki y Petrlic (2014) afirman que "la programación es más una habilidad que un conocimiento". El estudio de Ranjeeth (2011) encontró que el 50% de los estudiantes de programación informática en instituciones de educación superior tienden a adoptar un enfoque de *surface learning* en los cursos introductorios de programación. El estudio sugiere que los estudiantes adoptan este estilo de aprendizaje principalmente para cumplir con los requisitos del curso y obtener una calificación aprobatoria en las evaluaciones de programación.

Ante esta literatura discrepante, parece que la adopción de enfoques de *deep learning* y *surface learning* en la asignatura de programación no está completamente clara, lo que justifica la necesidad de un análisis más detallado sobre su influencia en el rendimiento de los estudiantes

2.5 Resumen de las hipótesis

Una vez acabada la revisión de la literatura, se ha optado por testear las siguientes hipótesis:

- ***Hipótesis 1 (H1):*** *Las mujeres perciben una dificultad mayor respecto a la asignatura de programación que los hombres*
- ***Hipótesis 2 (H2):*** *Los estudiantes que cursan bachillerato de ciencias obtienen un rendimiento mayor en la asignatura de programación en comparación con estudiantes de otras especialidades*
- ***Hipótesis 3 (H3):*** *Los estudiantes que afirman tener “habilidad” para la asignatura de programación obtienen, en términos generales, mayor rendimiento*

3. Material y Métodos

3.1 Descripción de la muestra

La base de datos utilizada en esta investigación se conformó a partir de los resultados de una encuesta ómnibus en formato papel, realizada entre el 21 de marzo y el 11 de abril de 2024 en la Universidad Pontificia Comillas. La muestra está integrada por estudiantes de dobles grados no STEM con componente de *Analytics* (E2+BA, E3+BA, E6+BA), pertenecientes a los cursos 3º, 4º y 5º.

La encuesta incluye un total de 40 preguntas que abordan estrategias de estudio en materias STEM, tales como el tiempo dedicado al estudio, la dificultad percibida y el rendimiento en asignaturas de matemáticas, estadística, programación y Machine Learning. Asimismo, se recopilaron variables generales como el género, la especialidad cursada en bachillerato y la nota de la EvAU. El contenido completo de la encuesta puede consultarse en el anexo del presente trabajo (capítulo 8).

Inicialmente, el conjunto de datos contaba con 152 observaciones y 48 variables. Para garantizar la calidad de la información, se llevó a cabo una primera fase de limpieza, en la que se eliminaron registros con valores nulos derivados de errores de transcripción.

En este proceso, se detectaron inconsistencias en la variable *EvAU*, ya que algunos encuestados utilizaron una escala de 0 a 14 mientras que otros emplearon una escala de 0 a 10. Debido a la imposibilidad de identificar con precisión la escala aplicada en cada caso, se optó por descartar esta variable para evitar sesgos en el análisis. Asimismo, en la variable relativa a la especialidad cursada en bachillerato, se registraron respuestas correspondientes a especialidades distintas de las ramas de Ciencias o Sociales (concretamente, *Bachillerato Internacional* con 7 observaciones y *Otras* con 5 observaciones). Dada la escasa representatividad en el conjunto de datos, se decidió excluir estas categorías, restando únicamente las especialidades de ciencias puras y ciencias sociales

Dado que el enfoque de este estudio se centra en las estrategias de estudio aplicadas a la asignatura de *Programación*, se eliminaron las preguntas relacionadas con otras materias STEM (matemáticas, estadística y Machine Learning). Sin embargo, se decidió conservar la pregunta relativa a la dificultad percibida en la programación de cada una de estas asignaturas, ya que resulta relevante para el posterior Análisis de Componentes Principales (PCA).

Por otro lado, las variables categóricas se transformaron en factores para asegurar un tratamiento adecuado durante el análisis. Asimismo, las variables *Consentimiento*, *Fecha* e *ID* fueron eliminadas, ya que no aportaban información relevante para el estudio.

Para abordar los valores faltantes, se empleó la técnica de imputación mediante el algoritmo *K-Nearest Neighbors* (KNN), configurado con $K = 3$. Esta metodología permitió identificar los tres vecinos más cercanos para cada observación con datos faltantes, calculando la media (para variables numéricas) o la moda (para variables categóricas) de dichos vecinos.

Por último, durante el análisis exploratorio se identificaron algunos valores atípicos. No obstante, dado que la escala con la que se está trabajando (0-10) permite cierta variabilidad en las respuestas, se decidió mantener estos registros ya que su inclusión no compromete la validez estadística de los resultados. Tras la depuración, la base de datos quedó conformada por un total de 139 observaciones y 23 variables.

3.2 Análisis de las variables independientes del modelo

Fruto de la revisión de la literatura, se concluye que existen múltiples factores que inciden directamente en el estudio de la asignatura de programación. Si bien muchas de las variables independientes que se incorporarán en los modelos de análisis no han sido objeto de estudio tan exhaustivo como otras (véase el tiempo relativo dedicado al estudio de programación, el grado de memorización o el propio grado universitario cursado) se consideran igualmente relevantes, ya que pueden aportar información valiosa y complementar el análisis desde diferentes perspectivas.

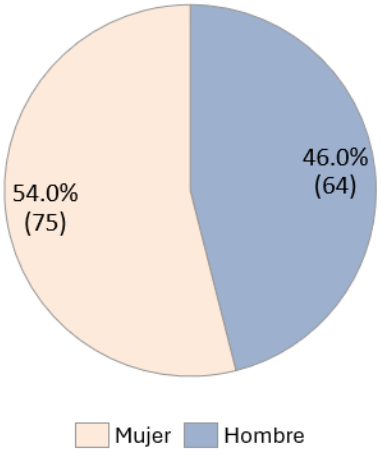
Aun así, estas variables se evaluarán de forma diferenciada en el momento de su inclusión, en función del contexto específico del modelo y de la relación causal que mantengan con la variable dependiente. Esta consideración permitirá adaptar su incorporación al análisis de manera coherente con los objetivos del estudio.

Variables	Tipo	Descripción
Género *(Genero)*	Categórica	Recoge el género de los participantes en la encuesta. Las opciones incluyen Hombre, Mujer y Otros
Grado *(Grado)*	Categórica	Representa el grado académico en el que se encuentra el participante. Las opciones que se presentan en la encuesta son: E2 + Analytics, E3 + Analytics y E6 + Analytics
Especialidad *(Especialidad)*	Categórica	Hace referencia a la especialidad cursada en bachillerato por el estudiante (ciencias puras o ciencias sociales)
Programacion_Memoria *(Memoria)*	Numérica discreta	Evalúa el grado en que los estudiantes memorizaron el contenido de las asignaturas de programación, siendo la escala de 0 a 10, donde 0 indica que no se memorizó nada y 10 que se memorizó todo
Programacion_Tiempo *(Tiempo)*	Numérica discreta	Mide el tiempo relativo que los estudiantes dedicaron a estudiar asignaturas de programación en comparación con otras asignaturas del grado, siendo la escala de 0 a 10, donde 0 significa que se dedicó mucho menos tiempo que a otras asignaturas y 10 que se dedicó mucho más tiempo
Programacion_ Teoría *(Teoria)*	Numérica discreta	Mide la estrategia de estudio empleada por los estudiantes al preparar asignaturas de programación, utilizando una escala de 0 a 10, donde 0 indica que no se apoyaron nada en la teoría para su estudio, y 10 que basaron completamente su preparación en contenidos teóricos
Motivo_Habilidad *(Habilidad)*	Numérica discreta	Recoge el motivo principal para elegir la carrera, indicando que el estudiante considera tener habilidad en las materias que se imparten

Motivo_Gustan *(Gustar)*	Numérica discreta	Refleja el motivo de elección de la carrera basado en el gusto por las materias involucradas

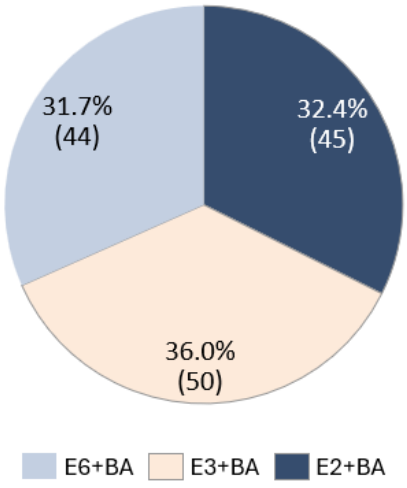
De todos los encuestados, el 54% son mujeres y el 46% son hombres, lo que refleja una muestra bastante equilibrada en términos de género. La distribución de los encuestados según el género se ilustra en la figura 1.

