

UNIVERSIDAD DE COSTA RICA  
SISTEMA DE ESTUDIOS DE POSGRADO

IMPLEMENTACIÓN DEL MODELO DE RASCH PARA LA GENERACIÓN  
AUTOMÁTICA DE PRUEBAS DE AULA

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del  
Programa de Estudios de Posgrado en Estadística para optar al grado y título de  
Maestría Profesional en Estadística

MANUEL FELIPE MASÍS SOTO

Ciudad Universitaria Rodrigo Facio, Costa Rica

2023

## **Dedicatoria**

Por supuesto primero las gracias a Dios por permitirme llegar a culminar esta etapa y a la Santísima Virgen María por su intercesión.

Dedico este trabajo a mi esposa Ericka y a mis hijos Juliana, Daniela y Sebastián, quienes siempre estuvieron y están apoyándome en mi caminar.

“Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Estadística de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Profesional en Estadística.”

---

Ph.D. Guaner Rojas Rojas

**Profesor Guía**

---

Ph.D. Eiliana Montero Rojas

**Lectora**

---

Ph.D. Francisco Torres Rojas

**Lector**

---

Manuel Felipe Masís Soto

**Sustentante**

## Tabla de contenido

<b>RESUMEN</b> .....	vii
<b>ABSTRACT</b> .....	viii
<b>Lista de gráficos</b> .....	ix
<b>Lista de cuadros</b> .....	ix
<b>Lista de figuras</b> .....	x
<b>CAPÍTULO I: INTRODUCCIÓN</b> .....	1
<b>1.1 Contexto</b> .....	1
<b>1.2 Objetivos de la investigación</b> .....	2
<b>1.2.1 Objetivo general</b> .....	2
<b>1.2.2 Objetivos específicos</b> .....	2
<b>1.3 Justificación</b> .....	3
<b>CAPÍTULO II: ESTADO DE LA CUESTION</b> .....	5
<b>2.1 La TRI y sus ventajas sobre la TCT</b> .....	5
<b>2.2 Fundamentos del modelo de Rasch</b> .....	7
<b>2.3 “The Examiner”: un programa informático para la construcción automatizada de pruebas de aula</b> .....	10
<b>CAPÍTULO III: METODOLOGÍA</b> .....	12
<b>3.1 Materiales</b> .....	12
<b>3.1.1 Elementos requeridos para el sistema computacional “The Examiner”</b> .....	12
<b>3.1.2 Población</b> .....	13
<b>3.2 Métodos</b> .....	13
<b>3.2.1 Funcionamiento general del software</b> .....	13
<b>3.2.2 Planteamiento del flujo de trabajo</b> .....	16
<b>3.2.3 Métodos relacionados con la TCT</b> .....	18
<b>3.2.3.1 Índice de dificultad del ítem</b> .....	18
<b>3.2.3.2 El coeficiente alfa</b> .....	18
<b>3.2.3.3 Índice de correlación biserial puntual</b> .....	19
<b>3.2.4 Métodos relacionados con la TRI: modelo de Rasch</b> .....	19
<b>CAPÍTULO IV: RESULTADOS</b> .....	23
<b>4.1 Instalación del software</b> .....	23

<b>4.2</b>	<b>Comprensión de la base de datos y del código fuente .....</b>	<b>24</b>
<b>4.2.1</b>	<b>Resumen del modelo de la base de datos.....</b>	<b>24</b>
<b>4.2.2</b>	<b>Comprensión de la organización y estructura del código fuente:.....</b>	<b>26</b>
<b>4.3</b>	<b>Análisis descriptivos, recreación de notas e indicadores de exámenes reales.</b>	<b>27</b>
<b>4.3.1</b>	<b>Análisis descriptivos.....</b>	<b>27</b>
<b>4.3.2</b>	<b>Recreación de notas del primer examen consignado en la base de datos e indicadores de referencia generados por el software para un examen de prueba.....</b>	<b>29</b>
<b>4.3.2.1</b>	<b>Recreación de notas del primer examen consignado en la base de datos. ....</b>	<b>29</b>
<b>4.3.2.2</b>	<b>Indicadores de referencia generados por el software para un examen de prueba. 30</b>	
<b>4.3.3</b>	<b>Análisis de exámenes reales consignados en la base de datos con un enfoque de TCT.</b>	<b>34</b>
<b>4.3.3.1</b>	<b>Análisis de la media real vs. media esperada para los 50 exámenes con mayor participación de estudiantes.....</b>	<b>35</b>
<b>4.3.3.2</b>	<b>Análisis del coeficiente alfa real vs. coeficiente alfa de referencia para los 50 exámenes con mayor participación de estudiantes. ....</b>	<b>37</b>
<b>4.3.3.3</b>	<b>Análisis de la desviación estándar como valor de referencia para la generación de exámenes: .....</b>	<b>40</b>
<b>4.3.3.4</b>	<b>Análisis del coeficiente de correlación biserial puntual.....</b>	<b>40</b>
<b>4.4</b>	<b>Generación y análisis de exámenes simulados con TCT.....</b>	<b>42</b>
<b>4.5</b>	<b>Análisis de exámenes reales y simulados a partir de un enfoque de la TRI ...</b>	<b>46</b>
<b>4.5.1</b>	<b>Bondad de ajuste del modelo de Rasch en exámenes reales y simulados.....</b>	<b>46</b>
<b>4.5.2</b>	<b>Ajuste de ítems y personas .....</b>	<b>46</b>
<b>4.5.2.1</b>	<b>Ajuste de ítems y personas en exámenes reales .....</b>	<b>46</b>
<b>4.5.2.2</b>	<b>Ajuste de ítems y personas en exámenes simulados .....</b>	<b>48</b>
<b>4.5.3</b>	<b>Construcción de una base de datos de ítems con índices de dificultad .....</b>	<b>50</b>
<b>4.5.4</b>	<b>Análisis de correlación entre índices de dificultad de la base de datos creada y los índices de dificultad de los exámenes simulados .....</b>	<b>52</b>
	<b>CAPÍTULO V: CONCLUSIONES Y DISCUSION.....</b>	<b>55</b>
	<b>REFERENCIAS.....</b>	<b>58</b>
	<b>ANEXOS.....</b>	<b>60</b>
	<b>ANEXO No.1: GUÍA ILUSTRADA DEL SOFTWARE “THE EXAMINER” .....</b>	<b>61</b>

<b>ANEXO NO.2: PROCEDIMIENTO PARA RECREACIÓN DE LAS NOTAS DE UN ESTUDIANTE (CON R Y CONECTÁNDOSE A LA BASE DE DATOS):.....</b>	<b>72</b>
<b>ANEXO NO.3: EXAMEN DE PRUEBA PARA CÁLCULO DE INDICADORES....</b>	<b>76</b>
<b>ANEXO NO.4: PROCEDIMIENTO PARA ANÁLISIS DE 50 EXÁMENES REALES CON MAYOR PARTICIPACIÓN DE ESTUDIANTES.....</b>	<b>78</b>
<b>ANEXO NO.5: PROCEDIMIENTO PARA ANÁLISIS GENERACIÓN DE MATRICES BINARIAS A PARTIR DE PREGUNTAS, ESTUDIANTES Y RESPUESTAS.....</b>	<b>80</b>
<b>ANEXO NO.6, PROCEDIMIENTO PARA ANÁLISIS DE EXÁMENES SIMULADOS CON ENFOQUE TCT. ....</b>	<b>86</b>
<b>ANEXO NO.7, PROCEDIMIENTO PARA ANÁLISIS DE EXÁMENES SIMULADOS CON ENFOQUE TRI. ....</b>	<b>88</b>

## RESUMEN

Este trabajo se circunscribe a temas relacionadas con la automatización de exámenes en el aula, partiendo de la teoría clásica de los tests (TCT), para luego analizar el aporte de la teoría de respuesta al ítem (TRI), específicamente por medio del modelo de Rasch y utilizando como base una herramienta de software denominada “The Examiner”, que permite la creación, calificación y análisis de pruebas de aula.

Se aprovecha la información consignada en la base de datos del software para 1428 ítems utilizados en exámenes reales y aplicados en una universidad estatal, con el fin de analizar los indicadores que el software calcula como predichos. Se realiza una comprobación de los cálculos internos del software, para concluir que la información generada es útil para la confección de exámenes. En este sentido y como elementos clave, el software aporta información sobre la media esperada para los exámenes aplicados y sobre el coeficiente de confiabilidad o consistencia esperado mediante el alfa de Cronbach. Los resultados del análisis permiten concluir que los cálculos que genera el software para estos valores predichos se pueden utilizar como información de referencia útil para la confección previa de exámenes a partir de ítems consignados con información histórica.

Por otro lado, se utiliza la información de la base de datos para ajustar el modelo de Rasch a las pruebas de aula ejecutadas, así como a pruebas simuladas y se concluye que este modelo genera un buen ajuste a nivel de exámenes, ítems y personas, por lo cual se convierte en una alternativa para incorporarlo como parte de las funcionalidades del software en versiones posteriores y así enriquecer el análisis que brinda el enfoque de la TCT.

**Palabras clave:** teoría clásica de los tests, TCT, teoría de respuesta al ítem, TRI, pruebas de aula automatizadas, modelo de Rasch

## ABSTRACT

This work is related to the automation of exams in the classroom, starting from the classical test theory (CTT) and then analyzing the contribution of the item response theory (IRT), specifically through the Rasch model and using a software tool called “The Examiner”, which allows the creation, grading and analysis of classroom tests.

The information recorded in the software database for 1428 items used in real exams and applied in a state university is used, in order to analyze the indicators that the software calculates as predicted. A verification of the internal calculations of the software is carried out, to conclude that the information generated is useful for the preparation of exams. In this line and as key elements, the software provides information on the expected average for the applied tests and on the expected alpha coefficient. The results of the analysis allow to conclude that the calculations generated by the software for these predicted values can be used as useful reference information for the preliminary preparation of examinations from items recorded with historical information.

On the other hand, the information from the database is used to adjust the Rasch model to the classroom tests carried out, as well as to simulated tests, and it is concluded that this model generates good adjustments either for exams, items, and people. Then, it becomes an alternative to incorporate it as part of the software functionalities in later versions and this way enrich the analysis provided by the TCT approach.

**Key Words:** classical test theory, CTT, item response theory, IRT, classroom exams automation, Rasch model



## Lista de gráficos

Gráfico No. 1: Ejemplo de una curva característica de un ítem.....	21
Gráfico No. 2: Ejemplo de una curva característica de un ítem.....	22
Gráfico No. 3: : Ejemplo de una curva característica de un ítem.....	22
Gráfico No. 4: Histograma alumnos por examen.....	28
Gráfico No. 5: Histograma 50 Exámenes con.....	34
Gráfico No. 6: Cantidad de alumnos por pregunta para exámenes simulados .....	42
Gráfico No. 7: Gráfico de jerarquía para coeficientes alfa 18 exámenes simulados .....	45
Gráfico No. 8: Ítems utilizados en exámenes reales vs. índice de dificultad .....	50
Gráfico No. 9: Índices de dificultad "ítems históricos" vs. ....	53
Gráfico No. 10: Selección de 20 ítems por índice de dificultad .....	54

## Lista de cuadros

Cuadro No. 1: Materias, exámenes y calificaciones en la base de datos.....	27
Cuadro No. 2: Valores para cálculo de media esperada para examen de prueba.....	31
Cuadro No. 3: Resultados intervalos de confianza exámenes reales analizados.....	36
Cuadro No. 4: Varianza según cantidad de respuestas buenas por .....	37
Cuadro No. 5: Categoría coeficientes alfa corregidos.....	39
Cuadro No. 6: Distribución por materia de preguntas.....	43
Cuadro No. 7: Resultados exámenes simulados .....	44
Cuadro No. 8: Exámenes a revisar por ajuste de ítems y/o estudiantes .....	47
Cuadro No. 9: Resultado de eliminación por Infit no adecuado.....	47
Cuadro No. 10: Exámenes simulados para revisión de valores <i>Infit</i> .....	48
Cuadro No. 11: Resultado de eliminar items/sujetos con valores de estadístico .....	49
Cuadro No. 12: Índices de dificultad por rangos.....	51
Cuadro No. 13: Resultados de exámenes simulados .....	52

## **Lista de figuras**

Figura No. 1: Entrada de datos al software.....	14
Figura No. 2: Proceso de generación de exámenes .....	15
Figura No. 3: Flujo de tareas a realizar.....	16
Figura No. 4: Pantalla principal del software en funcionamiento .....	23
Figura No. 5: Modelo de la base de datos .....	24
Figura No. 6: Vista examen de prueba para cálculo de indicadores.....	30
Figura No. 7: Ejemplo de mensajes sobre índices de correlacion biserial puntual por pregunta ....	41
Figura No. 8: Pantalla de inicio.....	61
Figura No. 9: Pantalla de datos .....	63
Figura No. 10: Pantalla exámenes.....	64
Figura No. 11: Ejemplo de Esquema de examen .....	65
Figura No. 12: Pantalla de asignación de boletos por pregunta .....	66
Figura No. 13: Pantalla de Creación de Preexámenes .....	67
Figura No. 14: Pantalla de Refinamiento de Preexámenes .....	68
Figura No. 15: Pantalla para convertir preexamen en examen .....	69
Figura No. 16: Pantalla de Análisis y Ajuste de Exámenes.....	70
Figura No. 17: Ejemplo de gráfico de preguntas según discriminación y dificultad .....	71

## CAPÍTULO I: INTRODUCCIÓN

### 1.1 Contexto

En la actualidad, la construcción y aplicación de exámenes sigue siendo una de las formas importantes que los docentes utilizan para evaluar el nivel de conocimiento adquirido por los estudiantes; este tipo de pruebas consideran aspectos específicos del aprendizaje generado con un cierto proceso de mediación pedagógica y contexto particular de dinámica del aula. En el proceso de construcción de pruebas de aula, normalmente la calidad de los ítems es desconocida, pocas veces documentada y en términos de confiabilidad rara vez se calculan sus medidas empíricas. (Moreira-Mora, y otros, 2022).

En este contexto y como parte de una investigación aplicada, se desarrolló en su momento un sistema computacional llamado “The Examiner”, que facilita la preparación, evaluación y análisis de exámenes, basado en una metodología experimental de generación de pruebas de selección única, que fueron aplicadas en cursos ofrecidos en distintos semestres en la carrera de Ingeniería en Computación de la Escuela de Computación del Instituto Tecnológico de Costa Rica. Como resultado de este proceso automatizado por medio del sistema computacional, se logró construir una base de datos con información histórica del comportamiento de cada examen y de cada ítem utilizado. (Torres-Rojas, 2008)

La información consignada por medio de “The Examiner” se circunscribe al ámbito de lo que se refiere a la Teoría Clásica de los Tests (TCT), ya que los puntajes totales de las pruebas para los examinados se basan en la suma de las respuestas a cada uno de los ítems individuales (Price, 2017) y se parte de la premisa de que la prueba esté bien diseñada, el puntaje obtenido debería ser una estimación razonable del conocimiento que tienen los estudiantes sobre el tema examinado (Holmes Finch, French, & Immekus, 2014).

A pesar de que el modelo propuesto por la TCT es un modelo útil y de amplia aplicación, presenta limitaciones que han venido a ser solventadas por otros modelos matemáticos que relacionan el rasgo latente o el puntaje de habilidad de un examinado, con la probabilidad de responder en una categoría de respuesta específica de un ítem. (Paek & Cole, 2020). Esta aproximación se refleja en un conjunto de modelos etiquetados de forma general como Teoría

de la Respuesta al Ítem (TRI) (Martínez Arias, Hernández Llorede, & Hernández Llorede, 2014)

El presente trabajo se ubica en el contexto de los modelos estadísticos de la TRI, específicamente por medio de la implementación del modelo de Rasch utilizando como base el software “The Examiner”, para analizar el desempeño en la recuperación de parámetros del modelo de la TCT, comparar con la generación de parámetros del modelo de TRI y sugerir ajustes para modificar el sistema computacional mencionado, con el fin de que sea también una herramienta para operacionalizar la aplicación del modelo de Rasch en la generación automatizada de pruebas de aula. En este sentido el modelo de Rasch, se referencia además como un modelo de medición, ya que se buscan datos que se ajusten al modelo para poder obtener sus ventajas (Prieto & Delgado, 2003), en lugar de ajustar en sí mismo el modelo a los datos.

## **1.2 Objetivos de la investigación**

### **1.2.1 Objetivo general**

Evaluar el funcionamiento del modelo de Rasch en la generación automática de pruebas de aula con el algoritmo del software “The Examiner”.

### **1.2.2 Objetivos específicos**

- a) Implementar el modelo de Rasch en la generación y análisis de pruebas informatizadas de aula.
- b) Comparar el desempeño en la recuperación de parámetros y validez de las pruebas de los modelos de TRI (particularmente del modelo de Rasch), en relación con el modelo de la TCT, en las pruebas informatizadas de aula.
- c) Recomendar ajustes al algoritmo para implementar un modelo de Rasch en el software.

### 1.3 Justificación

La elaboración de pruebas o exámenes sigue siendo hoy uno de los principales mecanismos de la educación formal para validar o evidenciar que el conocimiento ha sido comprendido y apropiado por parte de los estudiantes. El reto es que esas pruebas o exámenes realmente evalúen de forma objetiva el contenido y el nivel de los temas para los cuales fueron diseñadas y que el resultado de la medición represente razonablemente el nivel de conocimiento adquirido por las personas.

Según (Muniz, 2010), el enfoque clásico es el predominante en la construcción y análisis de los tests. Sin embargo, con el aporte de la TRI, se han propuesto otros modelos matemáticos que aportan mayor profundidad de análisis, relacionando los rasgos latentes con el verdadero puntaje asociado con el conocimiento de una persona en el campo o temática que está siendo evaluada con instrumentos o pruebas específicas.

Por otro lado, la tecnología y su beneficio en el aumento en la capacidad computacional, ha venido a mejorar la administración y procesamiento de grandes cantidades de información, que permiten una mayor profundidad de análisis e inclusive poder automatizar la creación de instrumentos en los que se podría inferir sobre los resultados que van a arrojar en el momento de su aplicación.

A propósito de ese desarrollo a nivel de tecnología, existe una alternativa de software que utiliza elementos de la TCT y que fue desarrollada como parte de una investigación aplicada en el Instituto Tecnológico de Costa Rica, con el nombre de “The Examiner”. Este software facilita la gestión de ítems de respuesta única para la confección de pruebas en el aula y realiza algunas predicciones sobre los resultados de la aplicación de las mismas, en indicadores como la media esperada para la prueba, partiendo del comportamiento individual de los ítems que han ido acumulando información histórica, es decir, que a partir de sus respuestas reales y conforme se consignan en la base de datos, permiten generar indicadores importantes como: el coeficiente de correlación biserial puntual como medida de discriminación del ítem y el Alfa de Cronbach para cuantificar el nivel de fiabilidad o consistencia interna de las diferentes pruebas, así como el efecto individual de cada ítem. (Torres-Rojas, 2008)

Con el desarrollo de este trabajo se desea contribuir con la valoración para la ampliación del alcance que el software ofrece a nivel de los aspectos que actualmente contempla relacionados con la TCT, por medio del estudio de los resultados de la implementación de modelos de la TRI, específicamente el modelo de Rasch, con el fin de operacionalizar el uso de este modelo como parte de las funcionalidades del software puede incorporar en versiones posteriores.

## **CAPÍTULO II: ESTADO DE LA CUESTION**

En este capítulo se desarrolla el estado de la cuestión sobre el tema de la TRI y sus ventajas sobre la TCT, el modelo de Rasch y el sistema computacional “The Examiner” como alternativa para la construcción de pruebas automatizadas de aula.

### **2.1 La TRI y sus ventajas sobre la TCT**

“La denominación TRI agrupa líneas de investigación psicométricas independientes iniciadas por Rasch (1960) y Birnbaum (1968). El factor común de estos desarrollos es que establecen una relación entre el comportamiento de un sujeto frente a un ítem y el rasgo responsable de esta conducta (rasgo latente). Para ello, recurren a funciones matemáticas que describen la probabilidad de dar una determinada respuesta al ítem para cada nivel del rasgo medido por este. El objetivo sustancial de la TRI es la construcción de instrumentos de medición con propiedades invariantes entre poblaciones. Si dos individuos presentan idéntico nivel de rasgo medido, ambos tendrán igual probabilidad de dar la misma respuesta, independientemente de la población de pertenencia.” (Attorresi, Lozzia, Abal, Galibert, & Aguerri, 2009)

Según Martínez Arias y otros (Martínez Arias, Hernández Lloreda, & Hernández Lloreda, 2014), la TRI parte de la premisa de que el comportamiento de un sujeto ante un ítem puede explicarse en términos de una o varias características del sujeto denominadas rasgos o aptitudes latentes y que lo que se puede estimar son las puntuaciones de los sujetos en esos rasgos y utilizar esos valores para explicar la puntuación que tendrá un sujeto en una prueba.

Las mismas investigadoras (Martínez Arias, Hernández Lloreda, & Hernández Lloreda, 2014) indican que, al realizar una revisión rápida de la TCT, se evidencia una serie de desventajas que intenta superar la TRI; a continuación, las desventajas más relevantes citadas por las autoras:

- a) Dependencia de los estadísticos del ítem de la población y/o muestra en la que se calculan: los estadísticos del ítem de la TCT dependen de la tendencia central y de la variabilidad del rasgo en la muestra y en la población; los índices de dificultad serán más altos (representando ítems más fáciles) cuando los participantes están por encima del promedio; los índices de discriminación tienden a ser más elevados en grupos de mayor variabilidad como efecto de la heterogeneidad del grupo sobre el coeficiente de correlación en el que se basa.
- b) Dependencia de los estadísticos de la prueba, de la población y/o muestra: los diferentes estadísticos como el coeficiente de fiabilidad, dependen de la variabilidad de los grupos en los que se calculan.
- c) Dependencia de las puntuaciones del conjunto particular de ítems utilizado en la prueba: la puntuación empírica total depende de los ítems específicos que componen la prueba y como resultado, no se puede hablar en términos generales de la aptitud del sujeto, sino que se debe especificar siempre la prueba con la que fue determinada esa puntuación determinada.
- d) La TCT no proporciona un modelo teórico para las respuestas a los ítems: no permite determinar el comportamiento del participante en función de la cantidad del rasgo latente que posee cuando responde a un ítem.

En contraposición con estas desventajas, las investigadoras Rojas y Montero (Rojas & Montero, 2021) indican una serie de ventajas de la TRI (refiriéndose específicamente al modelo de Rasch que es el utilizado en este trabajo), de las cuales las más relevantes se mencionan a continuación:

- A partir de la puntuación obtenida por el modelo de Rasch se puede establecer a nivel sustantivo los logros del estudiante en cuanto a procesos y contenidos presentes la prueba que se esté evaluando, así como las áreas de oportunidad en términos del constructo objeto de medición.
- Se puede generar una estimación del error para cada puntuación estimada en el constructo de interés, que se traduce la posibilidad de generación de pruebas de mayor calidad al contar con mejores herramientas de análisis del rasgo latente del participante así como la dificultad del ítem.



- El modelo de Rasch permite también permite establecer a nivel de los examinados si las respuestas brindadas en el examen sugieren posibilidad de respuestas al azar o con evidente descuido, así como la posibilidad de copia, información que a nivel investigativo y operativo puede ser de gran interés en torno a las posibles causas de estas situaciones y las formas de evitarlas.
- En el modelo de Rasch se cumple que diferencias iguales en la puntuación representan diferencias iguales en el constructo. Es decir, diferencias del mismo tamaño en las puntuaciones, representan diferencias del mismo tamaño en el constructo.
- El modelo de Rasch siempre es estimable, a diferencia de otros modelos TRI, lo cual es una ventaja para trabajos de toma de decisiones y no investigativos (como es el caso de las pruebas de aplicación en el aula), lo cual asegura que siempre se puede estimar el modelo y obtener resultados.

## 2.2 Fundamentos del modelo de Rasch

El modelo de Rasch es un caso particular de los modelos de TRI. Con base en los trabajos de Wind y Hua (2022), se indica que el modelo dicotómico de Rasch es el más simple de los modelos de la familia de Rasch y fue diseñado para utilizarse con situaciones en las que los ítems tienen como respuesta dos categorías, usualmente 0 y 1, las cuales indican el hecho de que una persona responda a un ítem particular, de forma incorrecta o de forma correcta respectivamente. Este modelo utiliza la suma de los puntajes de las respuestas para calcular estimaciones de la posición de las personas (en cuanto a su habilidad o nivel de competencia) y posición de los ítems (en cuanto al nivel de dificultad para contestarlo de forma correcta), por medio de una escala que representa la variable latente (escala de “log-odds” o escala “logit”). La diferencia entre la localización de una persona y un ítem se utiliza para calcular la probabilidad de obtener una respuesta correcta ( $x = 1$ ), en lugar de una respuesta incorrecta ( $x = 0$ ).

La ecuación del modelo dicotómico de Rasch se expresa por medio de la siguiente fórmula:

$$P(x_{ni} = 1) = \frac{e^{(\theta_n - \delta_i)}}{1 + e^{(\theta_n - \delta_i)}}$$

Donde:  $P(x_{ni} = 1)$ , se refiere a la probabilidad de contestar de forma correcta el ítem  $i$  por el sujeto  $n$ ,

$\theta_n$  se refiere al nivel de habilidad del sujeto  $n$ ,

$\delta_i$  se refiere al nivel de dificultad del ítem  $i$

El modelo de Rasch, también se puede expresar como una función logística, como se muestra a continuación:

$$\ln \left[ \frac{\phi_{ni1}}{\phi_{ni0}} \right] = \theta_n - \delta_i$$

Donde:

$\phi_{ni1}$ : es la probabilidad que la persona  $n$  conteste de forma correcta ( $x=1$ ), al ítem  $i$

$\phi_{ni0}$ : es la probabilidad que la persona  $n$  conteste de forma incorrecta ( $x=0$ ), al ítem  $i$  y que equivale a  $1 - \phi_{ni1}$

$\theta_n$ : se refiere a la habilidad o nivel de competencia de la persona  $n$

$\delta_i$ : se refiere al nivel de dificultad del ítem  $i$

Siguiendo a Prieto y Delgado (Prieto & Delgado, 2003), se pueden mencionar muchas ventajas que presenta el modelo de Rasch y entre las más relevantes que mencionan estos autores están:

- **Medición conjunta:** esto significa que los parámetros de las personas y los parámetros de los ítems se pueden expresar en las mismas unidades y se localizan en un mismo continuo. Esto hace que el modelo de Rasch sea más realista que el modelo propuesto por la TCT, ya que no es razonable pensar que todos los ítems miden la misma cantidad del constructo en evaluación. Además, esta condición de medición conjunta permite analizar la interacción entre individuos e ítems, lo que permite un análisis de

mayor posibilidad diagnóstica en cuanto a ítems en los que un individuo particular puede presentar mayor o menor habilidad y así no limitar el análisis al grupo como tal.

- **Objetividad específica:** esta condición establece que la diferencia entre dos personas en un atributo no debe depender de los ítems específicos con los que se estima esa diferencia y de igual manera, la diferencia entre dos ítems no debería depender de las personas específicas que se toman en cuenta para medirla. Con el modelo de Rasch se puede mostrar partiendo de dos personas con diferente nivel que contestan el mismo ítem, de la siguiente forma:

$$\ln \left[ \frac{\phi_{1i1}}{\phi_{1i0}} \right] = \theta_1 - \delta_i \text{ y } \ln \left[ \frac{\phi_{2i1}}{\phi_{2i0}} \right] = \theta_2 - \delta_i$$

Entonces, la diferencia entre las dos personas se estima con base habilidad de cada una, independientemente del ítem que contestaron:

$$\ln \left[ \frac{\phi_{1i1}}{\phi_{1i0}} \right] - \ln \left[ \frac{\phi_{2i1}}{\phi_{2i0}} \right] = (\theta_1 - \delta_i) - (\theta_2 - \delta_i) = (\theta_1 - \theta_2)$$

Por otro lado, si la misma persona contestara dos ítems de diferente dificultad, la diferencia entre los ítems no toma en cuenta a la persona que los contesta:

$$\ln \left[ \frac{\phi_{n11}}{\phi_{n10}} \right] - \ln \left[ \frac{\phi_{n21}}{\phi_{n20}} \right] = (\theta_n - \delta_1) - (\theta_n - \delta_2) = (\delta_2 - \delta_1)$$

Con lo expuesto anteriormente se puede decir que, si los datos se ajustan al modelo, la habilidad de las personas se puede estimar y comparar independientemente de la dificultad del ítem; por otro lado, la estimación del parámetro de dificultad de un ítem es independiente del nivel de habilidad de las personas a las cuales se les aplica. (Prieto & Delgado, 2003)

- **Propiedades de intervalo:** la escala *logit* que presenta el modelo de Rasch tiene

propiedad de intervalo, debido a que la interpretación de las diferencias en la escala es la misma a lo largo del atributo que se está midiendo. Esto es, que a diferencias iguales entre un sujeto y un ítem le corresponden probabilidades idénticas de una respuesta correcta.

- Especificidad del error típico de medida: una ventaja del modelo de Rasch sobre el modelo propuesto por la TCT, radica en el hecho que este último supone que los tests miden con la misma fiabilidad en todas las regiones de la variable en cuestión; en cambio el modelo de Rasch no asume esta situación sino que permite cuantificar la cantidad de información con la que se mide en cada punto de la dimensión analizada y permite seleccionar los ítems que incrementan la información en regiones del atributo específicas.

### **2.3 “The Examiner”: un programa informático para la construcción automatizada de pruebas de aula**

Como indica Torres-Rojas (2008), en las labores de docencia existe un alto consumo de horas tanto en la preparación como en la revisión de exámenes. Esta situación puede resultar de alguna forma tediosa para las personas que los preparan y podría derivar en el reciclaje de exámenes anteriores, construcción de ítems demasiado difíciles o fáciles y es hasta un momento posterior a la aplicación de las pruebas, cuando la persona que lo diseñó se percata que su examen estuvo bien o mal planteado.

En el caso de aplicación de pruebas de aula, una vez realizada la revisión de los exámenes, el docente consignará probablemente las notas en algún medio como una hoja de cálculo y la información que obtendrá se limitará al promedio general de calificaciones del grupo, la desviación estándar de esas calificaciones, tal vez la moda, podrá graficar un histograma para visualizar el comportamiento del grupo y en cuanto a los ítems utilizados, típicamente no se registrará un récord del comportamiento de los mismos. Entonces, según explica Torres-Rojas, no se aprovecha la oportunidad de realizar análisis detallados de la información subyacente en cada examen, no se realiza un análisis a nivel de cada uno de los ítems utilizados en las pruebas, no se establecen relaciones con datos históricos previos, no existen mecanismos para advertir sobre la presencia de preguntas mal diseñadas o mal

planteadas, no se pueden identificar preguntas que discriminen de forma efectiva y no se asimilan lecciones que permitan diseñar mejores exámenes para el futuro.

Circunscritos al ámbito de herramientas para la creación de exámenes de selección única, se puede comprobar que sí existen opciones a nivel comercial o libre que incluyen funcionalidades como la administración de bancos de preguntas para la generación aleatoria de pruebas y su calificación automática, hasta programas informáticos especializados para el análisis del resultado de las pruebas con un enfoque de TRI.

Ahora bien, siendo que la tarea docente incluye en la actualidad la construcción de exámenes, es deseable contar con herramientas y soluciones integrales que apoyen para: (1) gestionar bancos de preguntas segmentados por materia, temas y subtemas, (2) confeccionar pruebas automatizadas y aleatorizadas, visibilizando algunos indicadores de desempeño previo, (3) consignar la calificación de las pruebas de cada estudiante y mantener estadísticas de los exámenes aplicados, (4) mantener un histórico de cada uno de los ítems utilizados, (5) calcular indicadores de consistencia de la prueba y de los ítems, (6) calcular la discriminación individual de cada ítem, (7) presentar el comportamiento general del banco de preguntas que se han ido alimentado en una base de datos.

Siguiendo el trabajo de investigación de Torres-Rojas, sí se encuentran disponibles herramientas que automatizan partes del proceso de exámenes de aula<sup>1</sup>, pero no una herramienta que integre la generación del examen, el registro de datos históricos, la construcción de indicadores predictivos del comportamiento de los exámenes, análisis estadístico asociado y es en este contexto en el que se desarrolló “The Examiner” como una alternativa.

---

<sup>1</sup> Algunas opciones interesantes: (1) <https://assess.com/xcalibre/>

(2) [https://autopsych.shinyapps.io/version\\_1\\_0\\_0/](https://autopsych.shinyapps.io/version_1_0_0/)

(3) <https://itemanalysis.com/jmetrik-download/>

## CAPÍTULO III: METODOLOGÍA

### 3.1 Materiales

En esta sección se describen los materiales que se requirieron para la instalación y puesta en marcha del sistema computacional. Además, se declara la población utilizada en términos de la base de datos disponible al momento de la realización de este trabajo.

#### 3.1.1 Elementos requeridos para el sistema computacional “The Examiner”

En el momento de realización de este documento, la versión actualizada del sistema computacional correspondió a la 0.6 del año 2019.

Para lograr su funcionamiento, se requirió de la instalación de los siguientes elementos:

- Computadora funcionando con sistema operativo Linux: se eligió la distribución Linux Mint 21, Cinnamon 64-bit.
- Instalación de la base de datos Postgres en su versión 14+258.
- Instalación completa de Latex para la creación de los exámenes admitiendo figuras y fórmulas matemáticas.
- Instalación de la biblioteca GTK+2 y GTK+3 para creación de interfaces gráficas de usuario.
- Instalación de Gnuplot como programa para la generación de gráficas en 2 y 3 dimensiones.
- Instalación del compilador GCC.
- Instalación de Evince como elemento para despliegue de archivos en formato PDF.
- Compilación del archivo Examiner.tgz, con los paquetes de “The Examiner”.
- Recuperación del respaldo de la base de datos ex.data.2021.tgz en Postgres.
- Además, se utilizaron todos los archivos de código fuente desarrollados, para poder dar trazabilidad a las funcionalidades y cálculos internos del software.

### 3.1.2 Población

Para los fines de este trabajo, se utilizó un respaldo de la base de datos con la siguiente información:

- Cantidad de materias incluidas: 13 materias.
- Cantidad de ítems incluidos: 3500 ítems.
- Cantidad de exámenes aplicados: 158 exámenes.
- Cantidad de alumnos evaluados: 1833 alumnos con evaluaciones procesadas.

### 3.2 Métodos

En esta sección se explica de forma general el proceso necesario que se debe seguir para poder generar exámenes automáticos con el software, el flujo de tareas que se planteó para la realización de este trabajo, así como los métodos que se contemplaron para el análisis desde la perspectiva de la TCT y la TRI.

#### 3.2.1 Funcionamiento general del software

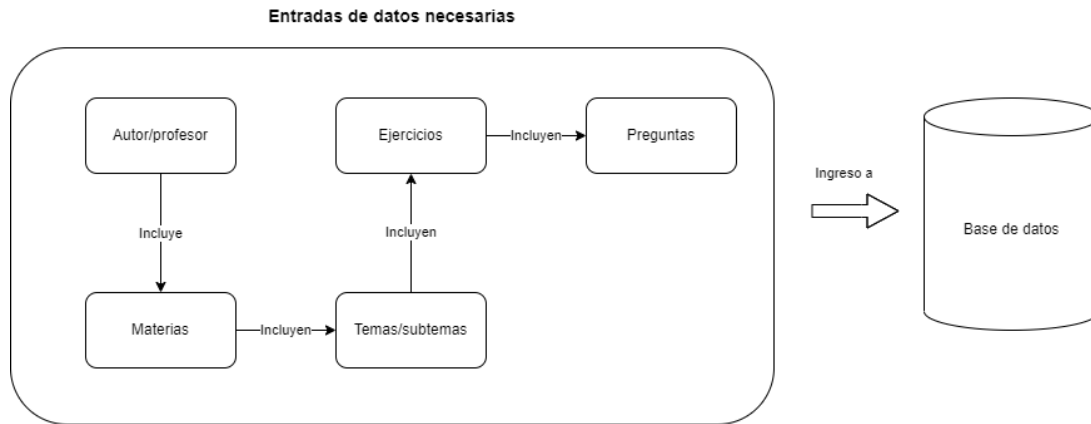
Partiendo de la correcta instalación del sistema computacional, en términos generales para poder generar exámenes automáticos el usuario debe completar dos procesos principales<sup>2</sup>:

- a) Proceso principal de entrada de datos: en la Figura No. 1 se presenta un esquema que menciona las entradas principales que se requieren a nivel de la interfaz de usuario para alimentar la base de datos.

---

<sup>2</sup> En el Anexo No.1 se presenta una guía ilustrada del software, con el fin de que el lector tenga una mejor noción de cómo luce el software y cómo funciona.

**FIGURA NO. 1: ENTRADA DE DATOS AL SOFTWARE**



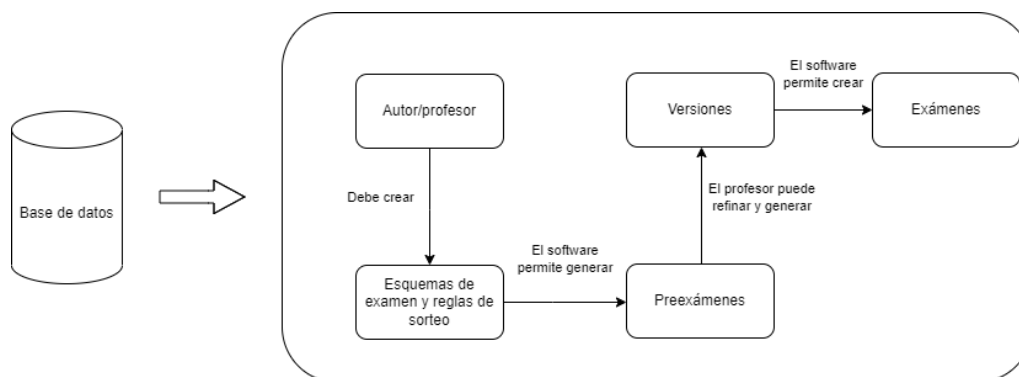
Fuente: elaboración propia

Siguiendo el flujo de la Figura No. 1, para comenzar a utilizar el software, se necesita incluir en el software un autor o profesor, quien es el actor que declara las materias para la base de datos. Estas materias contemplan a su vez temas y subtemas propios y una vez declarados esos temas y subtemas, se pueden incluir ejercicios y preguntas, que son la base para la generación de exámenes.

- b) Proceso de generación de exámenes: en la Figura No. 2, se muestra un esquema sobre la generación de exámenes con “The Examiner”.



FIGURA NO. 2: PROCESO DE GENERACIÓN DE EXÁMENES



Fuente: elaboración propia

Con base en la Figura No. 2 y partiendo de la premisa que se ha construido una base de datos poblada con al menos una materia, con temas y subtemas, preguntas y ejercicios, el autor/profesor debe seguir con la creación de un elemento que se define como “esquema” del examen que desea generar. Un “esquema” es una estructura que se construye sobre la base de una determinada materia, para la cual se escogen los temas y subtemas que se quieren evaluar y que incluye además la cantidad de preguntas que se deseen evaluar para cada uno de esos temas y subtemas. Adicionalmente, el autor/profesor debe parametrizar las reglas de “sorteo” para el proceso interno de selección del banco de preguntas.

El “sorteo” es un mecanismo por distribución aleatoria que utiliza el software para asignar a cada pregunta una determinada cantidad de “boletos” que le darán mayor o menor peso a la hora de ser elegidas en el proceso de selección automática para la construcción de exámenes. Estos “boletos” aplican para las siguientes características de cada pregunta: boletos por nivel de dificultad<sup>3</sup>, boletos por nivel de discriminación, boletos por novedad de la pregunta en la base de datos y boletos por la cantidad de estudiantes o usos que ha tenido la pregunta históricamente. Por defecto, el software asigna la misma cantidad de boletos al sorteo, pero en caso de querer parametrizarlo

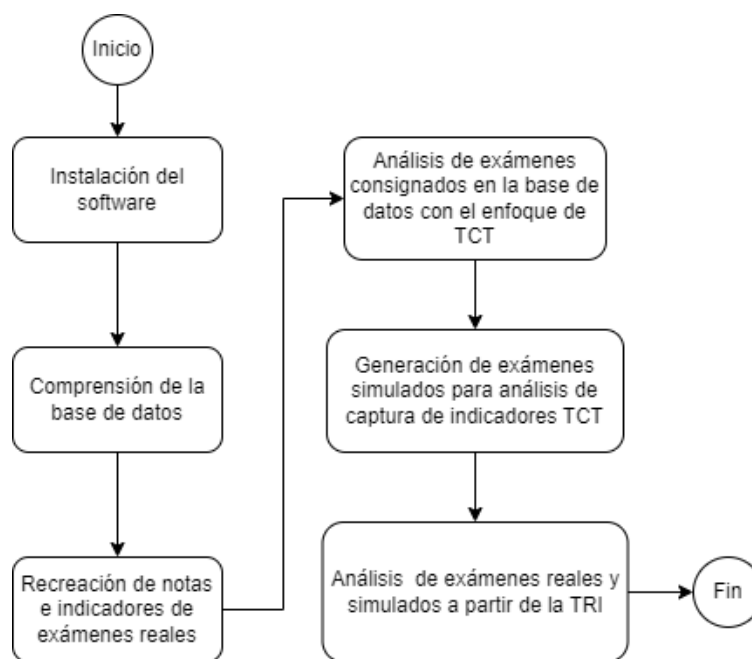
<sup>3</sup> Al ingresar una pregunta por primera vez, el criterio de dificultad es asignado por parte del autor/profesor y posteriormente se modifica con los valores reales de calificación del examen en el que la nueva pregunta sea utilizada.

de manera distinta, el autor/profesor podría manipular a discreción esas cantidades de “boletos”, dando mayor o menor cantidad de “boletos” a cualquiera de las características que se mencionaron.

### 3.2.2 Planteamiento del flujo de trabajo

En la Figura No. 3, se muestra el flujo de tareas que se plantea para poder realizar el proceso de generación de información y análisis de este trabajo.

FIGURA NO. 3: FLUJO DE TAREAS A REALIZAR



Fuente: elaboración propia

A continuación, el detalle de las tareas planteadas en la Figura No. 3:

- Instalación del software: la instalación limpia del software requiere el ambiente del sistema operativo Linux y la instalación de los paquetes mencionados en la sección de materiales.
- Comprensión de la base de datos y del código fuente: una vez instalado el software y el archivo de la base de datos, es necesario utilizar herramientas para

la visualización gráfica de las tablas, con el fin de comprender el funcionamiento en cuanto al almacenamiento de la información y adicionalmente, poder analizar de forma general el código fuente, para poder dar trazabilidad en cuanto a cálculos y funciones utilizadas a lo interno, con el fin de hacer comprobaciones y comparaciones de resultados posteriormente.

- Recreación de notas, indicadores de exámenes reales y análisis descriptivo: una vez entendida la forma de relación de almacenamiento en tablas y funcionamiento general del software, se utilizarán herramientas como el software R, hojas electrónicas y el software Python, para conseguir la recreación de resultados de exámenes y generar estadísticas descriptivas.
- Análisis de exámenes consignados en la base de datos con el enfoque de TCT: con un ambiente en el que se puedan recrear las notas de los exámenes reales, se procederá al análisis de parámetros con un enfoque de TCT.
- Generación de exámenes simulados para análisis de captura de indicadores TCT: adicionalmente a la revisión de exámenes reales consignados en la base de datos, se simularán pruebas por materias, para realizar análisis adicionales del enfoque de la TCT.
- Análisis de exámenes reales y simulados a partir de la TRI: se realizará el proceso de ajuste de los exámenes reales y simulados con un enfoque de la TRI, utilizando paquetes del software R (específicamente las librerías ltm, eRt, mirt).

### 3.2.3 Métodos relacionados con la TCT

Los métodos descritos a continuación tienen relación directa con la TCT y son elementos que están contemplados en la versión del software “The Examiner” que se estudió para este trabajo.

#### 3.2.3.1 Índice de dificultad del ítem

Una de las medidas que contempla “The Examiner” es el índice de dificultad, que se interpreta como la probabilidad de que una persona conteste de forma correcta un ítem  $i$ . Entonces, este índice asume valores entre 0 y 1, donde valores más cercanos al 1 indican preguntas más fáciles (en las que la proporción de respuestas correctas es alta) y valores más cercanos a 0 indican preguntas más difíciles (en las que la proporción de respuestas correctas es menor)

#### 3.2.3.2 El coeficiente alfa

El coeficiente alfa fue desarrollado por Cronbach en 1951 como una de las formas para medir la fiabilidad de una prueba y se representa por medio de la siguiente fórmula (Martínez Arias, Hernández Lloreda, & Hernández Lloreda, 2014):

$$\alpha = \frac{n}{n-1} \left( 1 - \frac{\sum_{j=1}^n \sigma_j^2}{\sigma_x^2} \right)$$

Donde:  $n$  es el número de elementos de una prueba,

$\sigma_j^2$  es la varianza de un ítem ( $j = 1, 2, \dots, n$ ),

$\sigma_x^2$  es la varianza total de la prueba.