Figura 1: Distribución de las observaciones en función del género



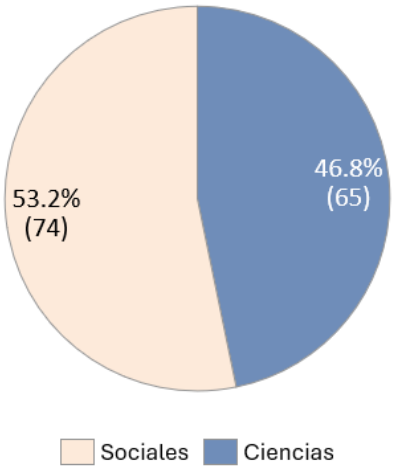
En cuanto al grado cursado por los participantes, también se observa una distribución equilibrada, aunque el grado de E3 + Analytics está ligeramente más representado en comparación con los demás. La figura 2 muestra la distribución detallada de las observaciones en función del grado cursado.

Figura 2: Distribución de las observaciones en función del grado



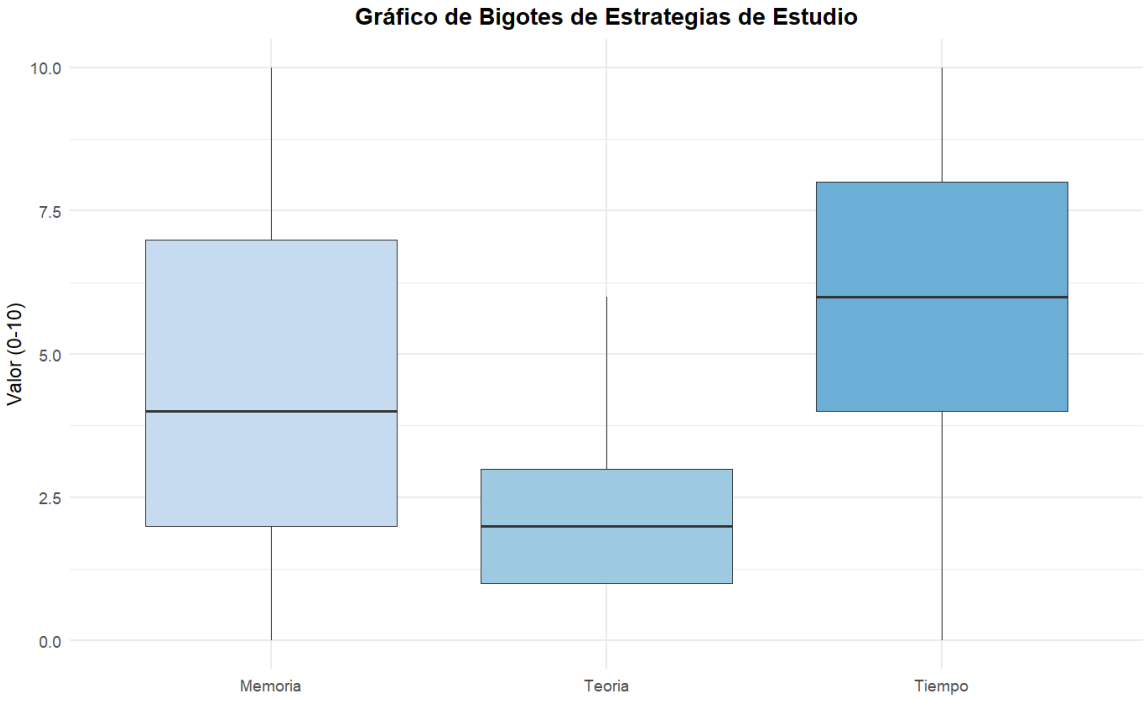
Respecto a la especialidad cursada en bachillerato, la especialidad de *Ciencias Sociales* (53%) está más representada que la de *Ciencias Puras* (46%) el conjunto de datos incluía también observaciones correspondientes al Bachillerato Internacional y a otras especialidades; sin embargo, tras el proceso de depuración, la distribución de observaciones quedó de la siguiente manera:

Figura 3: Distribución de las observaciones en función de la especialidad cursada en bachillerato



Las variables numéricas relativas al tiempo relativo dedicado a la programación, el grado de memorización y el uso de la teoría como enfoque metodológico se analizarán mediante gráficos de caja y bigotes:

Figura 4: Boxplot para las variables Memoria, Teoría y Tiempo



En cuanto al grado de memorización aplicado al estudio de programación, se observa una utilización moderada de esta estrategia entre los estudiantes, con una mediana situada en torno a 4. El rango intercuartílico (IQR), comprendido aproximadamente entre 2 y 7, indica una cierta dispersión en los enfoques de estudio. Además, la ligera simetría del gráfico indica que no hay un sesgo claro hacia puntuaciones muy bajas o altas.

Respecto al uso de la teoría como estrategia de estudio, los datos revelan una baja dependencia de este enfoque, con una mediana cercana a 2. El rango intercuartílico es reducido, abarcando valores entre 1 y 3, lo que muestra una baja dispersión y sugiere que la mayoría de los estudiantes no basa su aprendizaje principalmente en contenidos teóricos.

En lo que respecta al tiempo dedicado al estudio de programación en comparación con otras asignaturas, los datos muestran que la mayoría de los estudiantes invierte una cantidad considerable de tiempo, con una mediana situada en torno a 6. El rango intercuartílico (IQR) se extiende aproximadamente entre los valores 4 y 8, lo que indica una variabilidad moderada en el tiempo relativo empleado.

3.3 Formulación de los modelos

En el contexto de este trabajo se ha optado por utilizar modelos de regresión como principal herramienta de análisis, con el fin de estudiar las relaciones existentes entre las variables recogidas y cuantificar el efecto que estas tienen sobre la variable dependiente, identificando qué relaciones son estadísticamente significativas. Así, la regresión permite cuantificar cómo influye cada variable independiente en la variable dependiente, e identifica las relaciones significativas. Aunque este tipo de modelos también puede emplearse con fines predictivos, en este caso se ha adoptado un enfoque explicativo, centrado en obtener conclusiones claras y comparables.

Concretamente, cada variable dependiente ha sido analizada mediante una regresión OLS (mínimos cuadrados ordinarios) y un árbol de regresión. Esta combinación permite, por un lado, estimar el impacto individual de cada factor bajo supuestos lineales (OLS), y por otro, explorar posibles interacciones no lineales o jerárquicas entre variables (árbol), lo que enriquece el análisis y aporta una visualización más intuitiva de los patrones detectados en los datos.

3.3.1 Análisis de la variable dependiente del modelo - Dif_PC1

Para el análisis del primero modelo, como variable dependiente se ha escogido una medida agregada que representa la dificultad percibida en programación por parte de los encuestados “Dif_PC1”. La variable Dif_PC1 surge a partir de la reducción dimensional de cuatro variables seleccionadas que reflejan la dificultad percibida en programación en distintos lenguajes. El proceso y el razonamiento utilizado para el mismo se explican a continuación.

3.3.1.1Explicación Análisis Componentes Principales (PCA)

Como veníamos explicando anteriormente, durante el proceso de depuración de información se conservaron ciertas variables relacionadas con asignaturas de estadística, matemáticas y Machine Learning. Aunque estas variables no están directamente vinculadas a la asignatura de programación, se consideraron relevantes para el análisis debido a la información que aportan sobre la dificultad percibida en tareas de programación en diferentes contextos, en este caso en diferentes lenguajes. A continuación, se describen las variables seleccionadas:

Matematicas_P	mide la dificultad percibida por el estudiante en la parte de programación (MATLAB) de la asignatura de matemáticas
Estadistica_P	mide la dificultad percibida por el estudiante en la parte de programación (R) de la asignatura de estadística y probabilidad
MachineLearning_P	mide la dificultad percibida por el estudiante en la parte de programación (Python/R) de la asignatura de Machine Learning

Estas variables, junto con la variable "*Dif_Percibida*", nos aportan información relevante acerca de la dificultad percibida por los estudiantes a la hora de programar. La decisión de conservarlas se fundamentó en la realización posterior de una reducción de dimensionalidad mediante un Análisis de Componentes Principales (PCA). Esta técnica permite resumir la varianza capturando la mayor cantidad de información relevante y reduciendo el ruido y la redundancia.

Comenzamos calculando el índice KMO para medir la adecuación muestral para el PCA. Este da un resultado global de *0.77*, lo cual es suficientemente alto para justificar el uso del PCA.