El coeficiente alfa varía entre 0 y 1, donde valores más cercanos a 1 indican mejores resultados de fiabilidad de la prueba.

### 3.2.3.3 Índice de correlación biserial puntual

Se utiliza el índice de correlación biserial puntual ( $r_{pb}$ ) como medida de discriminación de un ítem. Se trata de un factor de correlación entre las respuestas dicotómicas de un ítem en relación con la calificación total de la prueba, donde puede asumir valores entre -1.0 y +1.0; valores cercanos a +1.0 indican que el ítem discrimina mejor entre las personas que tienen o no el conocimiento para responderlo de forma correcta. Un valor de 0 en el  $r_{pb}$  es un indicador que el ítem no discrimina entre personas con o sin el conocimiento para responderlo. El  $r_{pb}$  se calcula con la siguiente fórmula:

$$r_{pb} = \frac{M_1 - M_0}{S_n} \sqrt{\frac{n_1 n_0}{n^2}}$$

Donde:  $M_1$  es la media del puntaje global del examen incluyendo el grupo que contestó de manera correcta el ítem  $i$ ,

$M_0$  es la media del puntaje global del examen incluyendo el grupo que contestó de forma incorrecta el ítem  $i$ ,

$S_n$  es la desviación estándar de la prueba,

$n$  es la cantidad de personas que respondieron la prueba,

$n_1$  es la cantidad de personas que contestaron de forma correcta el ítem  $i$ ,

$n_0$  es la cantidad de personas que contestaron de forma incorrecta el ítem  $i$ .

### 3.2.4 Métodos relacionados con la TRI: modelo de Rasch

Anteriormente se abordaron los fundamentos del modelo de Rasch. En esta sección se mencionan los aspectos más de uso práctico del modelo como tal.

Se ha mencionado que el modelo de Rasch permite la medición conjunta de un rango latente de un individuo y un nivel de dificultad de un ítem.

Según el modelo de Rasch una persona que responda a un ítem equivalente a su umbral de competencia tendrá la misma probabilidad de responder de forma correcta o incorrecta, es decir que  $P(x=1) = P(x=0) = 0,50$  lo que significaría que:

$$\ln \left[ \frac{0,5}{1 - 0,5} \right] = \theta_n - \delta_i$$

$$\ln[1] = \theta_n - \delta_i$$

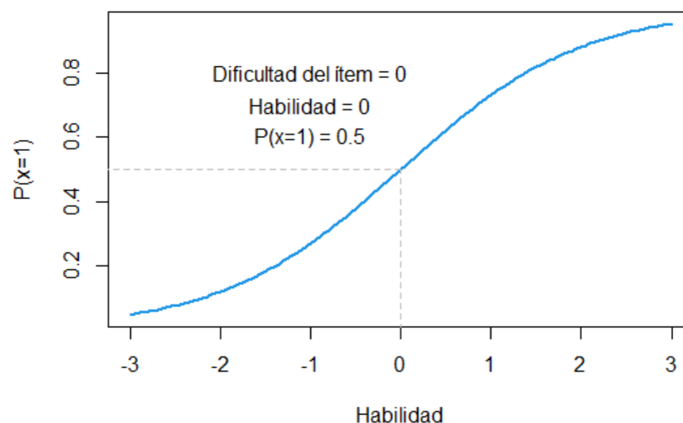
$$0 = \theta_n - \delta_i$$

Entonces, si la competencia del sujeto es mayor que la requerida por el ítem se cumple que  $(\theta_n - \delta_i) > 0$ ; y entonces la probabilidad de respuesta correcta sería mayor que la de respuesta incorrecta.

Por otro lado, si la competencia del sujeto es menor que la requerida por el ítem se cumple que  $(\theta_n - \delta_i) < 0$  y por consiguiente la probabilidad de respuesta correcta sería menor que la de respuesta incorrecta.

En los siguientes ejemplos gráficos se muestra el comportamiento de lo que se conoce como curvas características de los ítems, que son gráficos que muestran en el eje horizontal los valores de habilidad del individuo en escala *logit* y en el eje vertical la probabilidad de responder a un ítem de forma correcta.

**GRÁFICO NO. 1: EJEMPLO DE UNA CURVA CARACTERÍSTICA DE UN ÍTEM  
(CON DIFICULTAD DEL ÍTEM Y HABILIDAD IGUAL A 0)**

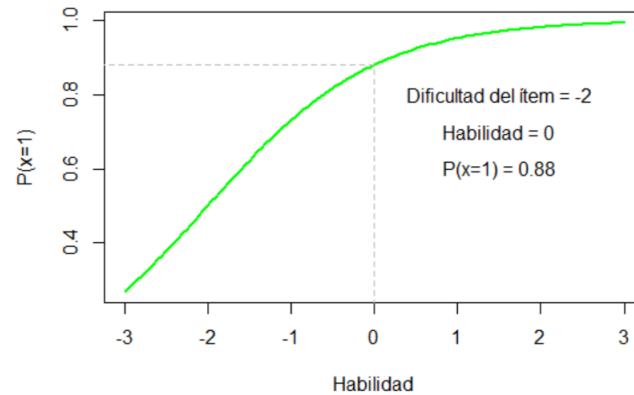


Fuente: elaboración propia

En el Gráfico No.1 se presenta el caso en el que una persona que presente un nivel de habilidad cero ( $\theta = 0$ ) en la escala *logit* y que el nivel de dificultad de la pregunta también sea cero ( $\delta = 0$ ), al aplicar la ecuación del modelo de Rasch, se genera una probabilidad de que su respuesta sea correcta de un 0.5. Esta es la situación “ideal” de discriminación de un ítem, en el que las personas en el umbral de su habilidad tienen 50% de probabilidad de contestar de forma correcta un ítem.

Por otro lado, suponiendo que la misma persona con nivel de habilidad o rasgo latente cero ( $\theta = 0$ ) se expone a una pregunta con nivel de dificultad -2 ( $\delta = -2$ ), aumenta la probabilidad de que su respuesta sea correcta a un valor de 0.88, como se muestra en el Gráfico No.2:

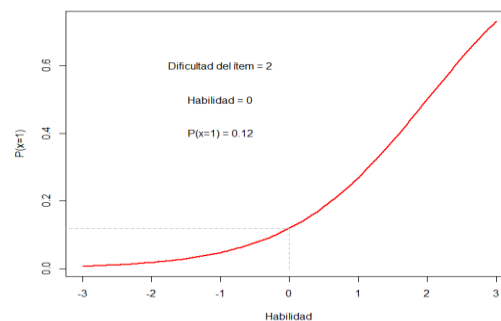
**GRÁFICO NO. 2: EJEMPLO DE UNA CURVA CARACTERÍSTICA DE UN ÍTEM**  
**(CON DIFICULTAD DEL ÍTEM = -2 Y HABILIDAD = 0)**



Fuente: elaboración propia

Finalmente, si la misma persona con nivel de habilidad o rasgo latente cero ( $\theta = 0$ ) se expone a un ítem con un nivel de dificultad 2 ( $\delta = 2$ ), disminuye la probabilidad de que su respuesta sea correcta a un valor de 0,12 como se muestra en el gráfico No.3:

**GRÁFICO NO. 3: EJEMPLO DE UNA CURVA CARACTERÍSTICA DE UN ÍTEM**  
**(CON DIFICULTAD DEL ÍTEM = 2 Y HABILIDAD = 0)**



Fuente: elaboración propia



## CAPÍTULO IV: RESULTADOS

En esta sección se presentan los resultados para cada una de las tareas consignadas en el flujo de tareas a realizar de la figura No.3.

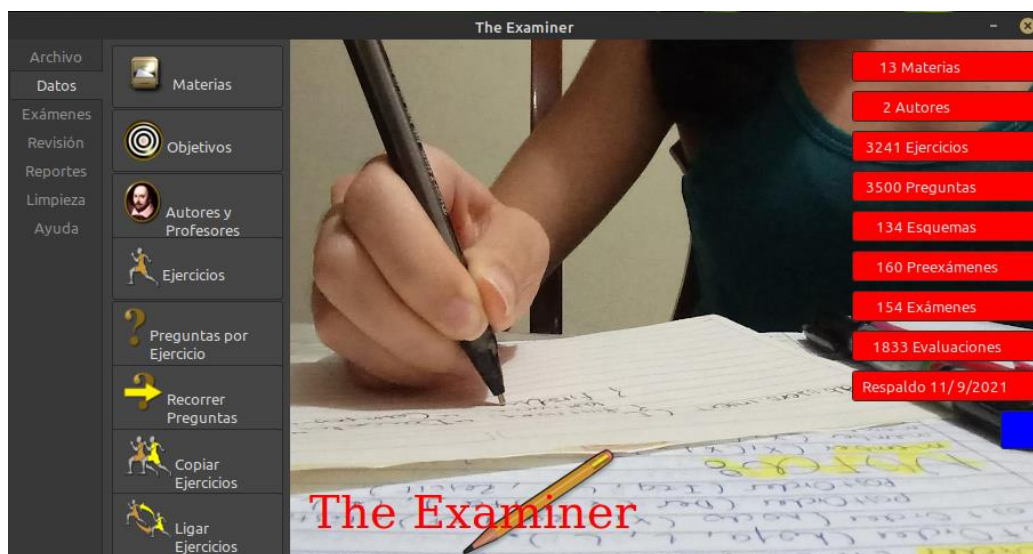
### 4.1 Instalación del software

El software en su versión actual requiere un sistema operativo Linux para lograr la instalación de los elementos requeridos, mencionados en apartados anteriores.

El resultado de esta tarea fue positivo y se logró poner en funcionamiento el programa, así como la restauración del archivo de respaldo de la base de datos suministrada a la fecha.

En la Figura No. 4 se muestra la pantalla de inicio del software en funcionamiento:

FIGURA NO. 4: PANTALLA PRINCIPAL DEL SOFTWARE EN FUNCIONAMIENTO



Fuente: “The Examiner”

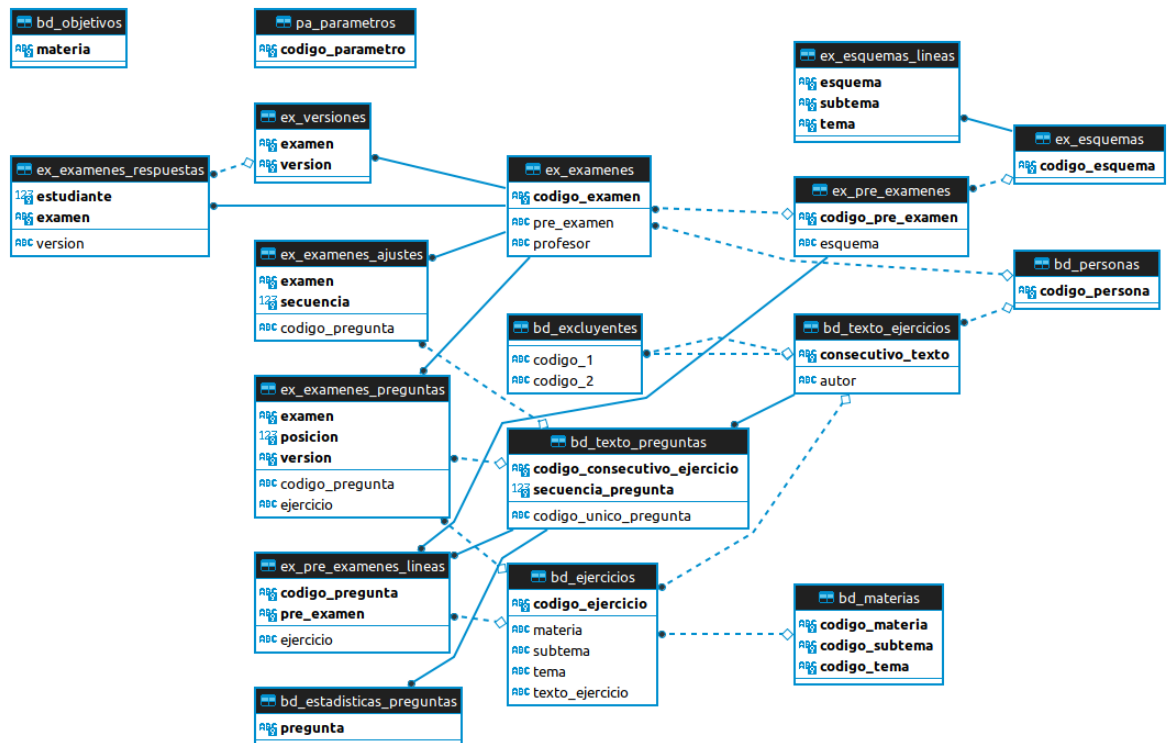
## 4.2 Comprensión de la base de datos y del código fuente

Con el software en funcionamiento, el siguiente paso consistió en explorar la construcción de la base de datos, con el fin de identificar las tablas clave para el análisis de la información histórica de los exámenes aplicados, así como el funcionamiento a lo interno del sistema.

### 4.2.1 Resumen del modelo de la base de datos

El modelo de la base de datos cuenta con 17 tablas. En la Figura No. 5 se puede observar un diagrama del modelo de la base de datos. Esta tarea fue necesaria para poder entender los flujos de la información en el software, tanto para la construcción de exámenes, así como su almacenamiento de datos, con el fin de comprender los algoritmos del sistema y poder recrear calificaciones.

FIGURA NO. 5: MODELO DE LA BASE DE DATOS



Fuente: base de datos “The Examiner”

A continuación, se agrupan las tablas según su utilidad y se presenta una descripción breve de las mismas:

- Tablas para el almacenamiento de información de ejercicios y preguntas:
  - bd\_personas: almacena la información del usuario del sistema (autor o profesor).
  - bd\_materias: almacena las materias, temas y subtemas ingresados por el usuario.
  - bd\_ejercicios y bd\_texto\_ejercicios: almacenan la información de los ejercicios, ligados a la tabla de materias.
  - bd\_texto\_preguntas: almacena los detalles de las preguntas asociadas a los ejercicios.
  - bd\_estadisticas\_preguntas: almacena la información de las preguntas conforme se registran datos históricos de los exámenes aplicados.
- Tablas para la creación de esquemas:
  - ex\_esquemas y ex\_esquemas\_lineas: almacenan los esquemas o estructuras de los exámenes que el autor/profesor desea construir (materia, temas y subtemas a evaluar, así como la cantidad de preguntas)
- Tablas para la creación de preexámenes y exámenes
  - ex\_pre\_examenes y ex\_pre\_examenes\_lineas: almacenan la información de los preexámenes que el autor/profesor decide fijar como candidatos a ser transformados en exámenes
  - ex\_examenes, ex\_examenes\_preguntas, ex\_versiones: almacenan la información identificación de los exámenes, la estructura de las preguntas que incluyen y en el caso de versiones, las diferentes posiciones en las que el software baraja las opciones de respuesta.
  - ex\_examenes\_respuestas: almacena las respuestas de los estudiantes para construir las calificaciones.
  - ex\_examenes\_ajustes: almacena en caso de ser necesario, ajusta a los exámenes debidos a situaciones particulares de las preguntas.

- Tablas misceláneas no utilizadas para este análisis:
  - pa\_parametros: almacena parámetros principales para el funcionamiento del software
  - bd\_objetivos: almacena información opcional de objetivos de aprendizaje
  - bd\_excluyentes: tabla de uso interno del sistema

#### **4.2.2 Comprensión de la organización y estructura del código fuente:**

La siguiente tarea preparatoria y fundamental para la realización de análisis de datos, consistió en comprender la organización y estructura de los códigos fuente del software, con el fin de poder dar trazabilidad a los algoritmos. Los archivos fuente están desarrollados en lenguaje C. A continuación, se presenta un listado de los nombres de los archivos fuente y una descripción breve de cada una de sus funcionalidades:

- Archivo EX: Archivo principal
- Archivo EX100: Actualización de Parámetros
- Archivo EX1010: Respalda la base de datos
- Archivo EX1020: Instala nuevas versiones
- Archivo EX2000: Actualización Archivo de Materias
- Archivo EX2010: Actualización Objetivos de Cursos
- Archivo EX2020: Actualización Archivo de Personas
- Archivo EX2030: Actualización Archivo de Ejercicios
- Archivo EX2040: Actualización Archivo de Preguntas
- Archivo EX2050: Recorre Preguntas por Materia, Tema Y Subtema
- Archivo EX2060: Copia Ejercicios
- Archivo EX2070: Crea ligas entre ejercicios
- Archivo EX3000: Crea un esquema de examen
- Archivo EX3010: Actualiza parámetros para la asignación de boletos a cada pregunta
- Archivo EX3020: Genera un preexamen a partir de un esquema predefinido
- Archivo EX3030: Refina un preexamen
- Archivo EX3040: Convierte un preexamen en examen
- Archivo EX3050: Imprime versiones de un examen
- Archivo EX4000: Revisa exámenes
- Archivo EX4010: Analiza exámenes
- Archivo EX4020: Actualiza base de datos de preguntas con la información de un examen ya ejecutado
- Archivo EX5000: Lista materias, temas y subtemas “The Examiner”
- Archivo EX5010: Lista ejercicios registrados

- Archivo EX5020: Análisis de la base de datos
- Archivo EX5030: Cruza dos variables asociadas a las preguntas y permite listar las preguntas en cada cruce (de momento limitado a dificultad vs. Discriminación)
- Archivo EX5040: Lista esquemas, preexámenes y exámenes
- Archivo EX6000: Borra exámenes viejos, revisa estado del examen y ofrece opciones de borrado
- Archivo EX6010: Elimina preexámenes viejos
- Archivo EX6020: Elimina esquemas viejos
- Archivo EX6030: Borra todos los ejercicios de un subtema, un tema o una materia

### 4.3 Análisis descriptivos, recreación de notas e indicadores de exámenes reales

Partiendo de la comprensión de las relaciones entre las tablas de la base de datos y una idea clara de la estructura de desarrollo del software, el siguiente paso consistió en explorar la base de datos y recrear un examen, para validación de los elementos de cálculo de notas.

#### 4.3.1 Análisis descriptivos

En el Cuadro No. 1 se muestra un resumen de la base de datos en cuanto a materias<sup>4</sup>, cantidad de exámenes por cada materia y total de calificaciones consignadas en la base de datos.

**CUADRO NO. 1: MATERIAS, EXÁMENES Y CALIFICACIONES EN LA BASE DE DATOS**

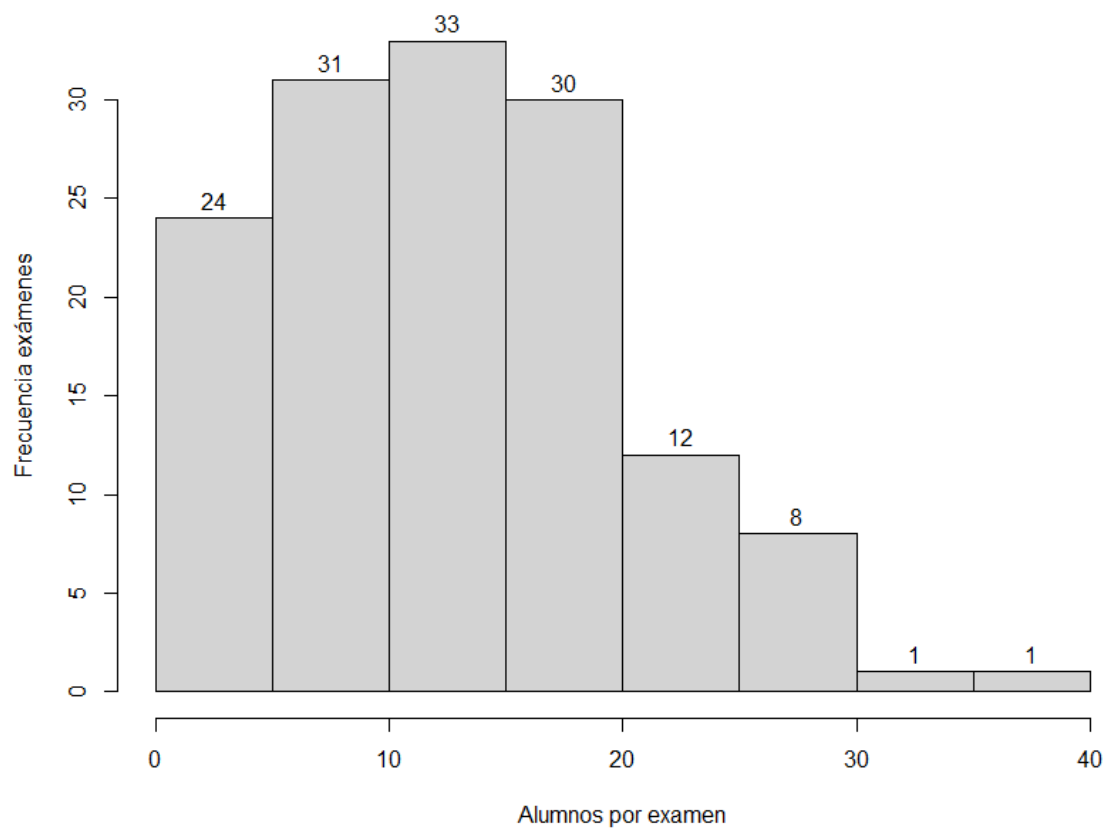
Código Materia	Nombre Materia	Exámenes por materia	Calificaciones por materia
COMP	Compiladores e Intérpretes	33	434
IO	Investigación de operaciones	30	272
ININ	Introducción a la Investigación	27	237
FOC	Fundamentos de Organización de Computadoras	17	273
AA	Análisis de Algoritmos	13	259
CG	Introducción a los Gráficos por Computadora	13	126
BMC	Biología Molecular Computacional	12	108
PCB	Paradigmas Computacionales de Bioinformática	7	78
MCD	Matemáticas para Ciencia de los Datos	2	46
<b>Total base de datos</b>		<b>154</b>	<b>1833</b>

Fuente: información consignada en la base de datos “The Examiner”

<sup>4</sup> Las materias en cuestión corresponden a cursos de programas impartidos por el Instituto Tecnológico de Costa Rica como Ingeniería en Computación y Maestría en Computación, así como del programa de Maestría en Bioinformática de la Universidad de Costa Rica.