Realizamos el Análisis de Componentes Principales y la varianza queda explicada por cada componente principal de la siguiente manera:

	Dimensión 1	**Dimensión 2**	**Dimensión 3**	**Dimensión 4**
Varianza	**2,51**	0,65	0,48	0,36
% Varianza	**62,8%**	16,2%	12,0%	9%

Figura 5: Varianza explicada del PCA (eigenvalues)

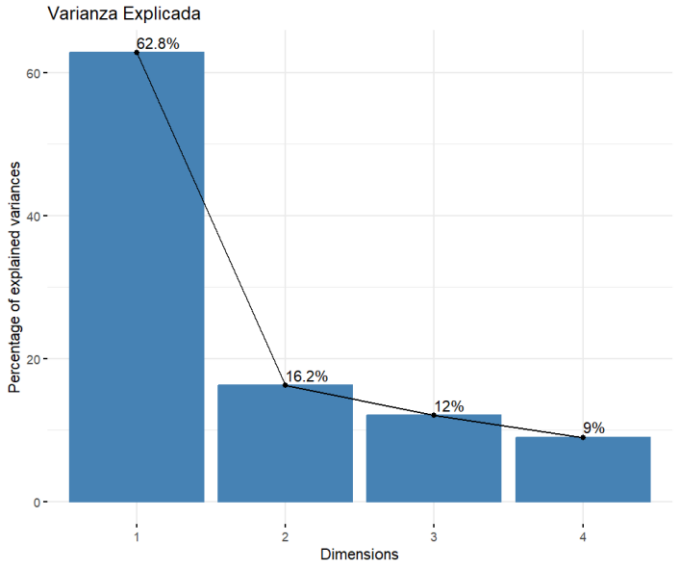
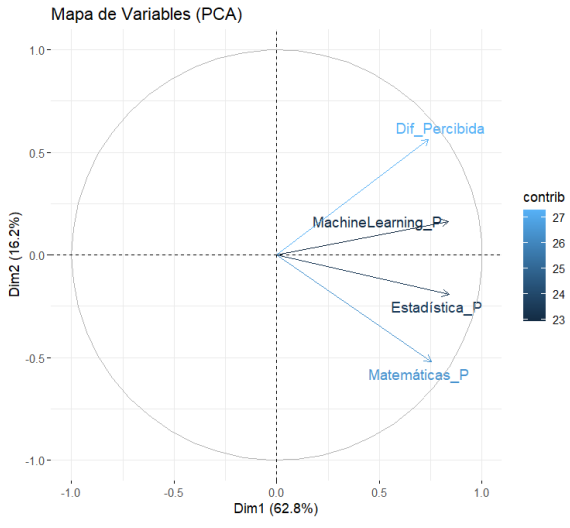


Figura 6: Biplot variables PCA



Como se observa en el gráfico, la mayor parte de la varianza (62.8%) queda explicada por la primera dimensión (Dim.1), la cual está fuertemente influenciada por la combinación de las cuatro variables consideradas: *Matemáticas_P, Estadística_P, MachineLearning_P* y

Dif_Percibida. Dado que esta componente principal otorga un peso similar a todas las variables y captura una proporción significativa de la varianza total, podemos utilizarla como variable dependiente en los modelos posteriores. Esta decisión permite sintetizar en un solo indicador la percepción general de dificultad en programación. Al tratarse de una variable continúa derivada de una combinación lineal estandarizada de las variables originales, Dif_PC1 se convierte en una medida más robusta y sintética para analizar la dificultad percibida, evitando problemas de multicolinealidad y capturando una dimensión latente común entre los ítems considerados.

Figura 7: Histograma variable Dif_PC1

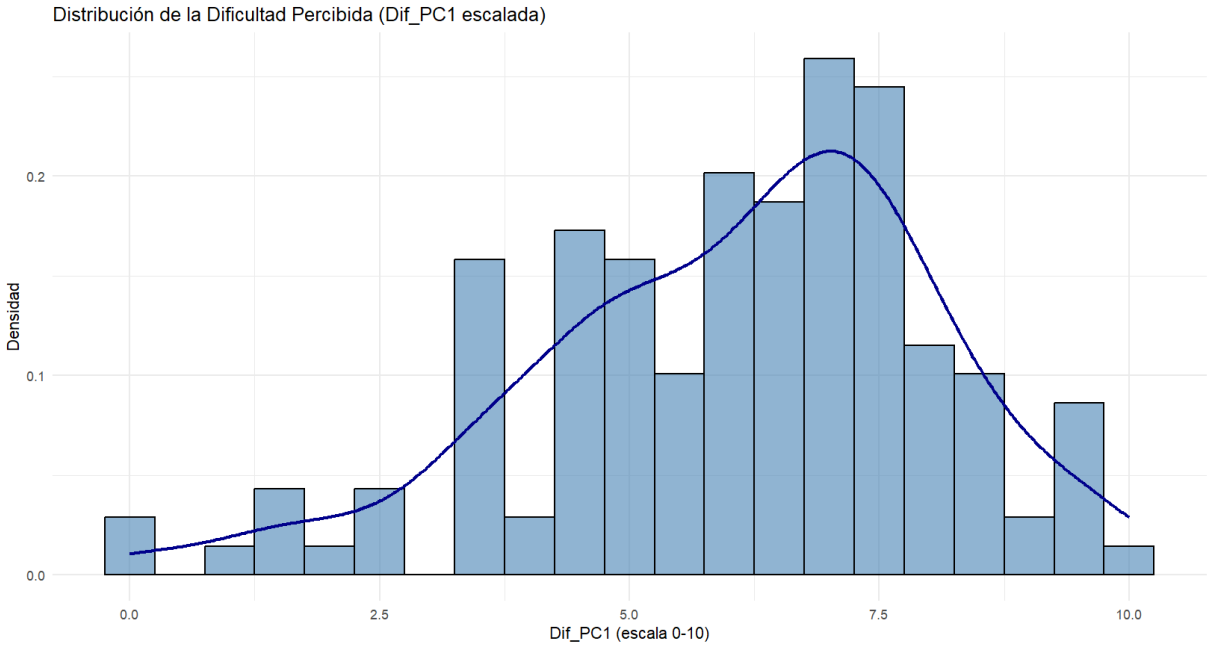
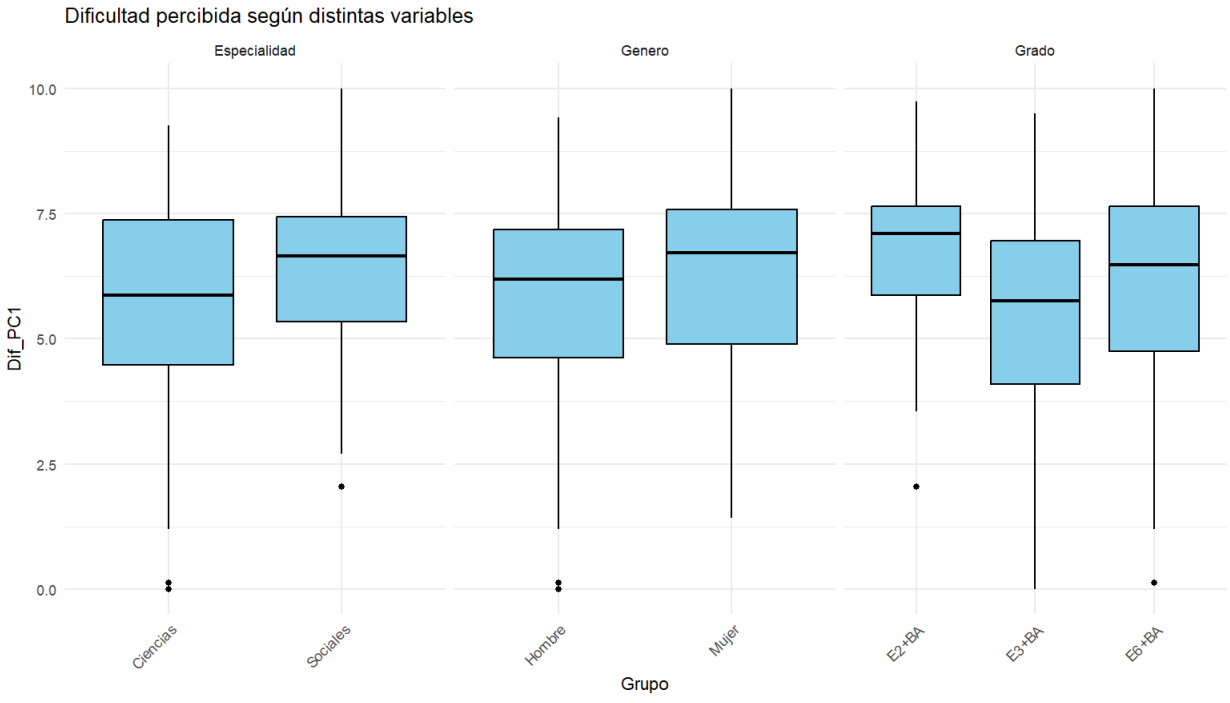


Figura 8: Boxplot combinado para variable Dif_PC1



Atendiendo a los gráficos, la mayoría de los estudiantes perciben una dificultad moderada-alta en programación. Haciendo un análisis preliminar, parece que esta dificultad se acentúa para algunos colectivos. Por ejemplo, parece que las mujeres, los estudiantes de E2+BA o los del bachillerato de ciencias sociales perciben mayor dificultad en la asignatura de programación. Estas diferencias serán contrastadas posteriormente mediante técnicas de regresión (OLS + árbol de regresión)

Como resumen de lo que se hará después, se ofrecen la siguiente tabla:

Modelo	Variable Dependiente	Variables Independientes	Tipo del Modelo	Objetivo
1: Regresión OLS 1	*Dif_PC1*	*Género, Grado, Especialidad, Gustar, Habilidad, Memoria, Teoría*	Lineal-Aprendizaje supervisado	Entender los factores que más afectan a la percepción de la dificultad
2: Árbol de Regresión 1	*Dif_PC1*	*Género, Grado, Especialidad, Gustar, Habilidad, Memoria, Teoría*	No lineal-Aprendizaje supervisado	Explorar patrones y jerarquías en la dificultad percibida