En el Gráfico No. 4 se puede apreciar la cantidad de alumnos que respondieron por examen. El promedio de estudiantes evaluados por examen es de 13 estudiantes, el 84% de los exámenes se aplicaron a 20 o menos estudiantes por grupo.

**GRÁFICO NO. 4: HISTOGRAMA ALUMNOS POR EXAMEN**



Fuente: generado a partir de información de la base de datos "The Examiner".

### **4.3.2 Recreación de notas del primer examen consignado en la base de datos e indicadores de referencia generados por el software para un examen de prueba.**

#### **4.3.2.1 Recreación de notas del primer examen consignado en la base de datos.**

Como punto de partida, se realizó la trazabilidad del primer examen consignado en la base de datos, aplicado a 9 estudiantes del curso de Biología Molecular Computacional. Dicho examen contó con 55 preguntas y tres versiones.

Básicamente, una vez realizada la prueba, el autor/profesor ingresa las respuestas de los estudiantes por medio de una interfaz, que va alimentando la base de datos. En este punto se va construyendo en la tabla de respuestas para cada estudiante, una hilera de caracteres, que almacena las respuestas que el estudiante contestó. Cada pregunta tiene 5 opciones de respuesta (A, B, C, D, E), de forma que la hilera contará con la cantidad de letras igual a la cantidad de preguntas que respondió el estudiante en el examen. Adicionalmente, se tuvo que tomar en consideración que el examen que responde cada estudiante puede tener versiones diferentes, por lo que se tuvo que desarrollar un procedimiento utilizando el programa R para conectarse a la base de datos Postgres, con el fin de poder extraer información de las tablas que incluyen exámenes, versiones de exámenes y respuestas, para poder identificar la posición de las respuestas de acuerdo a la versión específica que responde un alumno particular y así poder con eso reconstruir la calificación del alumno para su validación. En el Anexo No.2 se presenta el código desarrollado para el procedimiento mencionado.

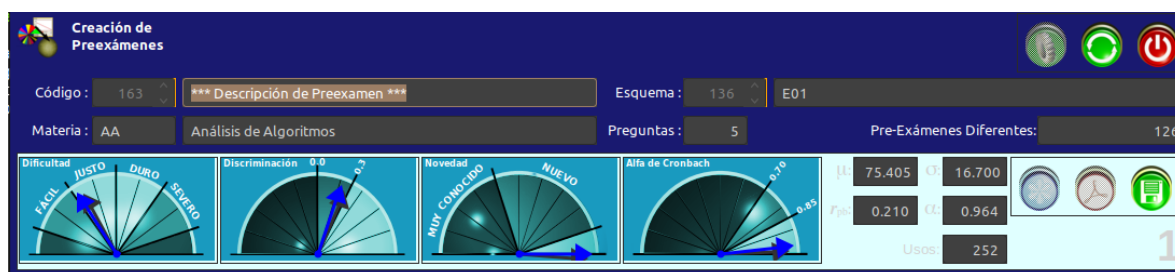
Con el procedimiento, se lograron recrear las calificaciones de los estudiantes para el examen específico, independientemente de la versión, con una media general del examen de 75,96 puntos. Con esta validación, lo siguiente fue generar un examen de prueba, para poder dar trazabilidad a los cálculos de indicadores que genera el software, como se presenta en el siguiente apartado.

### 4.3.2.2 Indicadores de referencia generados por el software para un examen de prueba.

Cuando se crea un examen, el software presenta una salida a través de la interfaz gráfica en la que se muestran algunos valores que sirven de referencia para tomar decisiones en cuanto a cuáles preexámenes son de interés por parte del autor/profesor para convertirlos en exámenes, los cuales son: media, desviación estándar, coeficiente alfa y coeficiente biserial puntual ( $r_{pb}$ ).

Entonces, se realizó un análisis para entender cómo calcula el software esos indicadores y para esto, se creó un usuario y se generó un nuevo examen de prueba en la base de datos. El examen de prueba tiene como identificador el número 163 y en este caso se diseñó con una única versión y 5 preguntas escogidas a discreción. Luego, siguiendo la lógica del software (1) se creó un esquema de examen, (2) se creó de preexamen, (3) se convirtió el preexamen en examen. El software le calculó una media esperada de 75.41 puntos, una desviación esperada de referencia de 16.7, un coeficiente alfa general de referencia de 0.964 y un  $r_{pb}$  de referencia de 0.21. En la Figura No. 6, se muestra la salida gráfica del software una vez generada la estructura del examen; adicionalmente, el anexo No.3 se presenta el archivo que se genera en PDF que ilustra cómo se visualizaría el examen.

FIGURA NO. 6: VISTA EXAMEN DE PRUEBA PARA CÁLCULO DE INDICADORES



Fuente: imagen generada a partir de la creación de un examen en “The Examiner”.

Con esta información, una tarea clave consistió en poder verificar los métodos de cálculo del software a nivel del código fuente, con el fin de poder generar análisis. A continuación, se muestra cómo se realizaron estas verificaciones.



a) Confirmación del cálculo de la media esperada:

El examen de prueba incluyó los ítems 000415, 002897, 002893 002891 y 002894. Se realizó su ubicación en la tabla de estadísticas de preguntas, con la siguiente información utilizada para el cálculo de la media esperada y que se presenta en el Cuadro No. 2:

**CUADRO NO. 2: VALORES PARA CÁLCULO DE MEDIA ESPERADA PARA EXAMEN DE PRUEBA**

Pregunta	Ultimo examen	Año último uso	Estudiantes que la han respondido	Cantidad respuestas correctas	Cantidad respuestas incorrectas	Dificultad ítem	Coficiente alpha	Coficiente biserial puntal
000415	00090	2018	67	58	9	0,866	0,940148	0,148506
002897	00081	2018	19	18	1	0,947	0,983496	-0,00905
002893	00110	2019	41	29	12	0,707	0,980746	0,447343
002891	00122	2019	66	59	7	0,894	0,966793	0,189257
002894	00110	2019	59	21	38	0,356	0,969093	0,208742

Fuente: información extractada de la tabla de estadísticas de preguntas de “The Examiner”, para el examen de prueba.

La media esperada para el examen, el software la calcula de la siguiente forma:

$$media\ esperada = \left[ \frac{suma\ de\ dificultades\ de\ ítems}{total\ de\ ítems} \right] * 100$$

Donde: *dificultad del ítem* para cada ítem, se calcula dividiendo la cantidad de respuestas correctas entre el número total de respuestas.

Entonces en este caso, la media esperada que se calcula es de 75,40 replicando el cálculo que se presenta en la interfaz gráfica de forma automática.

- b) Confirmación de la desviación estándar de referencia de los ítems a partir de la información histórica de la base de datos:

Como una medida de referencia, el software calcula una desviación estándar del conjunto de ítems escogidos para la prueba. Realizando la búsqueda de su procedimiento y forma de cálculo en los archivos de código fuente, se determinó que utiliza un escalamiento que se ejecuta de la siguiente forma:

$$desviación\_est\_conjunto\_ítems = \sqrt{\left[ \sum_{i=1}^n p_i(1 - p_i) \right] * \left[ \frac{100}{n} \right]^2}$$

Donde:  $p_i$  es el índice de dificultad de cada ítem y  $n$  se refiere a la cantidad total de ítems.

En este caso, con base en la información que se extrajo de la base de datos para los ítems del examen de prueba y la fórmula anterior, se calcula una desviación estándar de referencia para el conjunto de los 5 ítems elegidos de 16.69, que equivale efectivamente al valor presentado de forma automática en la interfaz gráfica del software.

- c) Confirmación del coeficiente alfa de referencia para el examen de prueba:

Conforme se ingresan calificaciones al software, la base de datos va almacenando los valores del coeficiente alfa para cada examen e ítem incluido y genera un cálculo de referencia que se confirmó con el procedimiento en el código fuente y que se realiza de la siguiente forma:

$$coeficiente\ alfa\ referencia = \frac{\sum_{i=1}^n [(x_i + 1) * \alpha_i]}{\sum_i^n (x_i + 1)}$$

Donde:  $x_i$  equivale a la cantidad de personas  $x$  que han respondido al ítem  $i$ ,  $\alpha$  corresponde al coeficiente alfa histórico del examen incluido el ítem y  $n$  es la cantidad de ítems incluidos en el examen.

En este caso, se confirmó que, con base en la información de los 5 ítems escogidos para el examen de prueba, el valor del coeficiente alfa mostrado en la salida de la interfaz gráfica efectivamente se calculó con un valor de 0.964.

d) Confirmación del valor de  $r_{pb}$  de los ítems a partir de la información histórica de la base de datos:

De forma similar al cálculo del valor de referencia del coeficiente alfa, el software calcula un valor de referencia del  $r_{pb}$ , que se comprobó a nivel del código fuente y que se puede expresar con la siguiente fórmula:

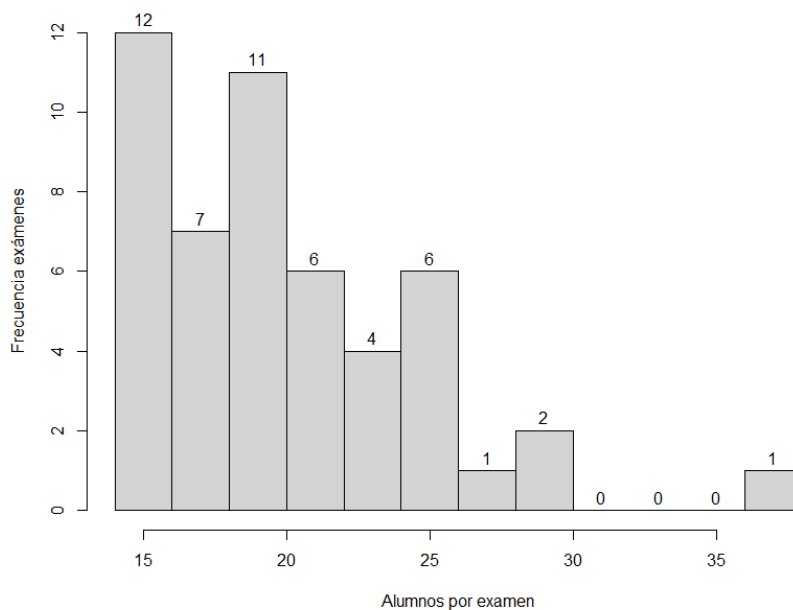
$$r_{pb\_referencia} = \frac{\sum_{i=1}^n [(x_i + 1) * (r_{pb})_i]}{\sum_i^n (x_i + 1)}$$

Donde  $x_i$  equivale a la cantidad de personas  $x$  que han respondido al ítem  $i$ , el valor de  $r_{pb}$  corresponde al valor del coeficiente biserial puntual histórico del ítem  $i$  consignado en la base de datos y  $n$  la cantidad de ítems incluidos en el examen de prueba. El valor calculado de  $r_{pb}$  de referencia fue de 0,21 de forma que se pudo confirmar que es el mismo valor que presenta el software de forma automática en la interfaz gráfica.

### 4.3.3 Análisis de exámenes reales consignados en la base de datos con un enfoque de TCT.

Con la comprensión de los cálculos internos de software, se procedió con el análisis de los exámenes reales consignados en la base de datos. Aquí cabe mencionar que una situación que se valoró fue la cantidad de alumnos que contestaron cada prueba, ya que hay que tomar en consideración que, por tratarse de pruebas de aula, el grupo que participa podría tratarse de un grupo con pocos estudiantes. Luego, con la exploración de la base de datos se tomó la decisión de incluir los 50 exámenes con mayor cantidad de estudiantes evaluados, que corresponden a exámenes con 15 o más estudiantes. Como se mencionó, esta situación es comprensible debido a la cantidad de alumnos por curso que típicamente se matriculan en un curso de carrera para un periodo académico determinado. A continuación, se presenta un histograma que refleja esta segmentación de los 50 exámenes con mayor cantidad de estudiantes:

**GRÁFICO NO. 5: HISTOGRAMA 50 EXÁMENES CON  
MAYOR PARTICIPACIÓN DE ESTUDIANTES**



Fuente: base de datos de “The Examiner”,

#### **4.3.3.1 Análisis de la media real vs. media esperada para los 50 exámenes con mayor participación de estudiantes.**

Con los 50 exámenes de 15 o más alumnos, se calcularon intervalos de confianza del 95% para la media de las calificaciones reales y se ubicó el valor de la media esperada que se generó por medio del software, con el fin de verificar qué porcentaje de esos valores se ubicaron dentro del intervalo de confianza construido, como una forma de evaluar el indicador pronosticado y se comprobó que en 42 exámenes de los 50 (es decir en el 84% de los casos), el valor de la media esperada quedó contenido en el intervalo de confianza para el examen particular y en el resto de los casos, es decir 8 de 50 exámenes ( o lo mismo que el 16%) quedaron por fuera de los límites de los intervalos de confianza correspondientes<sup>5</sup>.

En el Cuadro No. 3 se presenta un resumen de los resultados anteriores:

---

<sup>5</sup> En el anexo No.4, se presenta el código del procedimiento utilizado para esta sección.

**CUADRO NO. 3: RESULTADOS INTERVALOS DE CONFIANZA EXÁMENES REALES ANALIZADOS**

<b>examen</b>	<b>Límite Inferior IC</b>	<b>Media Examen</b>	<b>Límite superior IC</b>	<b>Desviación Estándar</b>	<b>Media esperada</b>	<b>En intervalo</b>
ex00002	62,4	68,3	74,2	13,1	68,6	Sí
ex00004	62,2	68,5	74,8	13,6	66,6	Sí
ex00008	62,5	69,1	75,7	13,9	68,8	Sí
ex00009	61,6	67,4	73,2	12,3	65,9	Sí
ex00039	66,2	69,6	73,1	10,9	66,5	Sí
ex00044	62,0	66,0	70,1	9,9	68,7	Sí
ex00048	63,2	67,9	72,7	9,3	69,3	Sí
ex00050	67,8	71,4	74,9	7,0	69,8	Sí
ex00065	60,3	65,1	69,9	10,7	65,9	Sí
ex00068	59,3	64,4	69,4	10,3	69,0	Sí
ex00075	57,8	66,0	74,2	16,1	65,9	Sí
ex00080	56,9	62,0	67,1	11,8	65,0	Sí
ex00081	52,0	58,0	64,0	13,4	52,6	Sí
ex00083	62,4	69,0	75,7	14,4	67,3	Sí
ex00085	62,6	68,1	73,6	12,2	66,7	Sí
ex00089	56,5	61,4	66,3	12,0	61,8	Sí
ex00090	55,5	60,1	64,7	11,2	60,1	Sí
ex00091	55,9	62,7	69,5	13,9	66,7	Sí
ex00092	58,7	64,6	70,5	11,7	67,6	Sí
ex00095	52,1	58,7	65,2	14,6	58,2	Sí
ex00098	67,8	72,6	77,4	9,9	70,7	Sí
ex00101	52,1	58,1	64,1	13,0	56,8	Sí
ex00103	65,3	70,4	75,5	11,5	66,4	Sí
ex00105	60,2	66,1	71,9	15,1	67,4	Sí
ex00106	50,3	56,4	62,5	15,6	62,0	Sí
ex00108	59,1	65,8	72,4	14,8	64,0	Sí
ex00112	61,4	64,5	67,5	8,5	64,0	Sí
ex00120	55,9	63,2	70,4	16,6	64,5	Sí
ex00121	62,3	66,7	71,2	12,4	66,3	Sí
ex00124	60,5	66,5	72,5	12,2	66,2	Sí
ex00125	55,6	61,6	67,7	13,8	65,5	Sí
ex00126	60,0	65,7	71,4	12,3	69,5	Sí
ex00127	65,5	70,7	75,9	10,6	68,4	Sí
ex00128	59,4	64,6	69,8	11,8	61,3	Sí
ex00130	59,4	67,6	75,9	17,9	72,9	Sí
ex00131	65,5	71,8	78,0	15,0	74,8	Sí
ex00137	63,4	66,8	70,3	6,8	65,1	Sí
ex00138	53,2	57,2	61,3	9,5	61,4	Sí
ex00139	56,8	61,3	65,9	11,8	63,9	Sí
ex00140	70,0	75,9	81,7	11,9	70,9	Sí
ex00146	53,8	59,4	65,0	12,8	61,6	Sí
ex00147	47,0	57,5	68,1	21,6	62,6	Sí
ex00060	67,4	72,7	77,9	12,3	66,2	No
ex00096	49,1	54,6	60,0	12,7	63,2	No
ex00102	50,6	54,6	58,7	10,6	60,4	No
ex00110	45,4	50,5	55,7	12,7	59,9	No
ex00113	50,3	54,2	58,1	10,4	61,4	No
ex00117	52,5	57,5	62,5	13,1	63,6	No
ex00122	50,0	55,1	60,2	12,0	67,4	No
ex00135	58,9	63,2	67,4	11,1	67,9	No

Fuente: intervalos construidos a partir de información de la base de datos de “The Examiner”, para los 50 exámenes con mayor participación de estudiantes.

#### 4.3.3.2 Análisis del coeficiente alfa real vs. coeficiente alfa de referencia para los 50 exámenes con mayor participación de estudiantes.

Para la construcción de intervalos del 95% de confianza del coeficiente alfa, se utilizó la librería Cronbach de R, específicamente con su función cron.ci y se detectó que el 100% de los valores esperados estaban por fuera de los intervalos de confianza.

Dado lo anterior, se procedió a realizar una validación a nivel de código fuente sobre el mecanismo de cálculo del coeficiente alfa, con el siguiente hallazgo: el cálculo en el coeficiente alfa generado por el software, utiliza la varianza de las notas obtenidas por los estudiantes, en el lugar de la varianza total del instrumento con base en la cantidad de respuestas correctas eliminando los ítems que no presentan variabilidad.

Esto se confirmó con una corrida de cálculo del coeficiente alfa para el primer examen de la base de datos. En el Cuadro No. 4 se presentan los insumos para el cálculo del coeficiente alfa real y para el coeficiente alfa según el software:

**CUADRO NO. 4: VARIANZA SEGÚN CANTIDAD DE RESPUESTAS BUENAS POR ESTUDIANTE Y VARIANZA SOBRE NOTA BASE 100**

<b>Estudiante</b>	<b>Cantidad respuestas buenas</b>	<b>Nota base 100</b>
1	26	70,3
2	24	64,9
3	27	73,0
4	30	81,1
5	26	70,3
6	22	59,5
7	30	81,1
8	16	43,2
9	22	59,5
<b>Varianza por columna</b>	<b>17,3</b>	<b>126,3</b>

Fuente: cálculo sobre examen 001 de la base de datos de “The Examiner”

Entonces, aplicando la fórmula para cálculo del coeficiente alfa:

$$\alpha = \frac{n}{n-1} \left( 1 - \frac{\sum \sigma_j^2}{\sigma_x^2} \right)$$

Donde:  $n$  es el número de elementos de una prueba,

$\sigma_j^2$  es la varianza de un ítem ( $j = 1, 2, \dots, n$ ),

$\sigma_x^2$  es la varianza total de la prueba.

Sabiendo que la sumatoria de las varianzas individuales de los ítems es de 6,247 tenemos que el coeficiente alfa calculado por el software a partir de la varianza de las notas con base 100 de los estudiantes es de:

$$\alpha = \frac{37}{37-1} \left( 1 - \frac{6,247}{126,3} \right)$$

$$\alpha = 0,977$$

Y si se realiza el cálculo del coeficiente alfa tomando en consideración la varianza calculada a partir de la cantidad de respuestas correctas, el valor corregido es de:

$$\alpha = \frac{37}{37-1} \left( 1 - \frac{6,247}{17,3} \right)$$

$$\alpha = 0,656$$

Cabe mencionar que este valor de 0,656 también se comparó con la salida de cálculo de coeficiente alfa del paquete Cronbach de R, como forma de validación del cálculo manual.



A partir de este hallazgo, se realizó se corrió un procedimiento para la construcción de intervalos de confianza del coeficiente alfa para los 50 exámenes analizados (en el anexo No.4) se presenta el código del procedimiento utilizado para esta sección) y se categorizaron sus valores centrales según Sheu et. Al (2005), donde valores superiores a 0,9 se consideran excelentes, valores en 0,8 y 0,9 se consideran buenos, entre 0,7 y 0,8 aceptables, entre 0,6 y 0,7 cuestionables, entre 0,5 y 0,6 pobre y menos de 0,5 inaceptables. Los resultados se muestran en el Cuadro No. 5:

**CUADRO NO. 5: CATEGORÍA COEFICIENTES ALFA CORREGIDOS**

<b>Categoría coeficiente alpha</b>	<b>Cantidad de exámenes</b>	<b>Porcentaje Acumulado</b>
Bueno ( $0,8 < \alpha < 0,9$ )	18	36%
Aceptable ( $0,7 < \alpha < 0,8$ )	18	36%
Cuestionable ( $0,6 < \alpha < 0,7$ )	11	22%
Pobre ( $0,5 < \alpha < 0,6$ )	1	2%
Inaceptable ( $0,5 < \alpha$ )	2	4%
<b>Total</b>	<b>50</b>	<b>100%</b>

Fuente: cálculos a partir de base de datos "The Examiner"

Se puede ver que sumando los exámenes con coeficientes alfa corregidos que se pueden categorizar como buenos o aceptables, el resultado es de 36 exámenes (es decir el 72%), 11 de los exámenes (o lo que significa el 22%) presentan una condición cuestionable y el resto, es decir 3 de los 50 exámenes (equivalente a un 6%) presentan una condición de pobre o inaceptable.

Con esta revisión se puede concluir que a pesar del hallazgo sobre el cálculo de los coeficientes alfa a nivel del software, el resultado de los coeficientes alfa corregidos, arrojan resultados bastante positivos.

#### **4.3.3.3 Análisis de la desviación estándar como valor de referencia para la generación de exámenes:**

La desviación estándar de las calificaciones de un examen, se han utilizado para la construcción de intervalos de confianza para la media y comparar ese segundo valor con el valor puntual de referencia que brinda el software. En cuanto al valor de referencia que genera el software respecto a la desviación estándar para un examen, el cálculo que se mostró en la sección 4.3.2.2 se trata de una forma de escalar una magnitud utilizando la sumatoria de los valores de varianza del ítem calculados por medio de las proporciones de aciertos y desaciertos de un ítem y luego calculándole la raíz cuadrada a esa sumatoria de varianzas, lo cual no es comparable con la desviación estándar producto de las calificaciones obtenidas por los estudiantes que participaron en el test. Es por lo que, no se realizará mayor profundización del valor de la desviación estándar como aspecto puntual para la generación de exámenes y toma de decisiones a la hora de escogencia de un preexamen particular.

#### **4.3.3.4 Análisis del coeficiente de correlación biserial puntual.**

Según (Martínez Arias, Hernández Lloreda, & Hernández Lloreda, 2014), una situación frecuente en análisis de ítems es relacionar el resultado del ítem dicotómico con la puntuación del test por medio del coeficiente de correlación biserial puntual y según (Meneses, y otros, 2013), “es muy sencillo ver que una alta correlación, cercana a 1, indica una gran discriminación del ítem, que valores cercanos a  $-1$  indican lo contrario (donde los buenos fallan el ítem y los malos lo aciertan) y que valores cercanos a 0 indican que nada tiene que ver acertar este ítem con el conocimiento que mide el conjunto de la prueba”.

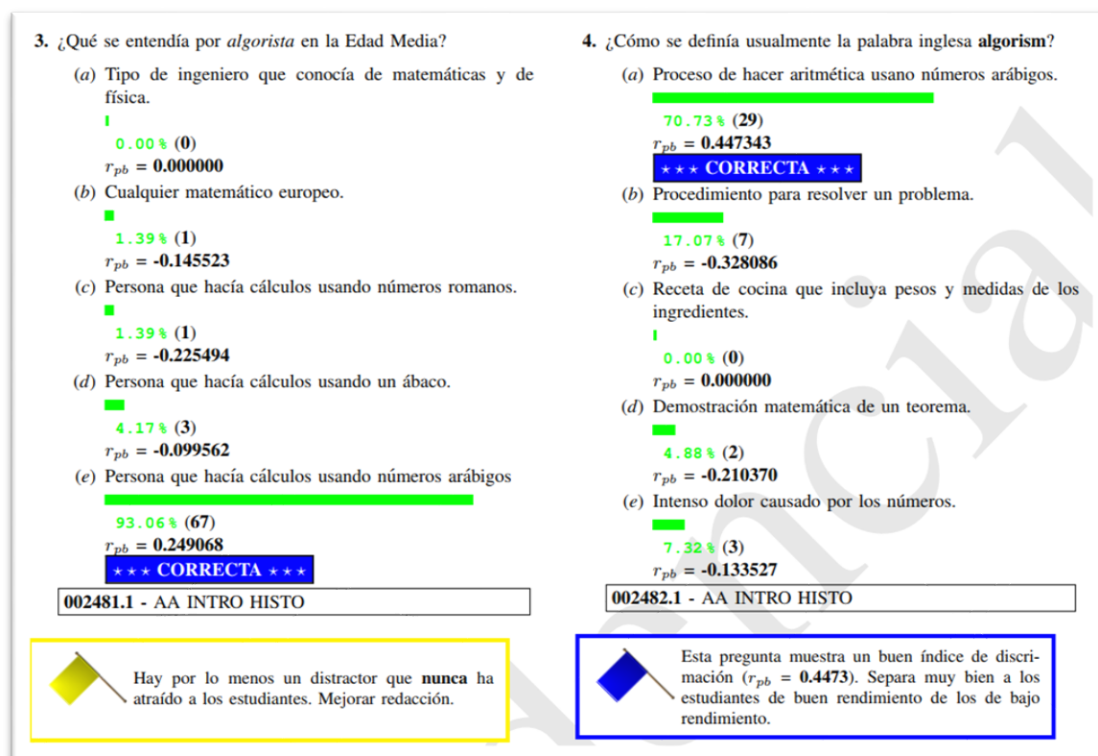
Ahora, cuando se genera un examen en el software, los ítems muestran sus valores históricos del  $r_{pb}$ , ya sean valores positivos o negativos. Esto se comprobó con el examen de prueba y se confirmó que uno de los ítems (00081), mostró un valor de correlación biserial puntual de  $-0,00905$ , que implicaría ser un ítem para no tomar en consideración. Luego, el valor de referencia de  $r_{pb}$  que se presenta como parte de la toma de decisiones para la elección de un examen, pondera de forma general los ítems como se mostró en secciones anteriores, lo cual aporta una referencia a nivel general para la comparación de preexámenes que se pueden generar con el software. Por medio de la función `cor.bis` del paquete `ltm` de R, se

analizó un examen real (el examen 00004, con el código que se puede observar en el anexo No.4), para fines de evaluación de ese grupo de ítems.

Se utilizó como criterio de decisión, un valor de  $r_{pb}$  de 0,2 para determinar si un ítem es aceptable o no en términos de discriminación. Los resultados de la revisión de los valores de  $r_{pb}$  de ese examen particular de 55 ítems son: 40 ítems con  $r_{pb}$  mayor a 0,2 (lo que significa el 73% de los casos), 14 ítems con valores menores a 0,2 (lo que significa el 25% de los casos) y 1 caso que no se toma en cuenta porque fue respondido de forma correcta por el 100% de los participantes.

Para la construcción de los preexámenes, cabe mencionar que el software habilita una serie de mensajes muy útiles a partir del  $r_{pb}$  de cada ítem individual e inclusive para cada una de las opciones de cada pregunta, con el fin de que el autor/profesor cuente con información para gestionar las preguntas y así poder mejorar aspectos de redacción o inclusive eliminación de opciones. En la Figura No. 7 se muestra cómo se visualiza ese tipo de mensajes:

FIGURA NO. 7: EJEMPLO DE MENSAJES SOBRE INDICES DE CORRELACION BISERIAL PUNTUAL POR PREGUNTA



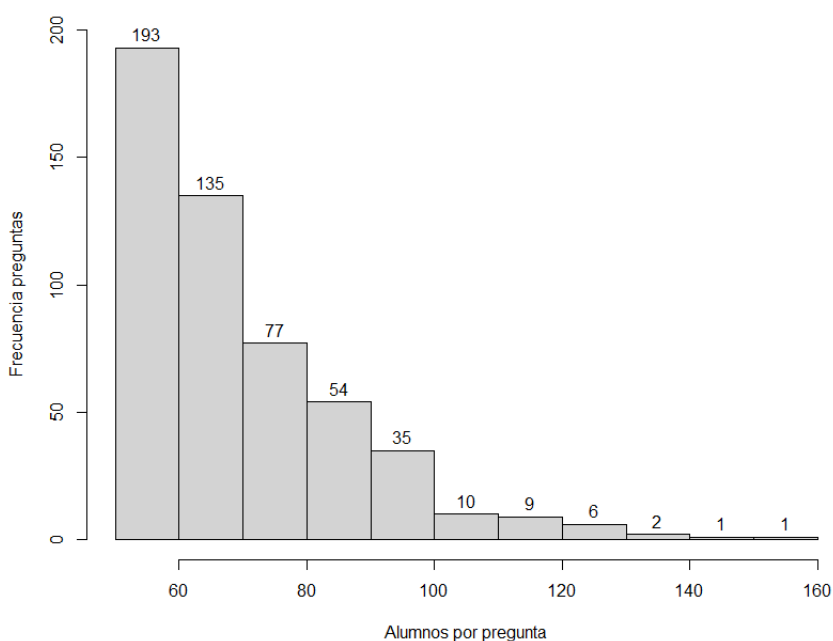
Fuente: "The Examiner"

#### 4.4 Generación y análisis de exámenes simulados con TCT

A partir de los resultados del análisis de los 50 exámenes reales con mayor cantidad de alumnos evaluados, se procedió a aumentar el estudio, realizando una comprobación por medio de la generación de 18 exámenes simulados. El procedimiento para la generación de esas pruebas simuladas fue la siguiente:

- Utilizando la base de datos, se identificaron las preguntas con mayor cantidad de respuestas, tomando como referencia un mínimo de 50 estudiantes por pregunta. Esta selección arrojó un total de 523 preguntas agrupadas, como se muestra en el Gráfico No. 6:

**GRÁFICO NO. 6: CANTIDAD DE ALUMNOS POR PREGUNTA PARA EXÁMENES SIMULADOS**



Fuente: "The Examiner"

La distribución de esas 523 preguntas con 50 o más respuestas, se agrupan por materia como se muestra en el Cuadro No. 6 :

**CUADRO NO. 6: DISTRIBUCIÓN POR MATERIA DE PREGUNTAS**

**CON 50 O MÁS RESPUESTAS**

<b>Código materia</b>	<b>Cantidad de preguntas</b>
COMP	156
FOC	117
ININ	97
AA	49
IO	37
BMC	34
PCB	18
CG	15
<b>Total</b>	<b>523</b>

Fuente: "The Examiner"

- Partiendo del enfoque por cada materia como premisa de mismo contenido evaluado, se creó un procedimiento para extraer de la base de datos preguntas, estudiantes y sus respuestas, convirtiéndolas en matrices binarias para poder realizar los cálculos necesarios (el procedimiento se puede observar en el anexo No.5)
- Se configuraron 3 exámenes por materia, escogiendo conjuntos de preguntas contestadas por un mismo grupo de estudiantes que hubieran cursado la materia en cualquier momento del tiempo.
- Se realizó el análisis para los exámenes simulados (en el anexo No.6 se presenta el código del procedimiento desarrollado)

Los resultados de la simulación se resumen en el Cuadro No. 7 :

**CUADRO NO. 7: RESULTADOS EXÁMENES SIMULADOS**

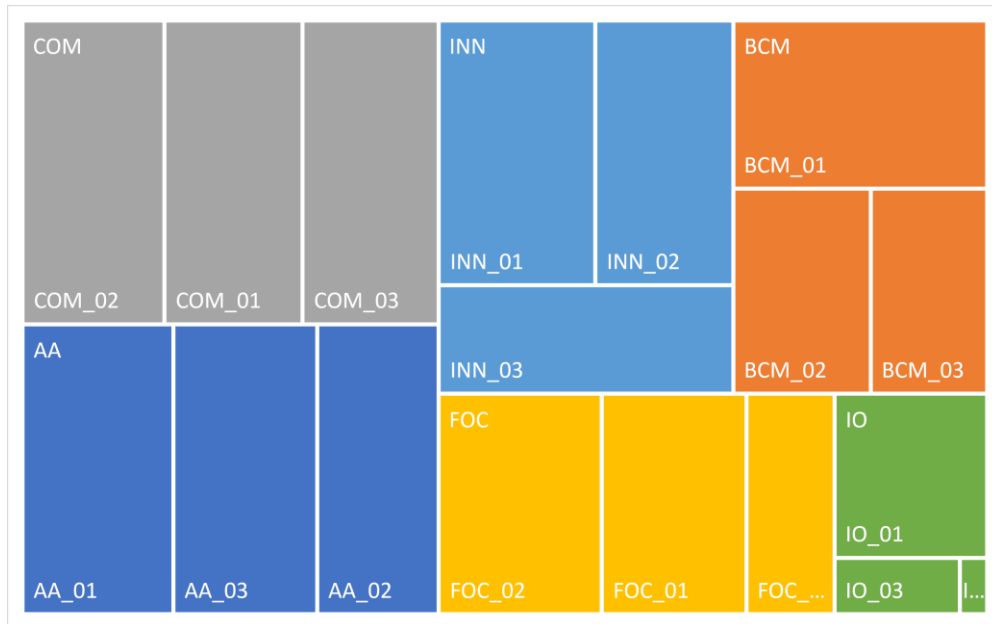
Código examen simulado	Cantidad alumnos	Cantidad de preguntas	Media predicha por el software	Límite inferior intervalo de confianza media simulada (95%)	media simulada	Límite superior intervalo de confianza media simulada (95%)	Coefficiente Alpha examen simulado
AA_01	25	22	52,14	41,63	49,45	57,28	0,80
AA_02	18	12	59,64	50,79	60,19	69,58	0,65
AA_03	18	12	64,76	58,35	68,06	77,77	0,77
BCM_01	17	15	59,19	51,23	61,57	71,91	0,78
BCM_02	9	11	59,45	50,13	60,61	71,08	0,52
BCM_03	16	7	73,10	66,34	75,00	83,66	0,44
COM_01	9	26	64,32	59,65	69,66	79,67	0,77
COM_02	21	36	63,44	63,39	69,58	75,76	0,79
COM_03	19	22	61,48	56,96	64,83	72,70	0,76
FOC_01	21	22	59,45	53,06	58,66	64,25	0,58
FOC_02	15	26	67,24	53,78	61,03	68,27	0,66
FOC_03	15	18	65,22	53,65	60,00	66,35	0,36
INN_01	19	33	64,64	61,94	68,26	74,58	0,76
INN_02	16	29	64,96	59,34	65,52	71,70	0,67
INN_03	30	24	58,07	56,38	61,25	66,12	0,59
IO_01	26	10	54,44	51,83	58,85	65,86	0,46
IO_02	15	10	64,72	56,65	64,00	71,35	0,03
IO_03	10	13	56,44	40,19	48,46	56,73	0,13

Fuente: elaboración propia a partir de base de datos “The Examiner”

Con el cuadro No.7 se puede observar que, en el 100% de los casos de la media predicha por el software, se ubicó dentro del intervalo de confianza simulado

En cuanto al coeficiente alfa, tomando en consideración que la fórmula que aplica internamente el software presenta un error de cálculo, no se realizarán comparaciones con los valores predichos por el software. Sin embargo, como se puede revisar en la tabla No.8, las magnitudes capturadas para el coeficiente alfa no sean del todo deseables, sí pueden brindar información útil a nivel de conjunto. Por ejemplo, utilizando un gráfico de jerarquías se puede visualizar el comportamiento por materia, como se muestra en el Gráfico No. 7 :

**GRÁFICO NO. 7: GRÁFICO DE JERARQUÍA PARA COEFICIENTES ALFA 18 EXÁMENES SIMULADOS**



Fuente: elaboración propia a partir de base de datos "The Examiner"

Tomando en consideración el Gráfico No. 7, se pueden generar observaciones interesantes por materia, como por ejemplo que la materia COM (Compiladores e intérpretes) y la materia AA (Análisis de algoritmos) presentan los mejores resultados de coeficientes alfa a nivel global, mientras que la materia IO (Investigación de operaciones) es la que presenta peores desempeños a nivel de fiabilidad de las pruebas simuladas.

## 4.5 Análisis de exámenes reales y simulados a partir de un enfoque de la TRI

### 4.5.1 Bondad de ajuste del modelo de Rasch en exámenes reales y simulados

De igual forma que para el análisis con un enfoque de TCT, se utilizaron los mismos 50 exámenes reales con mayor participación de estudiantes, con el fin de comprobar si el modelo de Rasch se adecuaba a los datos por medio de una prueba de bondad de ajuste<sup>6</sup>. Los resultados muestran que el modelo ajustó en 33 de los 50 exámenes (que equivale al 66%). Hubo 8 casos en los que el modelo no se ajustó presentando valores de  $p$  de 0,04 (que equivale al 16%) y 9 exámenes en los que el modelo no se ajustó presentando valores de  $p$  menores a 0,02 (que equivale al 18%).

Para los 18 exámenes simulados, también se utilizó una prueba de bondad de ajuste para evaluar el modelo de Rasch y se obtuvo que el modelo se ajustó en 16 de los 18 exámenes simulados (que equivale al 89%) y en los otros 2 no se logró ajustar el modelo (que equivale al 11%)

Si se toman en consideración ambos grupos de exámenes, tanto los reales como los simulados, se puede ver que el modelo se ajustó en un 72% de los casos.

### 4.5.2 Ajuste de ítems y personas

#### 4.5.2.1 Ajuste de ítems y personas en exámenes reales

Una vez realizadas las pruebas de bondad de ajuste, se utilizó el estadístico *Infit* para valorar el ajuste de los ítems y estudiantes de los exámenes reales y simulados. Partiendo del criterio de (Azpilicueta, Cupani, Ghío, Morán, & Garrido, 2019) sobre valores *Infit* en un rango de 0,5 a 1,5, para el caso de los exámenes reales, se pudo comprobar que 32 de los 50 (que equivale al 64%), se ajustaron tanto a nivel de bondad del ajuste general, sujetos e ítems. Hubo 9 exámenes, que generaron valores de ajuste aceptables tanto para ítems como para estudiantes, sin embargo, a nivel de bondad de ajuste del modelo no alcanzaron el valor de  $p$  esperado. Estos exámenes se sometieron a revisión de ajuste, eliminándoles ítems y/o sujetos

---

<sup>6</sup> Se utilizó para este fin el paquete eRm y su función `GoF.rasch()`, en el anexo No.6 se puede observar el código desarrollado para este fin.



que no ajustan al modelo.

En el Cuadro No. 8 se muestran los restantes 9 exámenes reales que se sometieron a revisión con el fin de eliminar ítems y/o sujetos que no presentaron buen ajuste y volver a valorar el nivel de ajuste.

**CUADRO NO. 8: EXÁMENES A REVISAR POR AJUSTE DE ÍTEMS Y/O ESTUDIANTES**

Examen	Valor de p, de ajuste	Item		Persona	
		INFIT mínimo	INFIT máximo	INFIT mínimo	INFIT máximo
ex00008	0,02	0,6	1,5	0,7	1,3
ex00050	0,04	0,9	1,1	0,7	1,6
ex00083	0,02	0,6	1,4	0,6	1,6
ex00098	0,02	0,8	1,3	0,7	1,7
ex00103	0,04	0,7	1,3	0,5	1,5
ex00105	0,02	0,6	1,4	0,6	1,5
ex00110	0,04	0,8	1,1	0,8	1,5
ex00130	0,16	0,5	1,4	0,5	1,9
ex00147	0,02	0,5	1,6	0,6	1,5

Fuente: elaboración propia a partir de datos The Examiner

Se realizó una depuración de ítems y/o estudiantes que no cumplieran con el valor de referencia del estadístico *Infit*, con los resultados del Cuadro No. 9:

**CUADRO NO. 9: RESULTADO DE ELIMINACIÓN POR INFIT NO ADECUADO**

Examen	Valor de p, bondad de	Item		Persona	
		INFIT mínimo	INFIT máximo	INFIT mínimo	INFIT máximo
ex00008	0,02	0,58	1,38	0,73	1,30
ex00050	0,04	0,92	1,08	0,69	1,46
ex00083	0,02	0,64	1,39	0,64	1,46
ex00098	0,02	0,75	1,34	0,67	1,29
ex00103	0,04	0,74	1,28	0,53	1,48
ex00105	0,02	0,63	1,38	0,56	1,48
ex00110	0,04	0,81	1,13	0,82	1,48
ex00130	0,48	0,55	1,34	0,54	1,27
ex00147	0,44	0,53	1,50	0,55	1,46

Fuente: elaboración propia a partir de datos The Examiner

En el examen ex00008, se eliminó un ítem, pero no impactó el resultado de la bondad de ajuste. En los casos que se eliminaron personas que sobrepasaban el valor superior del estadístico de *Infit* tampoco se obtuvo mejores resultados en la bondad de ajuste del modelo. Sin embargo, en el caso de los exámenes ex00130 y ex00147, que presentaron valores de *Infit* inferiores y superiores en ítems y estudiantes a la misma vez, al eliminar esos casos y volver a generar el procedimiento de bondad de ajuste, sí se registró una mejoría en el valor de  $p$ , convirtiéndose en exámenes que se ajustan al modelo de Rasch.

#### 4.5.2.2 Ajuste de ítems y personas en exámenes simulados

Se realizó el mismo ejercicio para los exámenes simulados. Resultando que, de los 18 ejemplares, 8 de los mismos (o lo que equivale al 44%) presentaron valores adecuados en cuanto al resultado de la prueba de bondad de ajuste, el ajuste de los ítems y el ajuste de personas.

Hubo 8 casos de exámenes que presentaron valores adecuados en la bondad de ajuste, pero presentaron valores menores o mayores a los esperados a nivel del estadístico de *Infit*. Lo mismo ocurrió en los casos de los exámenes con valores que no apoyan el resultado de la prueba de bondad de ajuste. Además, los dos casos que no generaron valores de  $p$  adecuado en la prueba de bondad de ajuste también presentaron valores de *Infit* por encima o por debajo del indicador esperado. Esos 10 casos se sometieron a revisión y son los que se presentan en la siguiente tabla:

**CUADRO NO. 10: EXÁMENES SIMULADOS PARA REVISIÓN DE VALORES *INFIT***

Examen	Valor de $p$ , bondad de ajuste	Item		Persona	
		INFIT mínimo	INFIT máximo	INFIT mínimo	INFIT máximo
AA_02.csv	0,44	0,66	1,08	0,46	1,26
AA_03.csv	0,32	0,54	1,25	0,22	1,66
BCM_01.csv	0,04	0,62	1,43	0,25	1,69
BCM_02.csv	0,44	0,72	1,22	0,4	1,27
BCM_03.csv	0,32	0,76	1,05	0,31	1,67
COM_01.csv	0,44	0,66	1,61	0,24	1,35
COM_03.csv	0,06	0,69	1,24	0,46	1,37
FOC_01.csv	0,04	0,79	1,19	0,61	1,73
IO_01.csv	0,24	0,81	1,13	0,42	1,62
IO_02.csv	0,32	0,96	1	0,68	1,54

Fuente: elaboración propia a partir de datos The Examiner

Al realizar la eliminación de ítems o sujetos con valores fuera del rango de 0,5 a 1,5, se generan los resultados en el Cuadro No. 11:

**CUADRO NO. 11: RESULTADO DE ELIMINAR ÍTEMS/SUJETOS CON VALORES DE ESTADÍSTICO  
INFIT FUERA DE VALOR ESPERADO.**

Examen	Valor de $p$ , bondad de	Item		Persona	
		INFIT mínimo	INFIT máximo	INFIT mínimo	INFIT máximo
AA_02.csv	0,48	0,68	1,07	0,51	1,26
AA_03.csv	0,14	0,53	1,34	0,50	1,60
BCM_01.csv	0,14	0,63	1,48	0,38	1,65
BCM_02.csv	0,44	0,72	1,22	0,40	1,27
BCM_03.csv	0,32	0,76	1,05	0,31	1,67
COM_01.csv	0,44	0,72	1,60	0,44	1,26
COM_03.csv	0,06	0,69	1,24	0,46	1,37
FOC_01.csv	0,06	0,69	1,23	0,50	1,37
IO_01.csv	0,32	0,83	1,14	0,53	1,48
IO_02.csv	0,12	0,96	1,00	0,68	1,50

Fuente: elaboración propia a partir de datos The Examiner

Con los resultados tabulados anteriormente, se puede ver que se logró que el 100% de los exámenes se ajustaran al modelo, al eliminar valores altos o bajos a nivel de estadístico *Infit*. En los exámenes que ya presentaban valor de  $p$  no significativo en la prueba de bondad de ajuste, aunque sufrieron alguna modificación en ese valor, mantuvieron esa condición de apoyar la hipótesis de ajuste al modelo de Rasch.