3.3.2 Análisis de la variable dependiente del modelo - Rendimiento

La segunda variable dependiente es *"Rendimiento"* que trata de representar la calificación obtenida por el estudiante en la asignatura de programación. Esta variable es de especial interés porque permite estudiar el desempeño académico real en contraste con otras variables más subjetivas

Figura 9: Histograma para variable Rendimiento

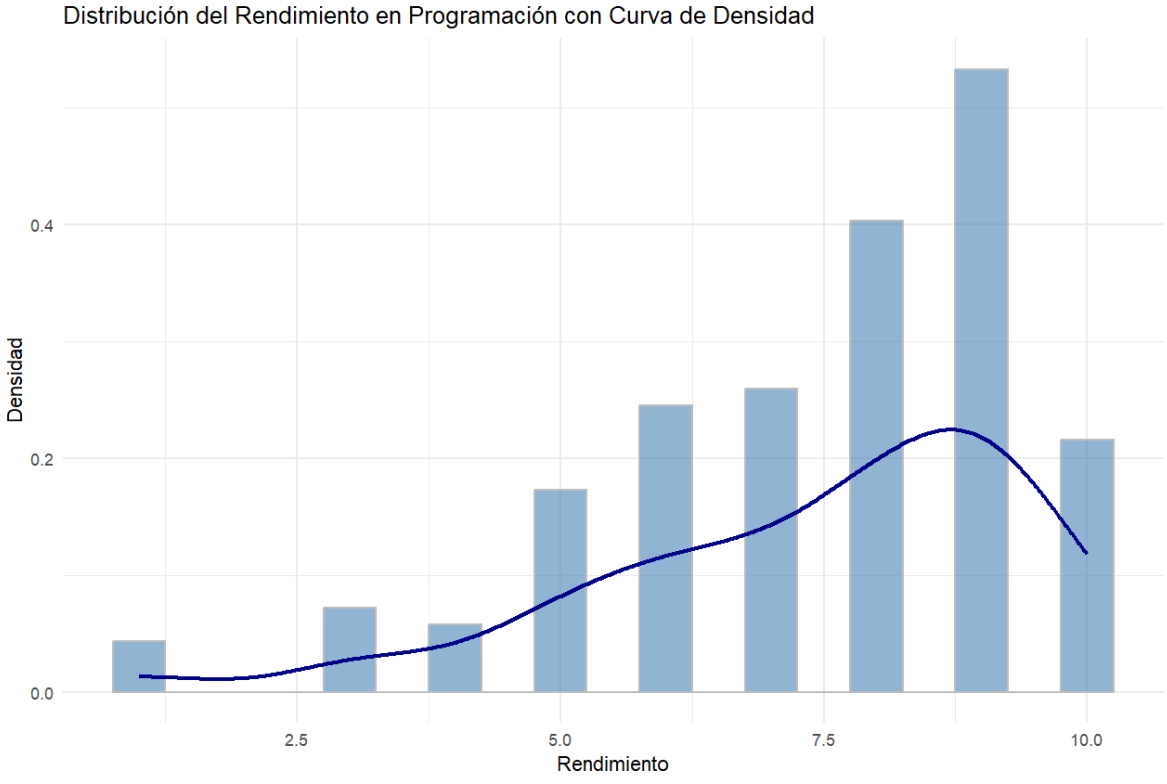
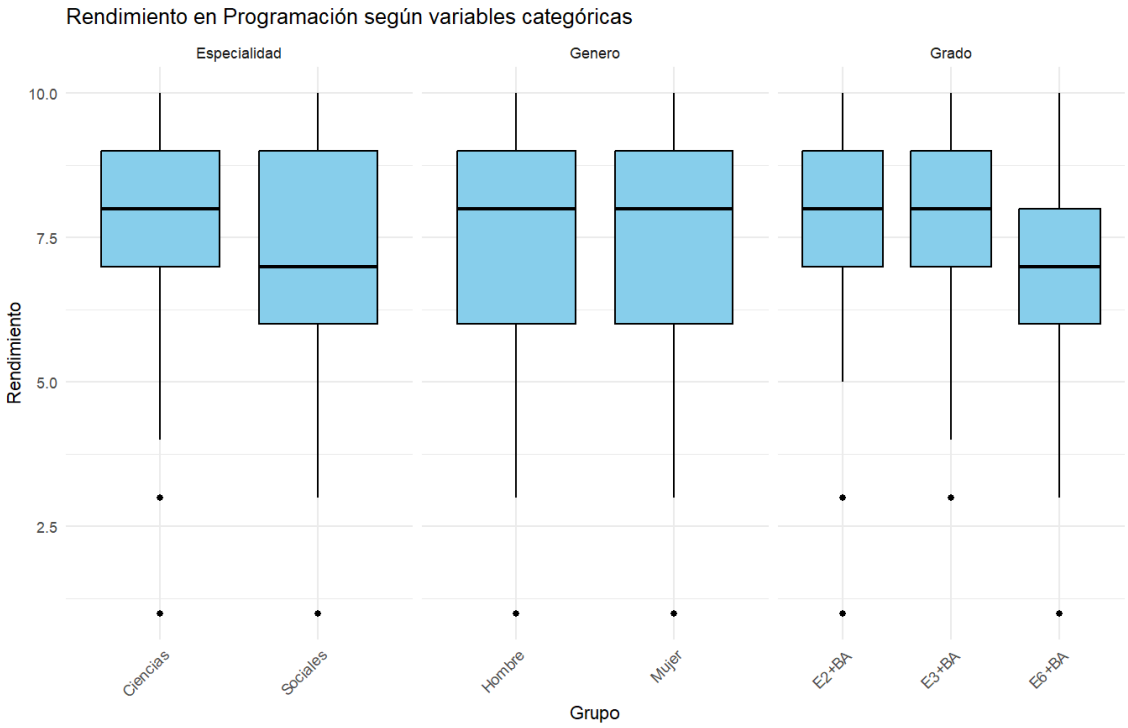


Figura 10: Boxplot combinado para variable Rendimiento



En este caso, podemos observar que la mayoría de los estudiantes obtiene un rendimiento medio-alto en programación, con una concentración notable de calificaciones entre 7 y 9. De nuevo, a primera vista se identifican diferencias en el rendimiento académico entre distintos grupos de estudiantes. Por ejemplo, los alumnos provenientes del bachillerato de ciencias parecen obtener mejores resultados que los de ciencias sociales. Asimismo, se aprecian ligeras variaciones entre géneros y entre los diferentes grados, destacando en particular un rendimiento algo inferior en el grupo de E6+BA.

Estas diferencias, aunque descriptivas en esta fase, serán contrastadas posteriormente mediante técnicas de regresión lineal (OLS) y árboles de decisión. A modo ilustrativo de lo que se hará después, se ofrecen la siguiente tabla:

Modelo	**Variable Dependiente**	**Variables Independientes**	**Tipo del Modelo**	**Objetivo**
3: Regresión OLS 2	*Rendimiento*	*Género, Grado, Especialidad, Gustar, Habilidad, Memoria, Tiempo, Teoría*	Lineal-Aprendizaje supervisado	Entender los factores que más afectan al rendimiento en programación
4: Árbol de Decisión 2	*Rendimiento*	*Género, Grado, Especialidad, Gustar, Habilidad, Memoria, Tiempo*	No lineal-Aprendizaje supervisado	Explorar patrones y jerarquías en el rendimiento en programación

Una vez formulados los modelos, procedemos a su interpretación.

4. Resultados y Discusión

4.1 Variable Dependiente Dificultad Percibida en Programación (Dif_PC1)

4.1.1 MODELO 1- Regresión OLS 1

En las tablas 1 y 2 se muestran los resultados del modelo de regresión lineal (OLS) para la variable dependiente *Dif_PC1* (obtenida a partir de un Análisis de Componentes Principales).

Tabla 1: Resultados del Modelo de Regresión OLS 1 - Dif_PC1

Variable	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.188966	0.455800	-0.415	0.679132
GeneroMujer	0.932440	0.261226	3.569	0.000502
GradoE3+BA	-1.161327	0.314657	-3.691	0.000328
GradoE6+BA	-0.909590	0.316496	-2.874	0.004737
EspecialidadSociales	0.408627	0.249715	1.636	0.104180
Gustar1	-0.314929	0.266738	-1.181	0.239891
Habilidad1	-0.392729	0.267849	-1.466	0.145000
Memoria	0.128418	0.047098	2.727	0.007283
Teoria	-0.002978	0.126795	-0.023	0.981301

Tabla 2: Resultados Globales del Modelo de Regresión OLS 1 - Dif_PC1

R^2 (Multiple R-squared)	**0.2569**
Adjusted R^2	0.2112
F-statistic	5.618 (8 y 130 DF), p-value= 4.031e-06

Los resultados muestran que:

1) El género es una variable altamente significativa ($p < 0.001$). El coeficiente positivo asociado a *GeneroMujer* indica que las mujeres tienden a percibir la asignatura como más difícil que los hombres, manteniendo constantes el resto de las variables.
2) El tipo de grado también resulta clave: los estudiantes de los dobles grados E3+BA y Grado E6+BA tienen coeficientes negativos y significativos, lo que implica que sus estudiantes perciben una menor dificultad que los de E2 + BA.
3) La variable Memoria es significativa ($p = 0.007$), y su coeficiente positivo indica que, a mayor memorización de los contenidos de programación, mayor dificultad percibida.
4) El resto de las variables, como la especialidad de origen (Sociales), el gusto por la programación, la habilidad percibida o la orientación teórica, no alcanzan niveles de significación estadística.