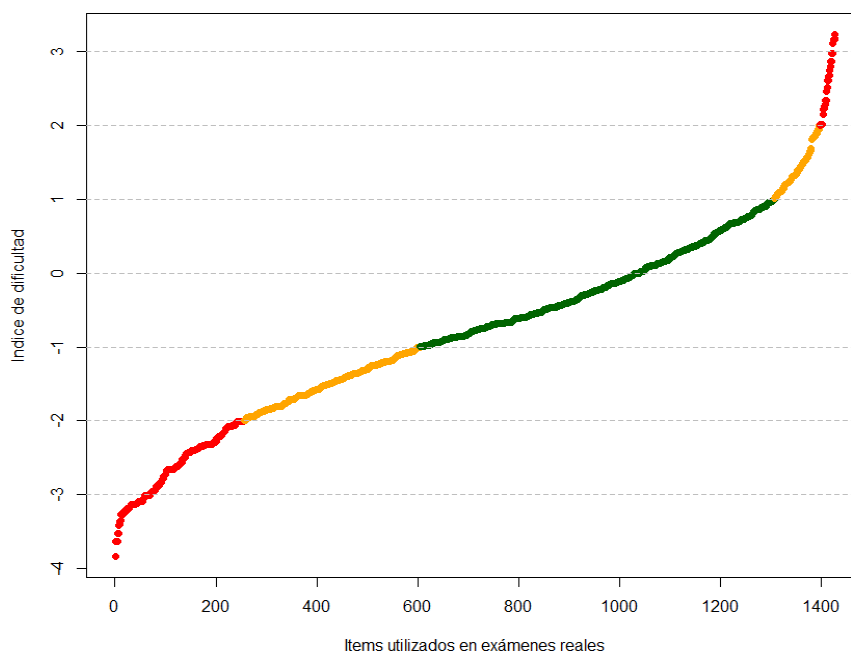
### 4.5.3 Construcción de una base de datos de ítems con índices de dificultad

Los resultados del apartado anterior reflejan la realidad y limitación de las pruebas en el aula, que pueden no generarse sobre la aplicación de gran cantidad de participantes, sin embargo, el ejercicio con respecto a la TRI, si se cuenta con el apoyo computacional adecuado, puede comenzar a generar información útil como lo es el ajuste en el cálculo de indicadores de dificultad de los ítems.

Entonces, el siguiente paso consistió precisamente en capturar en una base de datos el resultado de los indicadores de dificultad de los ítems utilizados en los exámenes reales, así como los índices de dificultad de los ítems utilizados en los exámenes simulados, con el fin de identificar el grado de correlación de esos índices.

Haciendo una revisión de la base de datos en conjunto, en el Gráfico No. 8: Ítems utilizados en exámenes reales vs. índice de dificultad, se puede observar una representación de los 1428 ítems que se utilizaron en los 50 exámenes con mayor participación, respecto al ajuste en sus índices de dificultad.

GRÁFICO NO. 8: ÍTEMS UTILIZADOS EN EXÁMENES REALES VS. ÍNDICE DE DIFICULTAD



Fuente: elaboración propia a partir de datos The Examiner

Haciendo un resumen de los elementos que conforman la figura No.15, tenemos como resultado el Cuadro No. 12:

**CUADRO NO. 12: ÍNDICES DE DIFICULTAD POR RANGOS**

<b>Rango dificultad</b>	<b>Frecuencia ítems</b>	<b>Porcentaje</b>
$d > 2$	31	2%
$1 < d < 2$	90	6%
$-1 < d < 1$	703	49%
$-2 < d < -1$	349	24%
$d < -2$	255	18%
<b>Total</b>	<b>1428</b>	<b>100%</b>

Entonces, se puede ver que la base de datos de índices de dificultad para los ítems utilizados en las pruebas reales (incluyendo todas las materias), el 49% se ubican en el rango de -1 a 1 en la escala *logit* y que se refieren a ítems con una buena capacidad de discriminación. Un 42% de los ítems presentan índices menores a -1, lo que significa un grupo de ítems de menor capacidad de discriminación, ya que su dificultad es menor y finalmente un 8% de ítems con valores superiores a 1 en escala *logit* que incluirían un grupo de ítems que tampoco tienen mejor discriminación, debido a sus mayores índices de dificultad.

#### 4.5.4 Análisis de correlación entre índices de dificultad de la base de datos creada y los índices de dificultad de los exámenes simulados

En el Cuadro No. 13 se pueden ver observar las medidas calculadas para los indicadores de los exámenes simulados, como lo es el coeficiente alfa, los índices de dificultad mínimos y máximos para el grupo de ítems contemplados en cada prueba y se incluye el cálculo del coeficiente de correlación entre los ítems de la base de datos generada como histórica (a partir de los ítems que conforman los 50 exámenes reales con mayor cantidad de estudiantes) y los índices de dificultad capturados con el ajuste de los exámenes simulados:

**CUADRO NO. 13: RESULTADOS DE EXÁMENES SIMULADOS**

Código examen simulado	Cantidad alumnos	Cantidad de preguntas	Media predicha por el software	Límite inferior int.conf. media simulada (95%)	Límite superior int.conf. media simulada (95%)	Coeficiente Alpha examen simulado	Dificultad Mínima ítems (histórico)	Dificultad Máxima ítems (histórico)	Dificultad Mínima ítems (prueba simulada)	Dificultad Máxima ítems (prueba simulada)	Coeficiente de correlación entre índices de dificultad
AA_01	25	22	52,14	41,63	57,28	0,80	-2,20	2,28	-1,67	2,35	0,89
AA_02	18	12	59,64	50,79	69,58	0,65	-2,36	0,73	-1,03	1,02	0,82
AA_03	18	12	64,76	58,35	77,77	0,77	-2,20	2,28	-2,11	2,69	0,91
BCM_01	17	15	59,19	51,23	71,91	0,78	-3,16	1,63	-3,26	3,21	0,89
BCM_02	9	11	59,45	50,13	71,08	0,52	-2,96	1,40	-2,35	1,44	0,83
BCM_03	16	7	73,10	66,34	83,66	0,44	-1,09	0,28	-1,27	0,28	0,62
COM_01	9	26	64,32	59,65	79,67	0,77	-3,09	2,34	-2,15	2,15	0,82
COM_02	21	36	63,44	63,39	75,76	0,79	-3,26	1,94	-3,28	1,05	0,92
COM_03	19	22	61,48	56,96	72,70	0,76	-3,05	1,94	-3,26	1,56	0,85
FOC_01	21	22	59,45	53,06	64,25	0,58	-3,14	2,01	-3,18	1,93	0,96
FOC_02	15	26	67,24	53,78	68,27	0,66	-2,34	1,36	-1,10	1,10	0,54
FOC_03	15	18	65,22	53,65	66,35	0,36	-2,24	2,57	-2,98	1,98	0,80
INN_01	19	33	64,64	61,94	74,58	0,76	-3,07	0,91	-3,17	0,88	0,91
INN_02	16	29	64,96	59,34	71,70	0,67	-2,85	2,25	-2,91	2,92	0,91
INN_03	30	24	58,07	56,38	66,12	0,59	-2,94	1,04	-3,52	-0,29	0,86
IO_01	26	10	54,44	51,83	65,86	0,46	-2,40	0,95	-2,58	2,10	0,94
IO_02	15	10	64,72	56,65	71,35	0,03	-3,00	0,54	-2,65	-0,41	0,86
IO_03	10	13	56,44	40,19	56,73	0,13	-1,70	0,89	-1,40	2,22	0,83

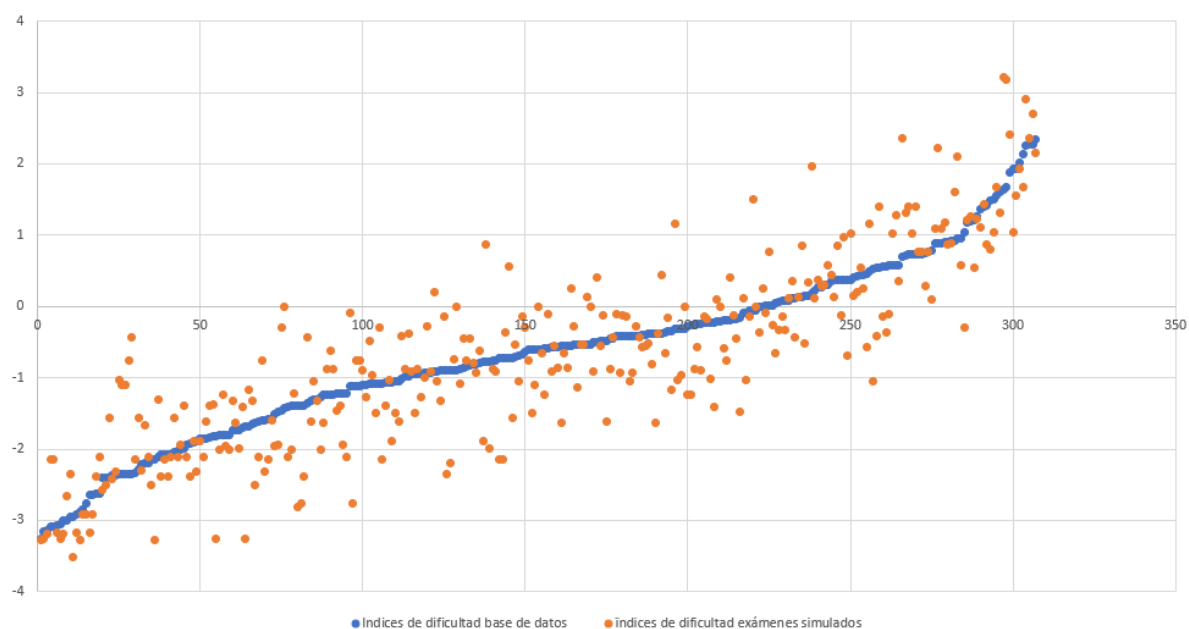
Fuente: elaboración propia a partir de datos The Examiner

Se puede observar que el valor predicho de la media del examen está incluido en el intervalo de confianza para el 100% de los casos, lo que hace ver que este indicador es información útil para la construcción de pruebas con el software.

En cuanto a los coeficientes alfa, ya se comentó anteriormente, que a pesar de que no en todos los casos la magnitud calculada presenta valores deseados (superiores a 0,7), en sí misma, la información generada permite observar comportamiento a nivel de materias y probablemente ajustando el cálculo a nivel de código fuente, permitirá información para depurar ítems que hasta este momento se tenían registrados con valores superiores a su valor real.

Generando un gráfico de dispersión de los índices de dificultad ajustados para el grupo de los ítems utilizados en la construcción de los 18 exámenes simulados y graficarlos respecto a los valores de los índices correspondientes de la base de datos que se construyó para los índices de dificultad de los ítems de los 50 exámenes con mayor cantidad de alumnos (ordenados de menor a mayor dificultad), se puede generar una visualización de cuál es su comportamiento, como se muestra en el Gráfico No. 9:

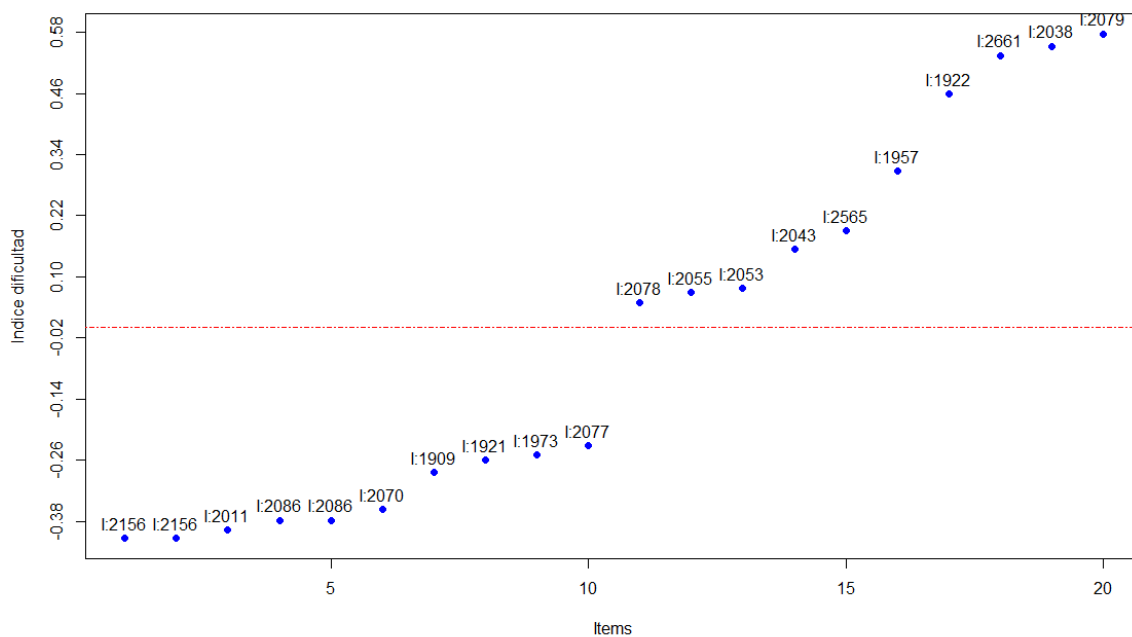
**GRÁFICO NO. 9: ÍNDICES DE DIFICULTAD "ÍTEMES HISTÓRICOS" VS.  
ÍNDICES DE DIFICULTAD ÍTEMES EXÁMENES SIMULADOS**



El índice de correlación general para estos 348 ítems es de 0,87 lo cual evidencia que los ajustes de los índices de dificultad permitirían en una nueva funcionalidad del software, solicitar ítems en rangos de dificultad específica y con ello apoyar la construcción de exámenes.

Este tipo de gráfico puede ser una herramienta interesante a la hora de confeccionar exámenes con el software, pues con los índices de dificultad registrados en la base de datos, se pueden “solicitar” ítems en determinado rango. Por ejemplo, tomando en consideración los ítems de la materia de Compiladores e Intérpretes, a continuación, se presenta un filtro de 20 ítems (10 ítems con rango de dificultad de -0,41 a 0 y otros 10 ítems con rango de dificultad de 0 a 0,58). Esto significaría que se contaría con una prueba “balanceada” con ítems en el umbral de habilidad. En el Gráfico No. 10 , se muestra cómo se visualizaría la selección de los 20 ítems descrita:

**GRÁFICO NO. 10: SELECCIÓN DE 20 ÍTEMS POR ÍNDICE DE DIFICULTAD**





## CAPÍTULO V: CONCLUSIONES Y DISCUSION

Utilizando el software “The Examiner” y su base de datos, el presente trabajo se enfocó en el análisis de la información que ofrece la herramienta para la construcción y parametrización de exámenes de aula, en principio desde el punto de vista de la TCT y luego generando información con un enfoque de TRI, con el fin de validar si el modelo de Rasch se ajusta tomando como base los indicadores históricos consignados en la base de datos y produciendo algunas simulaciones de exámenes para apoyar los resultados.

Inicialmente se tuvo que realizar la instalación de los componentes del software en el ambiente requerido para su correcto funcionamiento. En este sentido, desde el punto de vista del usuario medio, sí se requiere cierto dominio de herramientas, por lo que una guía de instalación en versiones posteriores será de mucha utilidad para facilitar esa tarea. Una vez lograda la instalación, se debe comentar que el software es intuitivo y fácil de utilizar. El orden de los diferentes módulos implica el orden del flujo de procesos para la creación de exámenes, lo que facilita la comprensión de tareas por parte del usuario. Con este trabajo, se facilita una guía ilustrada, que no pretende ser un manual de usuario, pero que si el lector la utiliza, confirmará que el uso de la herramienta es simple.

La revisión que se realizó respecto al enfoque de la TCT incorporado en el software, permitió confirmar la forma en la que los algoritmos generan preexámenes y cálculos de diferentes indicadores, como la media esperada a partir de los índices de dificultad calculados con las proporciones de aciertos históricas de los ítems y coeficientes alfa.

En primer lugar, se comprobó que la media estimada para un examen es un valor útil para la toma de decisión a la hora de su confección; en el caso de los 50 exámenes con mayor participación de estudiantes, en el 84% de los casos, la media estimada por el software está contenida en el intervalo del 95% de la media para sus calificaciones reales y en el caso de los 18 exámenes simulados, este valor esperado quedó incluido en el intervalo de confianza en el 100% de las ocasiones.

En el caso de los coeficientes alfa, el hallazgo a nivel del cálculo interno del software, mostró que las magnitudes de esos coeficientes está siendo afectada por la utilización de la varianza de las notas de los alumnos para su cálculo, siendo lo correcto utilizar la varianza de la cantidad de respuesta correctas por alumno. A pesar de esta situación de cálculo, corriendo los valores de coeficientes alfa para los 50 exámenes con mayor participación, se determinan valores aceptables en el 72% de los casos, lo cual es un aspecto bastante favorable. En el caso de los exámenes simulados, el 39% de los mismos generan valores adecuados y aunque esto no es lo idealmente esperado, se propone la utilización de un gráfico de jerarquía, en el que se puede mostrar el comportamiento de los exámenes por materia, con resultados interesantes que muestran una consistencia por ejemplo de la materia de Compiladores e Intérpretes como la materia con mejores coeficientes de confiabilidad en las pruebas.

Posteriormente, se corrieron rutinas para evaluar el ajuste del modelo de Rasch tomando como base 50 exámenes con mayor participación de estudiantes y 18 exámenes simulados, resultando que en un 72% de los casos, se consiguió un buen ajuste del modelo. También se realizaron pruebas de ajuste para los ítems y los estudiantes. En el caso de los 50 exámenes con mayor participación de estudiantes, se logró ajuste en el 86% de los casos, mientras que, en los exámenes simulados el 100% ajustó adecuadamente.

Partiendo de los resultados del ajuste, se generó una base de datos con los índices de dificultad desde una perspectiva de la TRI para 1428 ítems, que presentaron un rango de -4 a 3 en la escala *logit*, pudiendo observar que el banco de preguntas tiende a ser “fácil”, con un 42% de ítems con índices menores a -2. Para los exámenes simulados, se ejecutó el índice de correlación entre la dificultad de los ítems consignados en la base de datos mencionada y los índices propios de los mismos ítems en estos exámenes simulados, arrojando un valor de 0.87, lo cual es un indicador favorable para pensar que la construcción de exámenes a partir de información consignada en la base de datos es útil.

En conclusión, se puede argumentar que el software en versiones posteriores, sí puede ser una herramienta útil para la construcción de exámenes que aporten elementos de ambos enfoques. La TCT, sigue dando aspectos interesantes como la media esperada que es un

concepto cotidiano y asimilado en el ámbito docente, así como el concepto de coeficiente de alfa o Cronbach. Los aportes que puede brindar la TRI, nutren el proceso de confección de pruebas a partir del ajuste al modelo de Rasch, con el análisis de los índices de dificultad independientes de la persona evaluada.

Para investigaciones futuras, conforme se vaya generando mayor cantidad de información de los ítems, será interesante valorar el ajuste de otros modelos, con el fin de analizar parámetros como discriminación o indicadores de respuestas al azar por parte del estudiante.

En cuanto a temas de uso del software, para lanzamientos posteriores es importante contemplar la posibilidad de desarrollar en otros sistemas operativos como *Windows* o *iOS*, tomando en consideración que los usuarios potenciales para este software pueden ser docentes de nivel universitario de cualquier área o docentes de otros niveles educativos como profesores de secundaria, que por lo general no utilizan comúnmente el sistema operativo *Linux*. En esta misma línea, la utilización de *LaTeX* para la elaboración de ítems, puede ser una limitación para el usuario, por lo es válido analizar la posibilidad de poder incluir texto enriquecido como fórmulas matemáticas, utilizando de base otros editores de texto comerciales.

## REFERENCIAS

- Domínguez-Rodríguez, Y. (2022). Instrumentos y tipos de evaluación. *Con-Ciencia Serrana*, 4(7), 37-39.
- Moreira-Mora, T. E., Alfaro-Rojas, L., Brizuela-Rodríguez, A., Chacón-Vega, C., Gómez-González, E., Jiménez-Alfaro, K., . . . Villarreal-Galera, M. P. (2022). *Estándares de calidad para pruebas estandarizadas de alto impacto en el contexto académico y profesional costarricense*. Instituto de Investigaciones Psicológicas.
- Torres-Rojas, F. (2008). *The Examiner: Automatic Generation of "Good" Exams*. Instituto Tecnológico de Costa Rica.
- Paek, I., & Cole, K. (2020). *Using R for Item Response Theory Model Applications*. Taylor & Francis.
- Martínez Arias, M. R., Hernández Lloreda, M. J., & Hernández Lloreda, M. V. (2014). *Psicometría*. Alianza Editorial.
- Price, L. (2017). *Psychometric Methods, Theory Into Practice*. The Guilford Press.
- Holmes Finch, W., French, B., & Immekus, J. (2014). *Applied Psychometrics Using SAS*. Information Age Publishing, INC.
- Attorresi, H. F., Lozzia, G. S., Abal, F. J., Galibert, M. S., & Aguerri, M. E. (2009). Teoría de Respuesta al Item: Conceptos básicos y aplicaciones para la medición de constructos psicológicos. *Revista Argentina de Clínica Psicológica*, 23, 179-188.
- Prieto, G., & Delgado, A. (2003). Análisis de un test mediante el modelo de Rasch. *Psicothema*, 15(1), 94-100.
- Wind, S., & Hua, C. (2022). *Rasch Measurement Theory Analysis in R*. Taylor & Francis Group. doi:10.1201/9781003174660
- Rojas, S., & Montero, E. (2021). *El modelo de Rasch: una herramienta esencial para la evaluación*. Consejo Nacional de Rectores, Cuarto Informe del Estado de la Educación.
- Sheu, C.-F., Chen, C.-T., & Wang, W.-C. (2005). Using SAS PROC NLMIXED to fit item response theory models. *ARTICLES FROM THE SCIP CONFERENCE*. Behavior Research Methods. doi:10.3758/bf03192688
- Meneses, J., Barrios, M., Bonillo, A., Coscolluela, A., Lozano, L., Turbany, J., & Valero, S. (2013). *Psicometría*. Editorial UOC.
- Azpilicueta, A. E., Cupani, M., Ghío, B., Morán, V. E., & Garrido, S. J. (2019). Adaptación mediante el modelo de Rasch de tres medidas para estimar la decisión e indecisión de carrera y la ansiedad decisiva. *PERSPECTIVAS EN PSICOLOGÍA*, 16(1), 26-37.

Muniz, J. (2010). LAS TEORÍAS DE LOS TESTS: TEORÍA CLÁSICA Y TEORÍA DE RESPUESTA A LOS ÍTEMS. *Papeles del Psicólogo*, 31(1), 57-66.