En conjunto, el modelo presenta un R^2 del 25,7%, quedando mucha variabilidad sin explicar lo que es normal en ciencias sociales o educativas, donde influyen factores difíciles de observar o medir. Además, el estadístico F ($p < 0.001$) confirma que el modelo es significativo en su conjunto, es decir, que al menos una de las variables incluidas contribuye de forma relevante a explicar la dificultad percibida en programación.

Como conclusiones se observa que las variables que más influyen en la dificultad percibida en la asignatura de programación son el género del estudiante (mujer), el grado que cursa (E3 + BA y E6 + BA) y el uso de la memoria como estrategia de estudio.

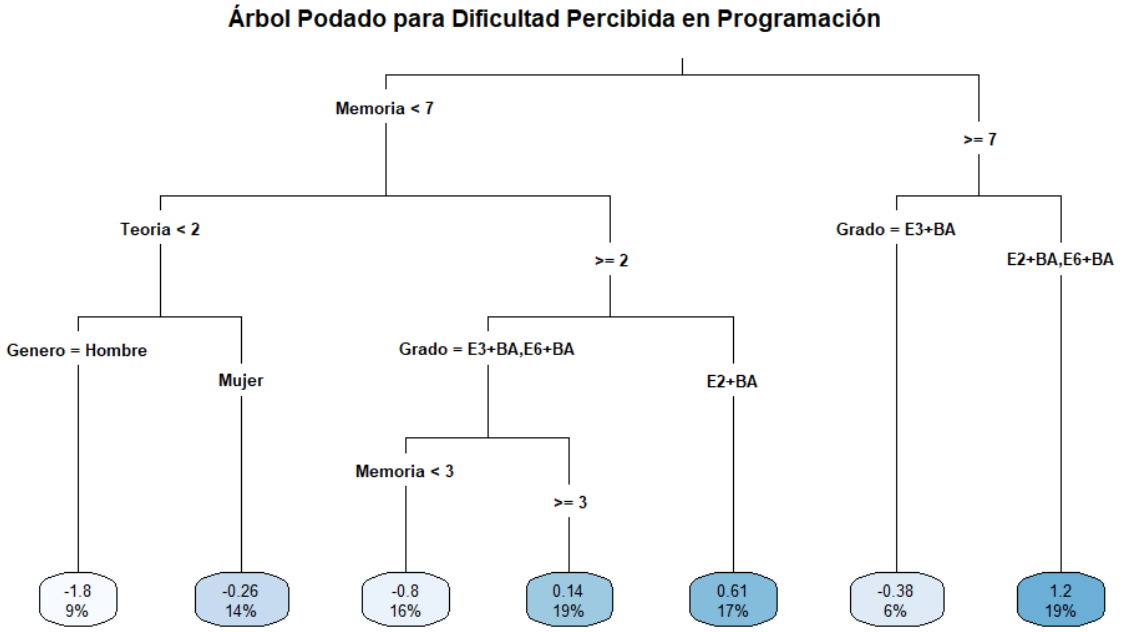
Con esto, y teniendo en cuenta el coeficiente positivo y la significación de la variable *GeneroMujer* en la regresión OLS puede confirmarse la **Hipótesis 1 (H1)**: *Las mujeres perciben una mayor dificultad en la asignatura de programación que los hombres*.

Con el objetivo de explorar posibles interacciones no lineales y reforzar conclusiones, analizamos el segundo modelo basado en árboles de regresión.

4.1.2 MODELO 2- Árbol de Regresión 1

En la *figura 11* se muestra el árbol de regresión correspondiente a la misma variable dependiente *Dif_PC1*. Este árbol fue podado utilizando el parámetro de complejidad (cp) con el fin de evitar el sobreajuste y mejorar su capacidad de generalización. El modelo resultante cuenta con siete hojas, cada una de las cuales representa un grupo de estudiantes con características similares. En cada hoja se indica la media de dificultad percibida y el porcentaje de estudiantes que pertenecen a ese grupo, donde los valores positivos reflejan una mayor dificultad percibida, mientras que los valores negativos indican una menor percepción de dificultad.

Figura 11: Árbol de Regresión para la variable dependiente Dif_PC1



El diagrama del árbol muestra que la variable más importante del modelo es *Memoria*, ya que aparece en la raíz del árbol. A partir de este punto, se ramifican otras variables como *Teoría, Género, Grado y Especialidad.* Al igual que la regresión OLS, el árbol permite identificar perfiles de estudiantes que tienden a experimentar más o menos dificultad en función de variables académicas y personales.

Generalmente, aquellos que basan su estudio en la memorización de la asignatura, tienden a percibir una mayor dificultad. Además, al analizar las ramas del árbol podemos observar otras interacciones relevantes:

1) Los estudiantes del grado E2+BA, independientemente del grado de memorización que apliquen al estudiar programación, tienden a percibir esta asignatura como más difícil.

En contraste, quienes cursan E3+BA muestran niveles notablemente más bajos de dificultad percibida. Esta diferencia puede explicarse a partir del contenido curricular de los grados: mientras que E3+BA combina Business Analytics con Derecho (mucha carga memorística), E2+BA se basa en Administración y Dirección de Empresas, y E6+BA en Relaciones Internacionales, donde la naturaleza del aprendizaje es distinta. Esto hace compresible el hecho de que los alumnos de E3+BA que utilizan estrategias de estudio basadas en la memorización cuando estudian programación reporten una experiencia más llevadera. Ellos están acostumbrados a preparar asignaturas teóricas a partir del aprendizaje literal, por lo que memorizar sintaxis o código no les supone un obstáculo significativo, mientras que para otros alumnos que no estén acostumbrados a ello puede ser una tarea mucho más tediosa.

2) Aquellos estudiantes que declaran un nivel bajo en el uso de teoría al estudiar tienden a percibir menos dificultad.
3) Los hombres que no se basan nada en la teoría y probablemente utilicen el método de "*deep learning*" son los que menos dificultad percibida reportan del modelo. Las mujeres, dentro de este grupo, también perciben poca dificultad, aunque más que los hombres.

4.2 Variable Dependiente Rendimiento en Programación (Rendimiento)

4.2.1 MODELO 3- Regresión OLS

En la tabla 3 se muestran los coeficientes estimados del modelo de regresión lineal para la variable dependiente *Rendimiento*. A diferencia del modelo anterior, en este caso se ha incluido la variable independiente *Tiempo*, ya que se considera que el tiempo dedicado al estudio de programación puede influir directamente en el rendimiento académico. Esta variable no se incluyó en el modelo anterior (dificultad percibida) porque se entiende que entre el tiempo de estudio y la dificultad percibida puede existir una relación causal bidireccional, lo que dificultaría su interpretación como predictor en un modelo explicativo.

Tabla 3: Resultados del Modelo de Regresión OLS 2-Rendimiento

Variable	***Estimate***	***Std. Error***	***t value***	***Pr(>|t|)***
(Intercept)	*8.550388*	*0.733441*	*11.658*	*<2e-16*
GeneroMujer	*-0.050520*	*0.374299*	*-0.135*	*0.8928*
GradoE3+BA	*-0.299010*	*0.495468*	*-0.603*	*0.5472*
GradoE6+BA	*-0.715606*	*0.440485*	*-1.625*	*0.1067*
EspecialidadSociales	*-0.561533*	*0.355097*	*-1.581*	*0.1162*
Gustar1	*-0.111397*	*0.363952*	*-0.306*	*0.76*
Habilidad1	*0.423517*	*0.375328*	*1.128*	*0.2612*
Memoria	*-0.136344*	*0.063240*	*-2.156*	*0.0329*
Tiempo	*-0.005428*	*0.079745*	*-0.068*	*0.9458*

Tabla 4: Resultados Globales del Modelo de Regresión OLS 2-Rendimiento

R^2 (Multiple R-squared)	**0.1185**
Adjusted R^2	0.0642
F-statistic	2.184 (8 y 130 DF), p-value = 0.0326

De las variables incluidas en el modelo, ha resultado estadísticamente significativa solo una de ellas: *Memoria.* Con un coeficiente de -0.136, queda reflejado que, a mayor uso de la memoria a la hora de estudiar la asignatura, menor rendimiento percibido. Este resultado sugiere que, al igual que en el caso de la dificultad, los estudiantes que abordan la programación con una estrategia memorística obtienen peores resultados.

En cuanto al resto de variables, parece que género, grado, especialidad, gusto, habilidad, tiempo de dedicación y teoría por si solas no explican de forma consistente el rendimiento en programación. Esto concuerda con la tabla 4 de resultados globales del modelo 2, ya que el mismo presenta un ajuste limitado y, aunque es significativo a nivel global, su capacidad para explicar el rendimiento es baja.