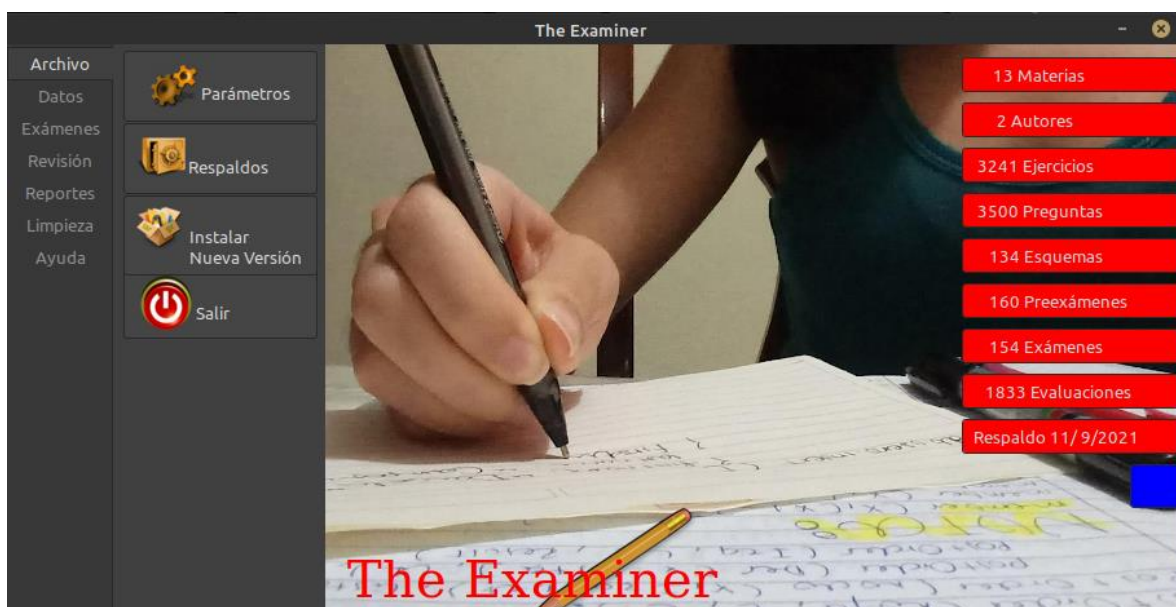
**ANEXOS**

## ANEXO No.1: GUÍA ILUSTRADA DEL SOFTWARE “THE EXAMINER”

En este anexo se listan las funcionalidades del software “The Examiner” en su versión 0.6. No se trata de un manual de usuario; solamente se harán descripciones breves de las diferentes funcionalidades, con el fin de contextualizar al lector en cuanto a las posibilidades y opciones que se pueden desarrollar con el software.

1. Archivo: al ejecutar el software se presenta la pantalla de inicio que se muestra en la Figura No. 8 que incluye opciones de administración del software (parámetros, respaldos, instalar nueva versión y salir), así como información resumen del contenido de la base de datos (cantidad de materias, cantidad de autores, cantidad de ejercicios, cantidad de preguntas, cantidad de esquemas, cantidad de preexámenes, cantidad de exámenes, cantidad de evaluaciones y fecha del último respaldo).

FIGURA NO. 8: PANTALLA DE INICIO

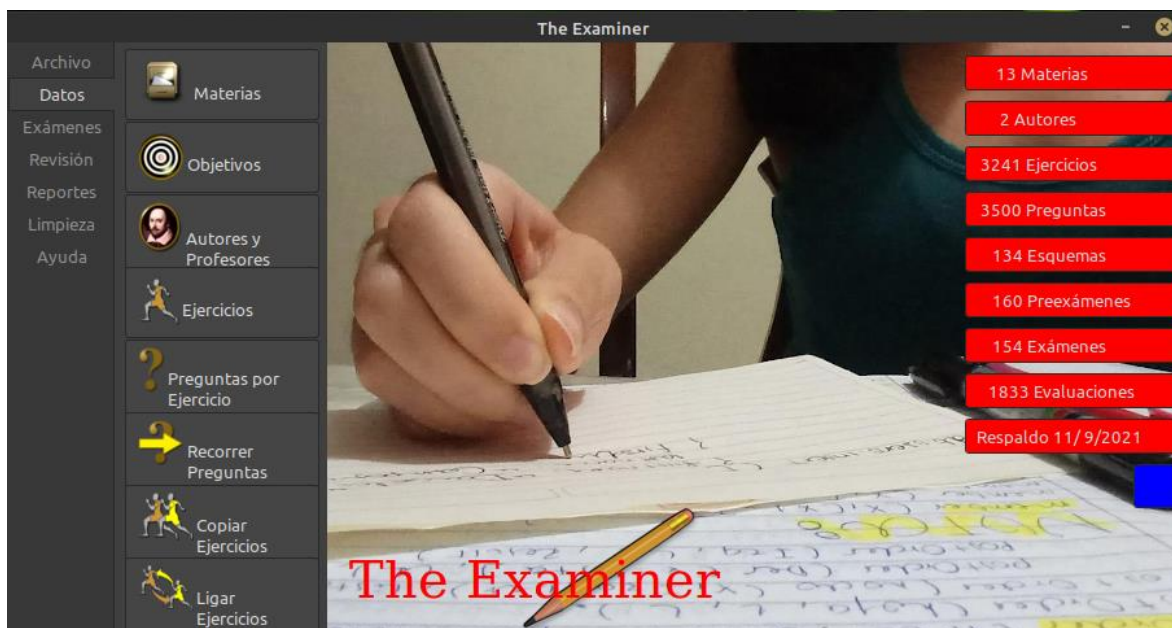


- a. Parámetros: se refiere a parámetros de ejecución del software, al escoger ese menú se desplegarán las siguientes opciones:
  - i. Interacción: permite algunas opciones para modificar la interfaz del software
  - ii. Programas: permite visualizar los programas que se están utilizando para desplegar documentos PDF, procesar código Latex a PDF, etc.
  - iii. Código Latex: permite ingresar paquetes de código Latex
  - iv. Directorios: permite configurar los directorios en los cuales se almacenan los exámenes, reportes, fuentes de Latex, fuentes de gnuplot, respaldos y figuras.
  - v. Datos: permite ingresar algunos datos globales que se utilizan como encabezado de exámenes, como nombre de la institución, escuela de la institución, programa, instrucciones y juramente de comportamiento durante pruebas,
  - vi. Presentaciones Beamer: permite parametrizar la salida de presentaciones que se construyen con Beamer.
- b. Respaldos: con esta opción se puede respaldar el software y la base de datos en una ruta específica.
- c. Instalar nueva versión: en el caso de existir nuevas versiones del software, establece las rutas para poder ubicarlas.



2. Datos: en la Figura No. 9; **Error! No se encuentra el origen de la referencia.** se muestra la pantalla de datos:

FIGURA NO. 9: PANTALLA DE DATOS



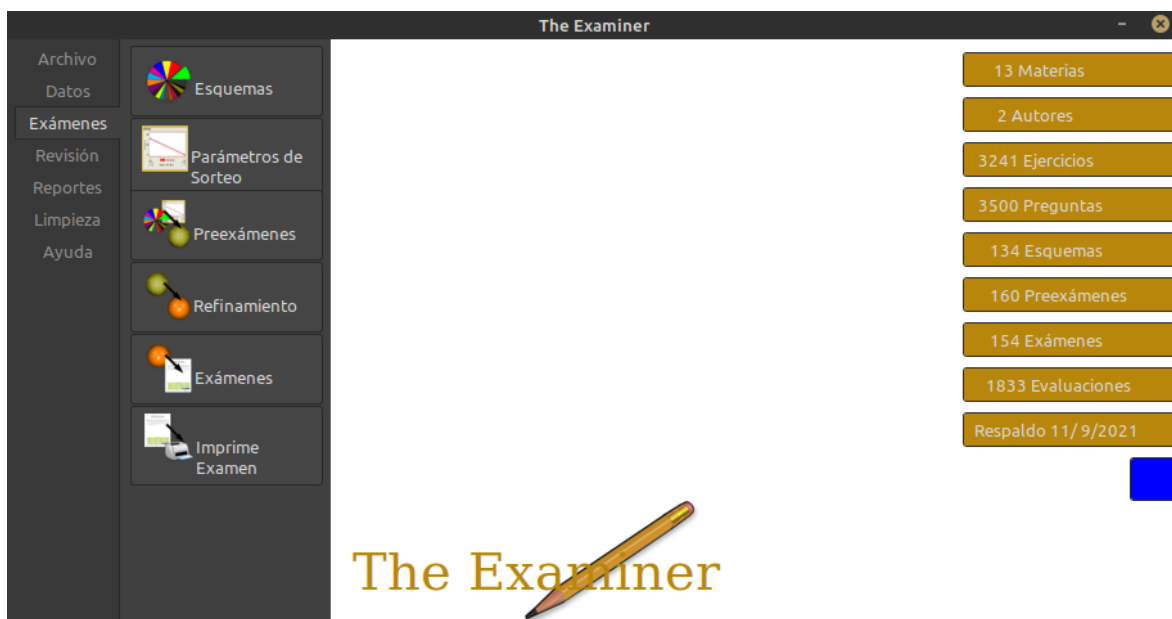
En esta pantalla se pueden realizar las siguientes acciones:

- a. Incluir materias: en esta parte del software, es la sección en la que el usuario puede incluir materias, con sus temas y subtemas. La intención es que posteriormente, los ejercicios y las preguntas se puedan incluir según esa clasificación (Materia – Tema – Subtema).
- b. Objetivos: en esta sección, se pueden incluir objetivos generales y específicos de aprendizaje para cada una de las materias.
- c. Profesores y autores de ejercicios: se pueden incluir identificaciones y nombres de autores y profesores de ejercicios, así como declarar su afiliación a una institución determinada.
- d. Ejercicios: en esta pantalla, el usuario puede visualizar y crear los ejercicios ingresados a la base de datos. Cada ejercicio nuevo se ingresa asociándolo a la materia, tema y subtema a la que aplica.

- e. Preguntas por ejercicio: en esta sección, el usuario puede configurar las preguntas que aparecen en el examen, así como las opciones para respuesta única, indicando cuál es la correcta.
- f. Recorrer preguntas: la pantalla de recorrer preguntas, permite visualizar la información registrada para cada pregunta a nivel histórico, desde su fecha de creación, hasta la fecha de último uso en algún examen. Además, en cuántos exámenes ha estado presente, información de la media, varianza, alfa de Cronbach (general y calculado en caso de que no hubiese estado presente en el examen), cantidad de estudiantes que la han respondido, cálculo de su índice de dificultad ( $p$ ) y  $r_{pb}$  calculado.
- g. Copiar ejercicios: permite copiar ejercicios entre diferentes materias.
- h. Ligar ejercicios: permite ligar ejercicios de diferentes materias.

3. Exámenes: en la Figura No. 10, se muestran las opciones del menú de exámenes:

FIGURA NO. 10: PANTALLA EXÁMENES



- a. Esquemas: un esquema consiste en la configuración general que tendrá un

preexamen en término de qué materia es la evaluada, qué temas y subtemas se contemplan y la cantidad de preguntas que se incluirán en cada uno ellos, así como sus porcentajes en la evaluación. En la Figura No. 11; **Error! No se encuentra el origen de la referencia.**, se puede observar un ejemplo de esquema de examen, para la materia Investigación de Operaciones, en la que se aplicarían 15 preguntas distribuidas en varios de los temas/subtemas específicos.

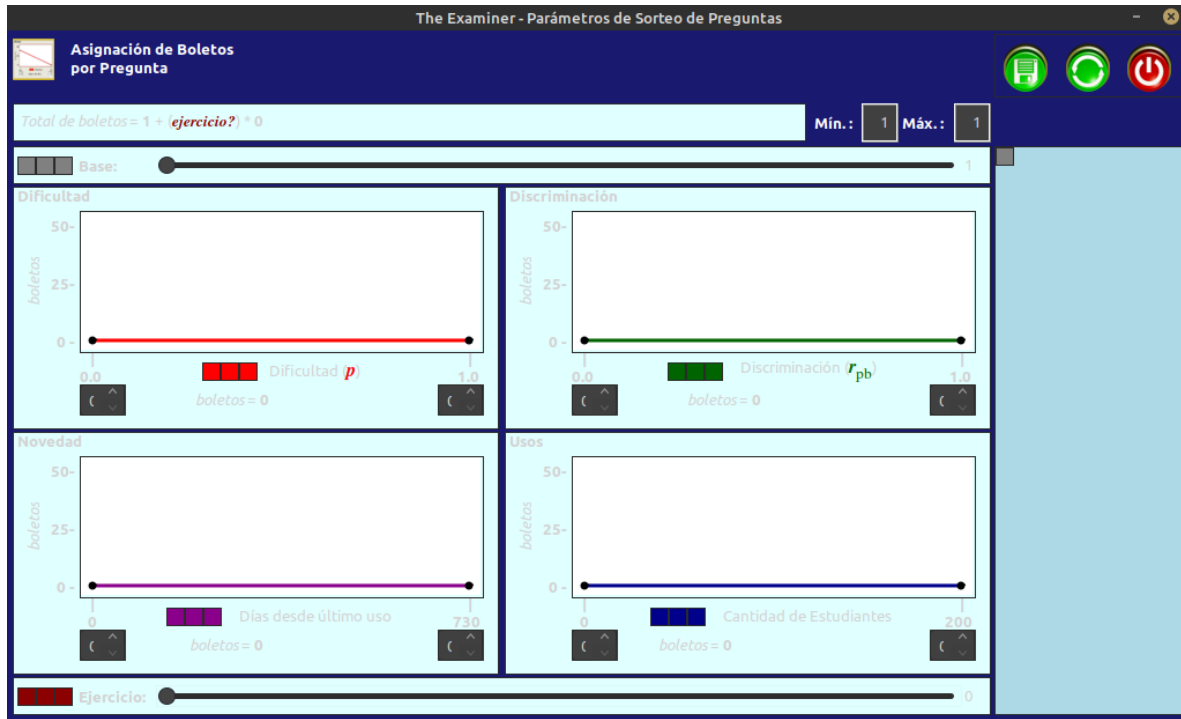
FIGURA NO. 11: EJEMPLO DE ESQUEMA DE EXAMEN

The screenshot shows the 'The Examiner - Definición de Esquemas' interface. At the top, there are navigation buttons for 'Básico', 'Porcentajes', and 'Cantidades'. The 'Cantidades' button is selected. Below the navigation, there are fields for 'Código: 134', 'Materia: IO Investigación de Operaciones', and 'Descripción: Esquema Examen Final 2021 - 1'. The main area is a table with columns for 'Tema', 'Subtema', 'Preguntas', and 'Total:'. The 'Total' column shows a value of 15. The table lists the following topics and subtopics with their respective question counts and percentages:

Tema	Subtema	Preguntas	Total:
ALGOR	Análisis de Algoritmos	1	6.67%
INTRO	Introducción	1	6.67%
LP	Programación Lineal	3	20.00%
PAPER	Artículos Adicionales	1	6.67%
	KAHNE Kahneman - Fast and Slow		
PD	Programación Dinámica	3	20.00%
PROYE	Control de Proyectos	3	20.00%
	CPM Critical Path Method		
TRANS	Problema de Transporte	3	20.00%

- b. Parámetros de sorteo: las preguntas que se incluirían en un examen dependerán del algoritmo de escogencia, que se basa en “boletos” que el usuario puede asignar, con base en las siguientes variables: dificultad de la pregunta (inicialmente basada en el criterio de su creador y posteriormente según el registro histórico de las respuestas), índice de discriminación, días desde el último uso de la pregunta y cantidad de estudiantes que la han respondido. En la se puede observar la pantalla de asignación de boletos por pregunta inicial. En ella se puede aumentar la cantidad de boletos a discreción del usuario (Figura No. 12)

FIGURA NO. 12: PANTALLA DE ASIGNACIÓN DE BOLETOS POR PREGUNTA



- c. Preexámenes: la creación de preexámenes se basa en la escogencia por medio del algoritmo de elección según boletos descrito anteriormente. El software permite la configuración de gran cantidad de exámenes con base en la materia, sus temas, sus subtemas y las cantidades de preguntas definidas en el esquema. Con esa información, el usuario puede visualizar un panel de preexámenes, en los que se puede observar su nivel de dificultad esperado, su nivel de discriminación esperado, su novedad, el alfa de Cronbach esperado. En la Figura No. 13 se puede ver un ejemplo de la pantalla de creación de preexámenes en la que el usuario puede ejecutar cuántas veces desee, para configurar un nivel de examen esperado en el que esté de acuerdo.

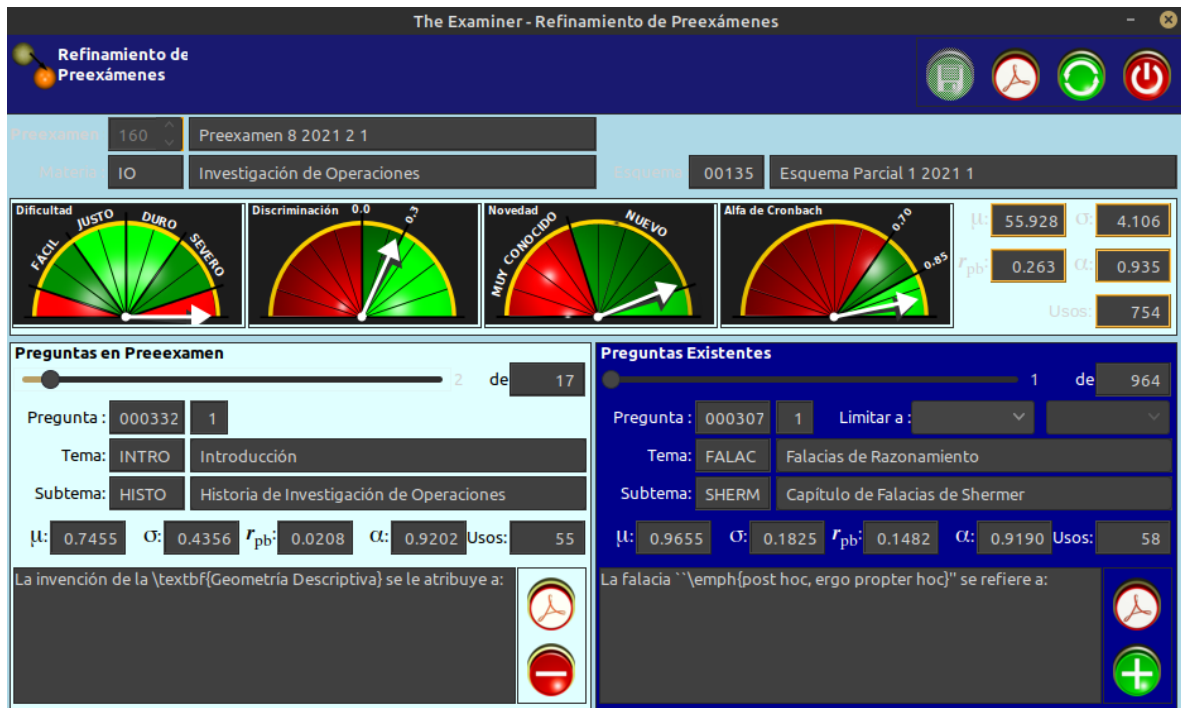
FIGURA NO. 13: PANTALLA DE CREACIÓN DE PREEXÁMENES



En la Figura No. 13, se puede observar en la primera posición del panel, que el preexamen generado presenta una dificultad categorizada como justa según el índice de dificultad de las preguntas incluidas y se pueden observar valores esperados como el promedio de nota (68.27), la desviación estándar esperada (10,374), el  $r_{pb}$  esperado (0,237), el Alfa de Cronbach esperado (0.907). En caso de que esta “configuración” de preexamen fuese valorada como adecuada por parte del usuario, el preexamen se puede “grabar” para la siguiente fase de refinamiento (en caso de querer ejecutar algún ajuste al mismo) o de convertirlo a examen si es el caso.

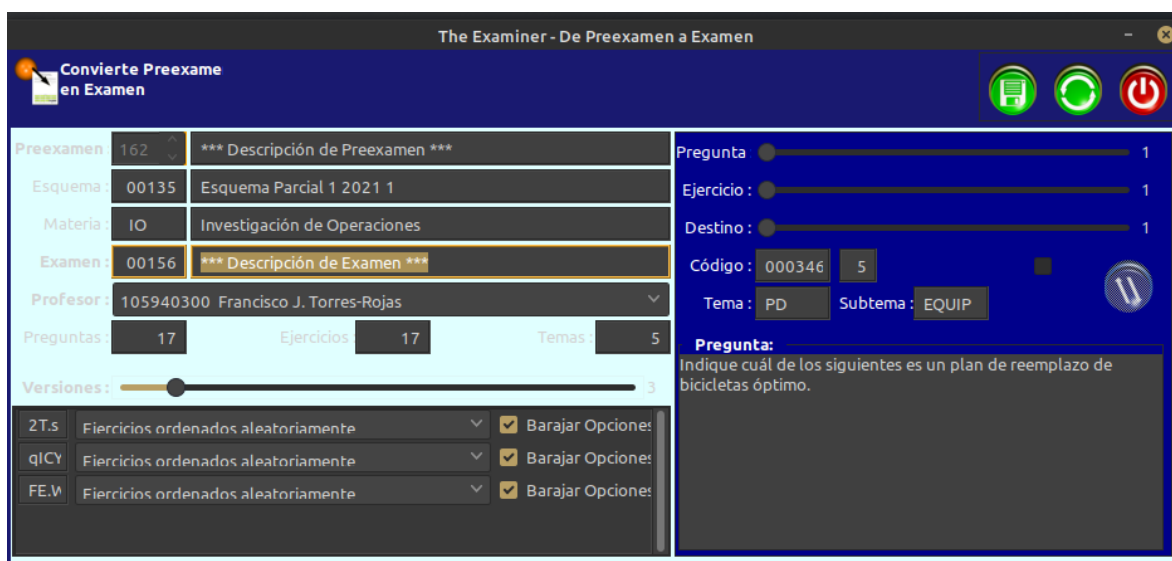
- d. Refinamiento: en caso de que el usuario considere la posibilidad de modificar el preexamen que ha elegido, puede utilizar la opción de refinamiento, con la cual podría eliminar y agregar preguntas. En la Figura No. 14 se puede ver un ejemplo de la pantalla en cuestión.