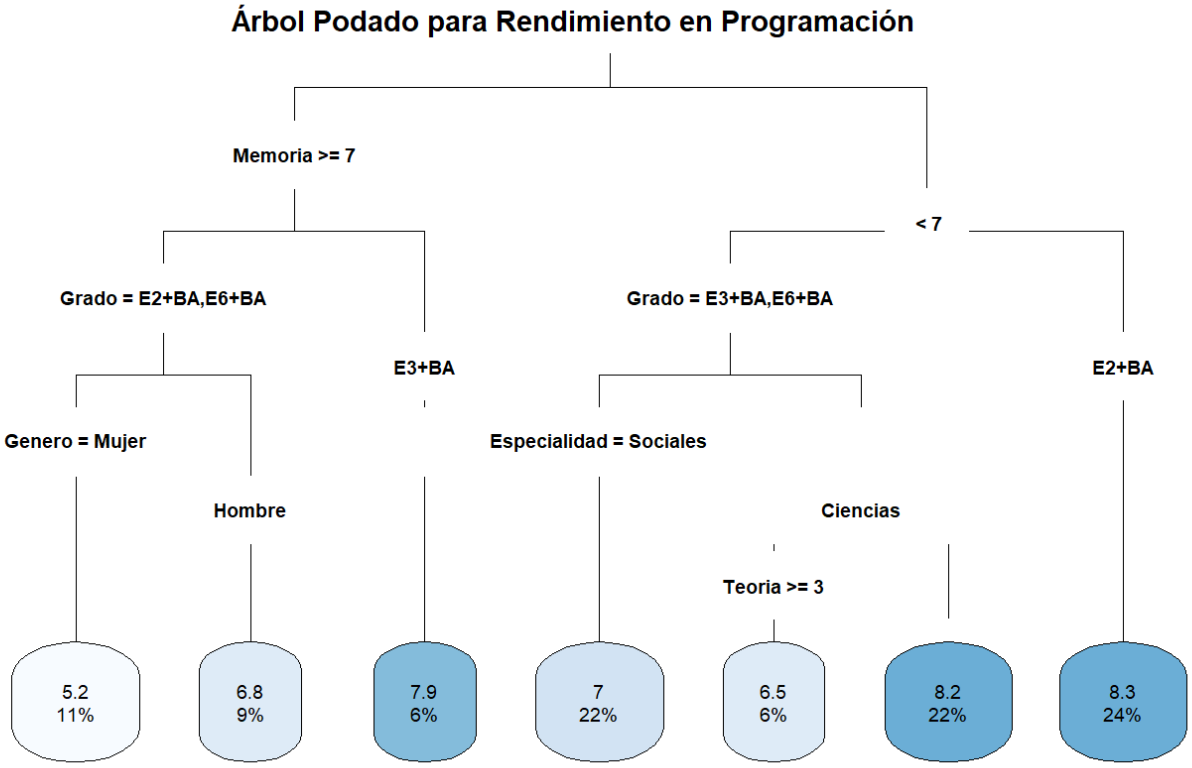
Por tanto, con base en los resultados obtenidos, las hipótesis H2 y H3 deben ser rechazadas, ya que no se han encontrado efectos estadísticamente significativos de la especialidad ni de las variables actitudinales sobre el rendimiento en programación.

- Hipótesis 2 (H2): *Los estudiantes que cursan bachillerato de ciencias obtienen un rendimiento mayor en la asignatura de programación en comparación con estudiantes de otras especialidades* → Rechazada, dado que la variable *Especialidad* no resultó significativa en el modelo OLS.
- Hipótesis 3 (H3): *Los estudiantes que afirman tener "habilidad" para la asignatura de programación obtienen, en términos generales, mayor rendimiento* → Rechazada, ya que la variable *Habilidad1* tampoco mostró un efecto significativo en el modelo.

Para obtener unas conclusiones más amplias, analizamos el árbol de regresión.

4.2.2 MODELO 4- Árbol de Regresión 2

Figura 12: Árbol de Regresión para la variable dependiente Rendimiento



Este árbol también fue podado utilizando el parámetro de complejidad (cp) para evitar sobreajuste y aportar claridad. De nuevo, la primera división del árbol se produce en torno a la variable *Memoria*, lo cual resulta coherente con los resultados obtenidos en la regresión OLS, donde fue la única variable estadísticamente significativa. En este sentido, se observa que los estudiantes que dependen menos de la memoria para estudiar programación (*Memoria* < 7) tienden, en general, a obtener un rendimiento más alto en la asignatura.

Además, si profundizamos en las ramas del árbol, podemos observar otras interacciones relevantes

1) En niveles estándares o bajos de memoria, los estudiantes del grado E2+BA tienden a obtener un rendimiento alto en programación (8.3) en comparación con sus compañeros de otros grados. En cambio, estudiantes del mismo grado con alta dependencia de la memoria obtienen el peor resultado (5.2), especialmente si son mujeres.
2) Los estudiantes del grado E3+BA muestran un rendimiento medio-alto independientemente de la estrategia de estudio. De hecho, los estudiantes de estos grados obtienen mayor rendimiento cuando estudian de memoria la asignatura. De nuevo, se observa como el nivel de memorización y el tipo de grado interactúan: mientras que en la mayoría de los casos estudiar "de memoria" se asocia a peores resultados, los alumnos de E3+BA vuelven a destacar cuando aplican este estilo de aprendizaje, obteniendo en este caso un rendimiento claramente superior.
3) En cuanto a la especialidad cursada en bachillerato, no se observan diferencias marcadas en el rendimiento. Ambos grupos presentan promedios similares: Sociales *(7.0)* y Ciencias *(entre 6.5 y 8.2)*, dependiendo del nivel de teoría

5. Conclusiones

Este trabajo ha tenido como objetivo principal entender cómo se percibe y afronta la asignatura de programación en el contexto de grados no STEM o grados híbridos. A partir de este enfoque, se recogió información detallada sobre las estrategias de estudio empleadas por los alumnos, con el fin de identificar patrones que permitan mejorar los métodos de aprendizaje y promover técnicas que se adapten de forma más eficaz a las características de cada grupo. Todo ello con el objetivo de optimizar tanto la comprensión de los contenidos como el rendimiento académico en la asignatura.

5.1 Sobre la Dificultad Percibida

Los análisis realizados en el trabajo permiten concluir que la dificultad percibida en programación no es homogénea entre el alumnado. Más allá de la variabilidad de los datos, se ha demostrado que variables como el género, el grado o el uso de la memoria como método de estudio son factores claves para explicar esta percepción.

Las mujeres, por ejemplo, tienden a percibir la asignatura de programación como más difícil. En un primer momento, esta diferencia podría atribuirse a factores estructurales, como una menor exposición previa a contenidos tecnológicos, sesgos de autoeficacia, o entornos de aprendizaje menos inclusivos. No obstante, los resultados muestran que, a pesar de esta percepción, las mujeres obtienen un rendimiento similar al de sus compañeros varones. En este sentido, la mayor dificultad percibida entre las mujeres parece estar relacionada no con un menor desempeño real, sino con factores psicológicos como la inseguridad académica, la autoexigencia o la falta de motivación ante el reto.

Por otro lado, se ha observado que los estudiantes del grado E2+BA perciben la asignatura de programación como más difícil que sus compañeros de otros grados. Esta diferencia no parece deberse únicamente al contenido del grado, sino también al perfil del alumnado que lo cursa, en términos de intereses, habilidades previas y formación académica. Aunque la variable

Especialidad no ha sido significativa desde el punto de vista estadístico, se ha quedado cerca de serlo (P-valor= 0.10) y los datos muestran una tendencia clara: mientras que muchos de los estudiantes de E3+BA (56 %) y E6+BA (44 %) vienen del bachillerato de Ciencias, en el caso de E2+BA esta proporción baja al 40 %, siendo mayoría los que proceden de Ciencias Sociales. Esto puede explicar, al menos en parte, por qué este grupo percibe la programación como más compleja: han estado menos expuestos a contenidos lógico-matemáticos en su etapa previa y, probablemente, afrontan la asignatura con menos herramientas y mayor inseguridad, lo que incrementa su percepción de dificultad.

5.2 Sobre el Rendimiento

En cuanto al rendimiento académico en programación, los resultados han sido menos contundentes que en el caso de la dificultad percibida. El modelo de regresión OLS ha mostrado un ajuste bastante limitado, con un R^2 del 11,85%, lo que indica que las variables analizadas explican solo una pequeña parte del rendimiento. Aun así, hay un patrón que se repite tanto en la regresión como en el árbol de decisión: la variable Memoria es la única que aparece como significativa.

De forma consistente, los estudiantes que se basan menos en la memorización para estudiar tienden a obtener mejores resultados. Esto refuerza la idea de que la programación no se aprende simplemente memorizando fórmulas o sintaxis, sino que requiere comprensión, práctica y razonamiento lógico. La literatura aquí fue clara, expresando que los estudiantes que aplican estrategias de tipo *deep learning* centradas la conexión de ideas y la aplicación práctica tienden a alcanzar un rendimiento significativamente más alto.

También se han observado algunas combinaciones interesantes en el árbol de regresión. Por ejemplo, los estudiantes con baja puntuación en memoria, que dedican poco tiempo al estudio, pero vienen de especialidades técnicas y con baja carga teórica, alcanzan los rendimientos más altos. Esto sugiere que no siempre se trata de estudiar más, sino de estudiar mejor y con un enfoque adecuado a la materia.

Por el contrario, variables como la especialidad cursada (Ciencias o Sociales) o la autopercepción de habilidad no han mostrado relación significativa con el rendimiento. Esto es relevante ya que contradice ciertas creencias comunes: venir del bachillerato de Ciencias no garantiza obtener mejores resultados, y creerse “bueno” en programación no siempre se traduce en un buen desempeño. De hecho, estos resultados entran en tensión con buena parte de la literatura revisada, donde en ambos casos se había identificado una correlación positiva y significativa entre estas variables y el rendimiento. En este estudio, sin embargo, dichos factores no parecen tener un peso real cuando se analizan junto al resto de variables, lo que invita a replantear algunas suposiciones comunes sobre los perfiles de éxito en esta asignatura.

Así, a pesar del bajo R^2 del modelo, los perfiles extremos identificados en el árbol de regresión (por ejemplo, estudiantes con baja memoria, baja teoría y poco tiempo de estudio que rinden muy bien) nos invitan a reflexionar sobre la eficiencia de ciertas estrategias de aprendizaje, más que sobre la cantidad de dedicación.

5.3 Conclusiones finales y aplicación práctica

Este trabajo ha permitido explorar cómo estudiantes de grados no puramente técnicos, como los programas híbridos en *Business Analytics*, abordan el aprendizaje de la asignatura de programación. Una vez concluido el análisis, se pueden proponer algunas recomendaciones que guíen a los alumnos. Estas conclusiones no se plantean como una fórmula única, pero si pretender ofrecer un entendimiento global sobre las estrategias de estudio que utilizan los estudiantes en estos contextos.

Además, estos *insights* también pueden ser de gran utilidad para el profesorado, ya que permiten entender mejor cómo viven los estudiantes esta asignatura y ofrecen claves para adaptar las metodologías docentes, orientándolas hacia enfoques que resulten más accesibles y motivadores.

En primer lugar, se ha llegado a la conclusión de que las estrategias de estudio centradas en la memorización se asocian con una mayor percepción de dificultad y con un peor rendimiento académico. Esto es lógico ya que, asignaturas como la programación donde la comprensión conceptual y la práctica son claves, la estrategia de estudio y el enfoque pueden marcar la diferencia. De este modo, resulta mucho más eficaz aplicar enfoques activos: resolver ejercicios, probar código, analizar errores y buscar la lógica detrás de los resultados parecen presentarse como los caminos más eficaces. Ahora bien, es importante resaltar que esta generalización no aplica a los estudiantes de E3 + BA, ya que ha quedado probado que, en su caso, estudiar de memoria no tiene un efecto negativo en su rendimiento académico. De hecho, en algunos escenarios les beneficia. Por tanto, aunque las recomendaciones generales se orientan a un enfoque más práctico y de entendimiento real, los estudiantes de Derecho podrían beneficiarse igualmente del estudio memorístico.

Además, se han encontrado divergencias en la percepción de dificultad dependiendo del perfil del estudiante. Aquellos estudiantes procedentes de itinerarios menos técnicos (bachillerato de Ciencias Sociales en este caso), parecen sentirse más intimidados ante la programación que sus compañeros del bachillerato de Ciencias. Los motivos que ya han sido discutidos en el apartado anterior pueden tratarse con la conveniencia de ofrecer materiales introductorios que refuercen aspectos como la lógica y fomenten el pensamiento estructurado, sobre todo en los primeros contactos con la materia.

También ha quedado reflejado que las mujeres tienden a percibir mayor dificultad, a pesar de que su rendimiento no es inferior al de sus compañeros. Esta diferencia entre capacidad y percepción puede estar relacionado con factores como la autoexigencia o la falta de confianza Por ello, es fundamental reforzar la confianza del alumnado y, sobre todo, evitar la estigmatización de la programación y de las asignaturas científicas como contenidos reservados a personas “muy inteligentes”. La programación, aunque muchas veces se percibe como una materia compleja (en parte por la imagen social que la envuelve), no tiene por qué ser inaccesible. Con el enfoque adecuado y los recursos apropiados cualquier estudiante puede alcanzar un rendimiento muy alto y ser exitoso en el proceso.

En resumen, los resultados del estudio apuntan a la necesidad de abordar el trabajo de esta asignatura desde una perspectiva práctica, alejándose de metodologías centradas en la memorización mecánica y promoviendo enfoques basados en la experimentación y la resolución de problemas. Por último, es importante que los propios alumnos entiendan que no existe un perfil único ideal para ser exitoso en la asignatura. Lo fundamental es que el enfoque de estudio sea activo, progresivo y orientado a la comprensión, no a la repetición o a la memorización sin sentido. Reforzar esta idea desde el aula y desde la orientación académica puede marcar una diferencia clave en la experiencia de aprendizaje y en los resultados obtenidos.

5.4 Limitaciones del estudio y futuras líneas de investigación

Antes de terminar, es importante señalar algunas limitaciones que condicionan el alcance del estudio. En primer lugar, la muestra utilizada es relativamente reducida (139 observaciones tras la depuración de los datos) y, aunque se ha trabajado con una muestra heterogénea en términos de grados, especialidades y género, el estudio cubre únicamente estudiantes de la Universidad Pontificia de Comillas. Esto puede limitar la extrapolación de los resultados a otros contextos académicos o grupos estudiantiles.

Por otro lado, la variable que se ha empleado para medir el rendimiento académico en programación se basa en una autoevaluación del propio estudiante, y no en calificaciones concretas, lo que introduce un gran margen de subjetividad y sesgo en probablemente una gran parte de las respuestas.

Finalmente, este trabajo abre la puerta a posibles líneas de investigación que podrían ayudar a completar el entendimiento de como los estudiantes de grados no técnicos aprenden programación. Por un lado, sería muy útil ampliar la muestra y repetir el estudio en otras universidades, ya que permitiría comprobar si los patrones observados se mantienen en distintos contextos o si cambian en función del entorno académico. También sería interesante trabajar con datos de rendimiento reales, como las calificaciones oficiales, en lugar de usar una autoevaluación del propio estudiante. Esta variable aporta mucha información relevante y, probablemente ayude a refinar las conclusiones. Por último, también sería interesante plantear un seguimiento que permita observar cómo evolucionan las percepciones de los alumnos sobre la programación, a medida que avanzan de curso. Este análisis permitiría ver si, con el paso del tiempo y la experiencia, los estudiantes se sienten más cómodos con la asignatura o si, por el contrario, la dificultad percibida aumenta a medida que el contenido se vuelve más complejo.

Aunque todavía queda mucho por investigar, este trabajo contribuye modestamente a mejorar la forma en que se enseña y se entiende la programación en contextos diversos, más allá de los perfiles puramente STEM.

6. Declaración respecto al uso de Chat GPT u otras herramientas de IAG

Por la presente, yo, Laura García Arnau, estudiante de doble grado de Derecho y Business Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado " Estrategias de Estudio en programación de grados no STEM" declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. Brainstorming de ideas de investigación: Utilizado para idear y esbozar posibles áreas de investigación.
2. Referencias: Usado conjuntamente con otras herramientas para identificar referencias preliminares que luego he contrastado y validado.
3. Interpretador de código: Para realizar análisis de datos preliminares.
4. Corrector de estilo literario y de lenguaje: Para mejorar la calidad lingüística y estilística del texto.
5. Sintetizador y divulgador de libros complicados: Para resumir y comprender literatura compleja.
6. Revisor: Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
7. Traductor: Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

7. Referencias

Amores, L. (2020). *La brecha de género: Análisis de perfiles de estudiantes y profesionales en las áreas STEM*. Universidad de Salamanca. https://gredos.usal.es/bitstream/handle/10366/143607/TFM_AmoresGarc%C3%ADaLBrechag%C3%A9nero.pdf?sequence=1

Arroyo-Barrigüete, J. L., Carabias López, S., Hernández, A., & Segura, M. (2023). Efecto de la especialidad en bachillerato en el rendimiento matemático en la universidad: Un estudio comparativo en grados de Administración de Empresas. *Revista de Educación*, *1*(402), 115–140. https://doi.org/10.4438/1988-592X-RE-2023-402-597

Basak, S. K., & Govender, D. (2015). Theoretical framework of the factors affecting university academics' job satisfaction. *International Business & Economics Research Journal (IBER)*, *14*(2), 317–326. https://doi.org/10.19030/iber.v14i2.9167

Baser, M. (2013). Attitude, gender and achievement in computer programming. *Middle-East Journal of Scientific Research*, *14*(2), 248–255. https://doi.org/10.5829/idosi.mejsr.2013.14.2.2007

Dodobara, F. R. (2005). Influencia de la autoeficacia en el ámbito académico. *Revista Digital de Investigación en Docencia Universitaria*, *1*(1), 1–16. https://www.redalyc.org/articulo.oa?id=498573055003

Domènech-Casal, J., Lope, S., & Mora, L. (2019). Qué proyectos STEM diseña y qué dificultades expresa el profesorado de secundaria sobre aprendizaje basado en proyectos. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, *16*(2), 2203. https://doi.org/10.25267/Rev_Eureka_ensen_divulg_cienc.2019.v16.i2.2203

European Commission. (2013). *Horizon 2020 Work Programme 2014–2015: General introduction* (C(2013)8631). https://ec.europa.eu/programmes/horizon2020/en/h2020-section/general-introduction

Figueiredo, F., & García-Peñalvo, F. J. (2018). Estrategias de enseñanza y aprendizaje de la programación en cursos universitarios. *Repositorio GRIAL*. https://repositorio.grial.eu/server/api/core/bitstreams/fd3fc97c-5dda-434a-9a6f-d44237f4cfc4/content

Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., ... & Tutty, J. (2006). Predictors of success in a first programming course. In *Proceedings of the 8th Australasian Computing Education Conference (ACE 2006)* (Vol. 52, pp. 189–196). https://dl.acm.org/doi/10.5555/1151869.1151894

García, A., Fernández, L., Martínez, P., & Torres, C. (2020). Perspectiva de género y fomento de la diversidad en la docencia de Ingeniería del Software. Conferencia, Universidad de Salamanca, Salamanca. https://www.researchgate.net/publication/342097906

Govender, I., Govender, D. W., Havenga, M., Mentz, E., Breed, B., Dignum, F., & Dignum, V. (2014). Increasing self-efficacy in learning to program: Exploring the benefits of explicit instruction for problem solving. *The Journal for Transdisciplinary Research in Southern Africa*, *10*(1). https://doi.org/10.4102/td.v10i1.19

Gunasekara, N. (2021). Gender differences in learning programming: An analysis by students' perception. *International Journal of Research and Innovation in Social Science (IJRISS)*, *5*(5), 313–319. https://www.researchgate.net/publication/351956307

Jiménez, R. G., & Fernández, C. J. (2016). La brecha de género en la educación tecnológica. *Ensaio: Avaliação e Políticas Públicas em Educação*, *24*(92), 743–771. https://doi.org/10.1590/S0104-40362016000300010

Kanaparan, G., Cullen, R., & Mason, D. (2019). Effect of self-efficacy and emotional engagement on introductory programming students. *Australasian Journal of Information Systems*, *23*. https://doi.org/10.3127/ajis.v23i0.1825

Lindblom-Ylänne, S., Parpala, A., & Postareff, L. (2019). What constitutes the surface approach to learning in the light of new empirical evidence? *Studies in Higher Education*, *44*(12), 2183–2195. https://doi.org/10.1080/03075079.2018.1482267

Martínez-Mejía, R. D. (2021). *Factores determinantes para el aprendizaje de programación de computadoras* (Tesis doctoral, University of Puerto Rico, Río Piedras). https://hdl.handle.net/11721/2617

Marton, F., & Säljö, R. (1976). On qualitative differences in learning: 1–Outcome and process. *British Journal of Educational Psychology*, *46*(1), 4–11. https://doi.org/10.1111/j.2044-8279.1976.tb02980.x

Palomares-Ruiz, A., Cebrián, A., & López-Pina, J. (2020). E-igualdad de género y rendimiento académico en entornos virtuales de aprendizaje: Un estudio inter-sujetos. *Formación Universitaria*, *13*(5), 137–146. https://doi.org/10.4067/S0718-50062020000500137

Perrupato, S. D. (2020). La relación entre teoría y práctica en el campo de la didáctica: Reflexiones pretéritas y proyecciones actuales. https://doi.org/10.29393/Pa67-4RTPCX0004

Ranjeeth, L., & Padayachee, I. (2024). Factors that influence computer programming proficiency in higher education. *South African Computer Journal*, *36*(1). https://doi.org/10.18489/sacj.v36i1.18819

Ranjeeth, S. (2011). The impact of learning styles on the acquisition of computer programming proficiency. *Alternation*, *18*(1), 336–353. https://typeset.io/pdf/the-impact-of-learning-styles-on-the-acquisition-of-computer-2xrmdwf8qq.pdf

Razo Godínez, M. L. (2008). La inserción de las mujeres en las carreras de ingeniería y tecnología. *Perfiles Educativos*, *30*(121), 63–96. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-26982008000300004

Rivera, S. M., Chotto, M. C., & Villalobos-Murillo, J. (2017). Participación de las mujeres en la carrera de Ingeniería Informática de la Universidad Nacional y su desempeño en los cursos de programación. *Revista Electrónica Educare*, *21*(1), 17. https://doi.org/10.15359/ree.21-1.2

Sarría-Stuart, Á., Gómez-Sarría, A., Martínez Casanova, L. M., & Suárez Vélez, I. (2023). Desempeño académico en la enseñanza de la programación: Consideraciones desde la perspectiva de género. *Mendive. Revista de Educación*, *21*(2). http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-76962023000200015

Spada, M. M., & Moneta, G. B. (2011). A metacognitive-motivational model of surface approach to studying. *Educational Psychology*, *32*(1), 45–62. https://doi.org/10.1080/01443410.2011.625610

Winslow, L. E. (1996). Programming pedagogy: A psychological overview. *ACM SIGCSE Bulletin*, *28*(3), 17–22. https://doi.org/10.1145/234867.234872

8. Anexo

8.1 Anexo 1: Encuesta

***Fecha**:_25/02/2024__*

***Grado (señala la opción que corresponda)**: E2 + Analytics / E3 + Analytics / E6 + Analytics*

***Curso (señala la opción que corresponda)**: 3º / 4º / 5º*

Soy mayor de edad y acepto que mis respuestas, tratadas de forma absolutamente anónima, sean empleadas con la finalidad de investigación académica

- ☐ *Sí*
- ☐ *No*

En asignaturas de programación del doble grado (Programación) consideras que no tuviste de aprender nada de memoria (marca el 0) lo aprendiste todo de memoria (marca el 10). Selecciona lo que corresponda.

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

En asignaturas de programación del doble grado (Programación) ¿aproximadamente qué porcentaje del tiempo de estudio dedicaste a estudiar teoría y que porcentaje a practicar problemas? Selecciona lo que corresponda.

☐0% teoría – 100% problemas
☐20% teoría – 80% problemas
☐40% teoría – 60% problemas
☐60% teoría – 40% problemas
☐80% teoría – 20% problemas
☐100% teoría – 0% problemas

Comparado con otras asignaturas de tu doble grado ¿Cuánto tiempo dedicaste a estudiar las asignaturas de programación (Programación)? Valóralo en una escala del 0 (mucho menos) al 10 (mucho más).

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

En asignaturas de programación del doble grado (Programación), cuando estudias/estudiabas la asignatura ¿lo hacías usando dispositivos electrónicos o con papel y bolígrafo? Selecciona lo que corresponda.

☐Solo uso dispositivos electrónicos (ordenador, tablet, etc.)
☐La mayoría del estudio lo hago con dispositivos electrónicos (ordenador, tablet, etc.)
☐La mitad en papel y la mitad con dispositivos electrónicos
☐La mayoría del estudio lo hago en papel
☐Todo el estudio lo hago en papel

Centrándonos solo en la parte de programación en Matlab, ¿Cuál consideras que es el grado de dificultad de LA PARTE DE PROGRAMACIÓN de las asignaturas de matemáticas (Álgebra y Cálculo)? Valóralo en una escala del 0 (extremadamente fácil) al 10 (extremadamente difícil).

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

Centrándonos solo en la parte de programación en R, ¿Cuál consideras que es el grado de dificultad de LA PARTE DE PROGRAMACIÓN de las asignaturas de estadística

(Estadística y Probabilidad, y Métodos Estadísticos)? Valóralo en una escala del 0 (extremadamente fácil) al 10 (extremadamente difícil).

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

Centrándonos solo en la parte de programación en R y/o Python, ¿Cuál consideras que es el grado de dificultad de LA PARTE DE PROGRAMACIÓN de las asignaturas de Machine Learning (Machine Learning I, II y III, o las que hayas cursado de estas hasta el momento)? Valóralo en una escala del 0 (extremadamente fácil) al 10 (extremadamente difícil).

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

¿Cuál consideras que es el grado de dificultad de las asignaturas de programación (Programación)? Valóralo en una escala del 0 (extremadamente fácil) al 10 (extremadamente difícil).
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

¿Cuál consideras que ha sido tu rendimiento (nota obtenida) en las asignaturas de programación (Programación)? Valóralo en una escala del 0 (muy por debajo de la media de mi clase) al 10 (muy por encima de la media de mi clase).
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9 ☐10

¿Qué especialidad cursaste en bachillerato?
☐Ciencias ☐Ciencias Sociales ☐Otras

¿Cuál es tu género?
☐Hombre ☐Mujer ☐Otros

¿Qué nota obtuviste en la EvAU (fase general, escala del 0 al 10)? _____