FIGURA NO. 14: PANTALLA DE REFINAMIENTO DE PREEXÁMENES



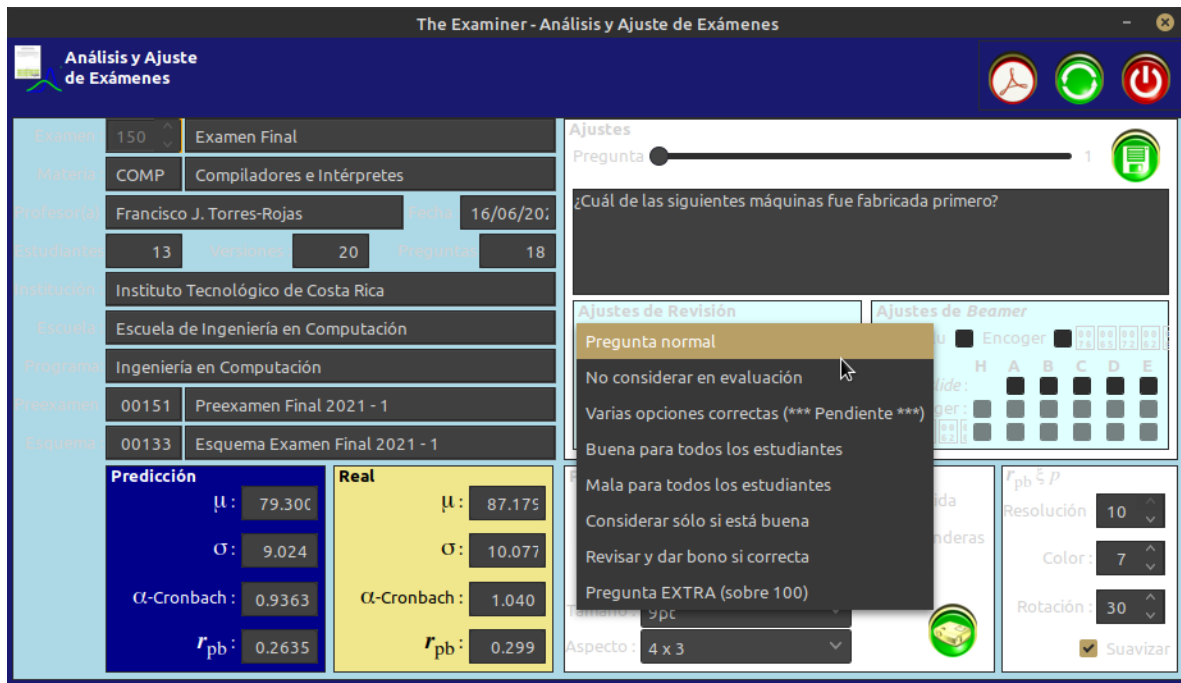
- e. Exámenes: en esta opción, el usuario puede escoger un preexamen y convertirlo a examen y tiene la posibilidad de generar versiones de este, en cuanto al orden de las preguntas y “barajar” las opciones de respuesta única, como se muestra en la Figura No. 15 .

FIGURA NO. 15: PANTALLA PARA CONVERTIR PREEXAMEN EN EXAMEN



- f. Imprime examen: con esta opción se puede generar el examen en formato PDF, incluyendo la hoja de repuestas.
4. Revisión: la sección de revisión cuenta con 3 opciones:
    - a. Revisión de exámenes: el usuario puede incluir de forma manual las respuestas de cada estudiante que participó en la prueba.
    - b. Análisis y ajuste de exámenes: en caso de que el usuario detecte alguna anomalía, como podría ser una pregunta mal redactada, el software permite realizar ajustes colectivos con varias opciones, como se muestra en la Figura No. 16.

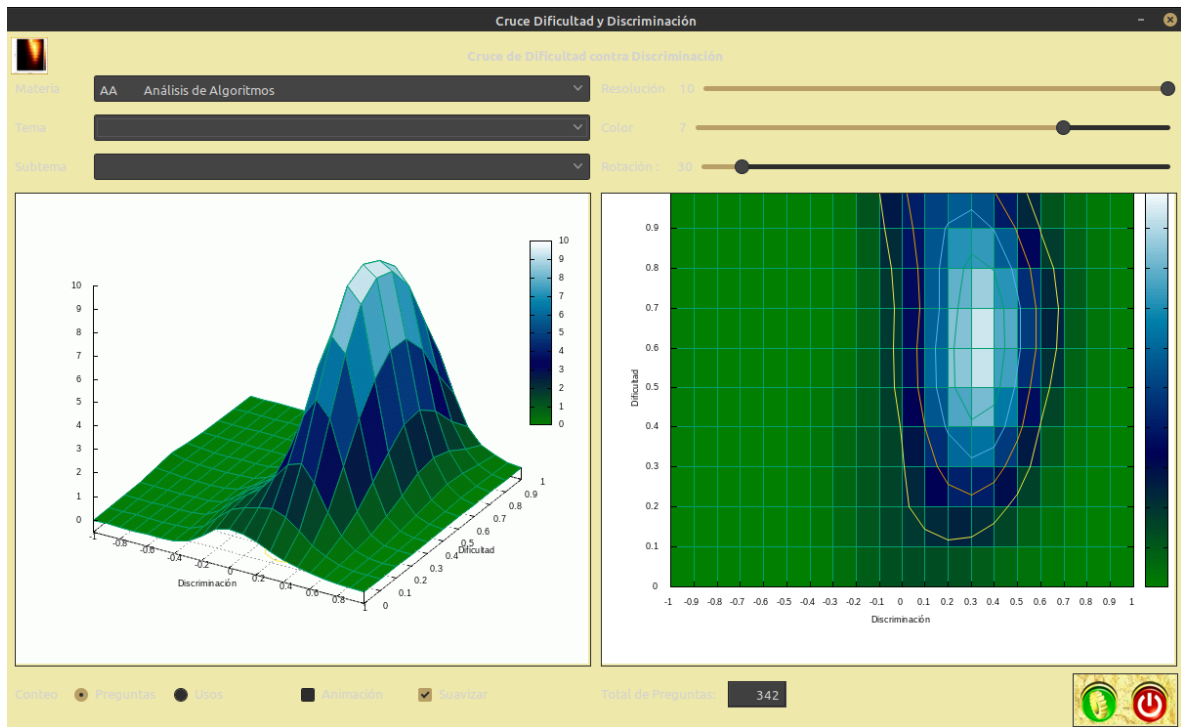
FIGURA NO. 16: PANTALLA DE ANÁLISIS Y AJUSTE DE EXÁMENES



- c. Actualización de estadísticas: una vez que se revisa el examen, con esta opción se actualizan las tablas con información histórica en la base de datos.
5. Reportes: la sección de reportes incluye la posibilidad de generar archivos en formato PDF de las materias, de los ejercicios, del análisis de las preguntas y la lista de esquemas, preexámenes y exámenes registrados en la base de datos. Adicionalmente, se pueden generar gráficos de relieve respecto a preguntas y su comportamiento en términos de discriminación y dificultad. En la Figura No. 17 se presenta un ejemplo de ese tipo de gráfico.



FIGURA NO. 17: EJEMPLO DE GRÁFICO DE PREGUNTAS SEGÚN DISCRIMINACIÓN Y DIFICULTAD



6. Limpieza: las opciones de limpieza se refieren a operaciones de administración de la base de datos para: eliminar exámenes, eliminar preexámenes y eliminar esquemas.
7. Ayuda: en la versión actual, la opción de ayuda únicamente muestra la versión del software actual (versión 0.6)

## ANEXO NO.2: PROCEDIMIENTO PARA RECREACIÓN DE LAS NOTAS DE UN ESTUDIANTE (CON R Y CONECTÁNDOSE A LA BASE DE DATOS):

```
##### RECREACION DE NOTAS DE UN ESTUDIANTE #####
##### Identifica las respuestas de un alumno respecto a la
##### versión que contestó y lo califica con base en las
##### respuestas de la base de datos de preguntas
library(RPostgreSQL)
library("lrm")
library("eRm")
library("psych")
library("difR")
library("TAM")
library("dplyr")
library("stringr")
library(tidyverse)

# Como encabezado, me conecto a la base de datos
con <- dbConnect (PostgreSQL(),user="felipe",password="judase03",dbname="EXAMENES")

## Comienza el análisis del "examen #1":
## como primer paso, se toma el primer examen de la base de datos que es el siguiente:

examen <- dbGetQuery(con, "select * from ex_examenes where codigo_examen = '00001'")
head(examen)

## Este examen se realizó con 3 versiones y 55 preguntas:
preguntas_versiones <- dbGetQuery(con, "select * from ex_examenes_preguntas where examen = '00001'")
preguntas_versiones

## El examen fue contestado por 9 alumnos, que fueron los siguientes:
respuesta <- dbGetQuery(con, "select * from ex_examenes_respuestas where examen = '00001'")
respuesta

## Existe un string en la tabla anterior en la que se guarda la secuencia de respuestas del alumno, ## que
respondió en una versión específica de examen. A continuación, la versión y el string de
## respuestas:
respuestas_alumno <- dbGetQuery(con, "select version, respuestas from ex_examenes_respuestas where
examen = '00001'")
respuestas_alumno <- as.character(respuestas_alumno$respuestas)
respuestas_alumno
respuestas_split <- strsplit(respuestas_alumno, split="")
respuestas_separadas <- strsplit(respuestas_alumno[1], "")
respuestas_separadas <- unlist(respuestas_separadas)

## Voy a filtrar una pregunta, basado en la versión 3SIK, pregunta 1:
## pregunta_01 <- dbGetQuery(con, "select * from ex_examenes_preguntas where examen =
##'00001' and ejercicio = '000235' and codigo_pregunta= '000259'")
##pregunta_01
# Ahora, en la versión 3SIK, pregunta 1, la respuesta correcta es la C, que corresponde a la opción_4. ##
Significa que en la versión MYiD, la corecta sería la 31 posición A y en z-UJ, pregunta 41, posición ## C
```

```

# Entonces, cómo se que la correcta en la parametrizada en el ejercicio-pregunta?????
# Para ello voy y reviso el ejercicio y la pregunta dentro del ejercicio y determino que la
# parametrización dice E (colàgeno)

## La respuesta correcta del ejercicio 235. pregunta 259 es :
e235_p259 <- dbGetQuery(con, "select respuesta from bd_texto_preguntas where codigo_unico_pregunta =
'000259'")
e235_p259

# La respuesta es "colàgeno", que se parametrizó en la opción "E", Que equivale a un 4. Entonces, #en la
secuencia de la versión de examen, adonde está el 4?, por ejemplo en el examen versión 3SIK, #está en la
posición C, el alumno tendría que haber respondido una C.
# Entonces, se sabe que la respuesta de la pregunta de un enunciado es una letra, que finalmente #es un alias
de número (A=0, B=1, C=2, D=3 y E=4)
# Siendo así, voy a crear un objeto con todos los ejercicios-preguntas y sus respuestas:

ejercicios_preguntas_respuestas <- dbGetQuery(con, "select codigo_consecutivo_ejercicio,
codigo_unico_pregunta, respuesta from bd_texto_preguntas")
ejercicios_preguntas_respuestas
dim(ejercicios_preguntas_respuestas)[1]

numeros <- sample(0,size = dim(ejercicios_preguntas_respuestas)[1], replace=TRUE)
ejercicios_preguntas_respuestas <- cbind(ejercicios_preguntas_respuestas,numeros)

ejercicios_preguntas_respuestas$numeros <-
replace(ejercicios_preguntas_respuestas$numeros,ejercicios_preguntas_respuestas$respuesta == "B", 1 )

ejercicios_preguntas_respuestas$numeros <-
replace(ejercicios_preguntas_respuestas$numeros,ejercicios_preguntas_respuestas$respuesta == "C", 2 )

ejercicios_preguntas_respuestas$numeros <-
replace(ejercicios_preguntas_respuestas$numeros,ejercicios_preguntas_respuestas$respuesta == "D", 3 )

ejercicios_preguntas_respuestas$numeros <-
replace(ejercicios_preguntas_respuestas$numeros,ejercicios_preguntas_respuestas$respuesta == "E", 4 )

ejercicios_preguntas_respuestas[ejercicios_preguntas_respuestas$codigo_unico_pregunta == '000259',]

## En este paso, voy a hacer el merge; "este merge tiene los insumos para encontrar el vector de ## respuestas
por version de examen"
# En este bucle lo que voy a hacer es reemplazar letra por la letra correspondiente a la posición de #la
respuesta correcta. Por ejemplo, si la respuesta es "E", entonces el número de respuesta correcta #es 4,
entonces necesito ubicar en qué opción está el número 4. Si estuviera en la opción_01, la #respuesta correcta
de esa versión de examen sería "A"

colnames(preguntas_versiones)
colnames(ejercicios_preguntas_respuestas)
colnames(ejercicios_preguntas_respuestas)[2] <- "codigo_pregunta"

merge01 <- merge(preguntas_versiones, ejercicios_preguntas_respuestas, by = "codigo_pregunta")
#head(merge01)

```

```
letras <- sample("A",size = dim(merge01)[1], replace=TRUE)
merge01 <- cbind(merge01,letras)
```

# En este bucle lo que voy a hacer es reemplazar letra por la letra correspondiente a la posición de # la respuesta correcta.

```
for (i in 1: dim(merge01)[1]) {
  if (merge01$opcion_1[i] == merge01$numeros[i]) {
    merge01$letras[i] <- "A"
  } else {
    if (merge01$opcion_2[i] == merge01$numeros[i]) {
      merge01$letras[i] <- "B"
    } else {
      if (merge01$opcion_3[i] == merge01$numeros[i]) {
        merge01$letras[i] <- "C"
      } else {
        if (merge01$opcion_4[i] == merge01$numeros[i]) {
          merge01$letras[i] <- "D"
        } else {
          if (merge01$opcion_5[i] == merge01$numeros[i]) {
            merge01$letras[i] <- "E"
          }
        }
      }
    }
  }
}
}
```

```
comprobacion01 <- merge01[merge01$codigo_pregunta == '000259',]
comprobacion01 #Esta comprobación SI funcionò
```

## En este segmento, replico código de "The Examiner", sobre el ex de calificaciones, para el examen ## 00001 y versión MlyD, para construir el vector de respuestas de un examen y versión específica

```
fran_01 <- dbGetQuery(con, "SELECT version, respuesta, posicion, codigo_pregunta, opcion_1, opcion_2,
opcion_3, opcion_4, opcion_5 from ex_examenes_preguntas, bd_texto_preguntas where examen = '00001'
and version = 'z-UJ' and codigo_pregunta = codigo_unico_pregunta order by posicion")
```

```
numeros <- sample(0,size = dim(fran_01)[1], replace=TRUE)
fran_01 <- cbind(fran_01,numeros)
```

```
fran_01$numeros <- replace(fran_01$numeros,fran_01$respuesta == "B", 1 )
fran_01$numeros <- replace(fran_01$numeros,fran_01$respuesta == "C", 2 )
fran_01$numeros <- replace(fran_01$numeros,fran_01$respuesta == "D", 3 )
fran_01$numeros <- replace(fran_01$numeros,fran_01$respuesta == "E", 4 )
letras<- sample("A",size = dim(fran_01)[1], replace=TRUE)
fran_01 <- cbind(fran_01,letras)
```

```
for (i in 1: dim(fran_01)[1]) {
  if (fran_01$opcion_1[i] == fran_01$numeros[i]) {
    fran_01$letras[i] <- "A"
  }
}
```





1. ¿Cuál es el origen de la palabra *álgebra*?
- a) Significa "números en letras" en árabe.
  - b) Viene del latín y significa "espíritu del aire".
  - c) Fue tomado del título de un libro de texto persa.
  - d) Nombre de un matemático persa muy importante en la Edad Media.
  - e) Nombre de una ciudad en Uzbekistan donde se cree que se inició el álgebra.
- 
2. La palabra *algoritmo* proviene de:
- a) Persa al-Jwarizmi.
  - b) Italiano para *ingenio*.
  - c) Latín para *solución*.
  - d) Error de impresión de Fibonacci tratando de escribir "logaritmo".
  - e) Árabe *al mudee'*, la luz.
- 
3. En la Edad Media, se denominaba de esta manera a una persona que hacía cálculos usando un ábaco.
- a) agnóstico
  - b) apotecario
  - c) computador
  - d) abacista
  - e) algorista
- 
4. ¿Cómo se definía usualmente la palabra inglesa *algorism*?
- a) Proceso de hacer aritmética usando números arábigos.
  - b) Intenso dolor causado por los números.
  - c) Procedimiento para resolver un problema.
  - d) Receta de cocina que incluía pesos y medidas de los ingredientes.
  - e) Demostración matemática de un teorema.
- 
5. ¿Cuál se considera que es el algoritmo más antiguo conocido?
- a) Problema de las 7 Reinas.
  - b) Axiomas de Euclides.
  - c) Algoritmo de Euclides.
  - d) Piedra, Papel o Tijera.
  - e) Teorema de Pitágoras.
-

## ANEXO NO.4: PROCEDIMIENTO PARA ANÁLISIS DE 50 EXÁMENES REALES CON MAYOR PARTICIPACIÓN DE ESTUDIANTES

```
#####
##### REALIZA EL ANALISIS DE LOS 50 EXAMENES #####
##### CON MAYOR CANTIDAD DE ALUMNOS #####
##### REPORTADOS EN LA BASE DE DATOS #####
#####

library(psych)
library(readxl)
library(ltm)
library(Cronbach)
library(Rmisc)

rm(list=ls())

# Windows
setwd("C:/Users/felip/OneDrive/Documentos/MEGA/MEGAsync/PP-UCR")

examenes <- c("ex00002.csv", "ex00004.csv", "ex00008.csv", "ex00009.csv", "ex00039.csv", "ex00044.csv",
             "ex00048.csv", "ex00050.csv", "ex00060.csv", "ex00065.csv", "ex00068.csv", "ex00075.csv",
             "ex00080.csv", "ex00081.csv", "ex00083.csv", "ex00085.csv", "ex00089.csv", "ex00090.csv",
             "ex00091.csv", "ex00092.csv", "ex00095.csv", "ex00096.csv", "ex00098.csv", "ex00101.csv",
             "ex00102.csv", "ex00103.csv", "ex00105.csv", "ex00106.csv", "ex00108.csv", "ex00110.csv",
             "ex00112.csv", "ex00113.csv", "ex00117.csv", "ex00120.csv", "ex00121.csv", "ex00122.csv",
             "ex00124.csv", "ex00125.csv", "ex00126.csv", "ex00127.csv", "ex00128.csv", "ex00130.csv",
             "ex00131.csv", "ex00135.csv", "ex00137.csv", "ex00138.csv", "ex00139.csv", "ex00140.csv",
             "ex00146.csv", "ex00147.csv")

# Calcula media real, desviaci?n e intervalos de confianza
medias <- data.frame(examen= character(), Li_media = numeric(), media_real = numeric(),
                    Ls_media=numeric(), desvest =numeric())
for(i in 1:length(examenes))
{
  archivo <- read.csv(file=examenes[i], sep = ";",header=T)
  archivo <- archivo[,-1]
  prom_filas <- rowMeans(archivo)*100
  prom_exam <- mean(prom_filas)
  n <- length(prom_filas)
  var <- var(prom_filas)*(n-1)/n
  desv <- sqrt(var)
  me <- 1.96 * desv/sqrt(n)
  li_media <- prom_exam - me
  ls_media <- prom_exam + me
  medias[i,1] <- examenes[i]
  medias[i,2] <- li_media
  medias[i,3] <- prom_exam
  medias[i,4] <- ls_media
  medias[i,5] <- desv
}

```



```

# Calcula alpha para los 50 examenes
alphas <- data.frame(examen= character(), Li_alpha = numeric(), alpha = numeric(), Ls_alpha=numeric())

for(i in 1:length(examenes))
{
  archivo <- read.csv(file=examenes[i], sep = ";",header=T)
  archivo <- archivo[,-1]
  a <- cronbach.alpha(archivo)
  it <- cron.ci(archivo, conf = 0.95, type = "logit", B = 1000)
  alphas[i,1] <- examenes[i]
  alphas[i,2] <- it[[1]]
  alphas[i,3] <- a$alpha
  alphas[i,4] <- it[[2]]
}

# Calcula correlaci?n biserial para cada item de UN EXMANEN(del paquete ltm)
archivo <- read.csv(file="ex00004.csv", sep = ";",header=T)
archivo <- archivo[,-1]

coef_bis <- function(tabla, level = 2) {
  score <- rowSums(tabla)
  tabla <- data.frame(tabla)
  coef <- lapply(tabla, biserial.cor, x = score, level = 2, use = "complete.obs")
  status <- ifelse(coef >= .2, "Aceptable", "No aceptable")
  data.frame(
    item = names(coef),
    rbis = matrix(coef),
    status = as.character(status)
  )
}

Rpb <- coef_bis(archivo)

```

## ANEXO NO.5: PROCEDIMIENTO PARA ANÁLISIS GENERACIÓN DE MATRICES BINARIAS A PARTIR DE PREGUNTAS, ESTUDIANTES Y RESPUESTAS.

```

from enum import Enum
import os
import random

#define los elementos de informacion que se pueden obtener de las respuestas de los estudiantes
InfEst = Enum('Inf_Est', ['EXAMEN', 'VERSION', 'RESPUESTAS'])

#función que recibe el nombre de un archivo de texto con datos por línea
#Cada dato separado por ";", cada registro separado por "\n"

def cargarArchivo(nombreArchivo):
    print("Procesando archivo: ", nombreArchivo, " ...")
    numRegistro = 1
    with open(nombreArchivo, "r", encoding="utf-8") as datos:
        valores = []
        for linea in datos:
            valores.append([x for x in linea.strip().split(";")])
            numRegistro+=1
    print(str(numRegistro), " registros procesados!")
    return valores

# getRespuestaAPregunta:funcion que retorna la respuesta de un codigo de pregunta
#     localizada en RespuestasCorrectas
# devuelve una letra entre A y E si la encuentra
# devuelve None en caso de no hallarla.

def getRespuestaAPregunta(codigo_pregunta):
    for reg in respuestasCorrectas:
        if reg[2] == codigo_pregunta:
            return reg [8]
    return None

#decodificarLetra retorna el valor numerico asociado a una letra de respuesta
#devuelve un valor entre 0 y 4 si la letra está entre A y E
#devuelve -1 en caso contrario

def decodificarLetra(letra):
    if letra in "ABCDE":
        return ord(letra) - 65
    return -1

# getEstudiante:funcion que retorna los datos de un estudiante localizado
#     en RespuestasAlumnos
# devuelve el registro completo de los datos del estudiante
# devuelve None en caso de no hallarla.
def getEstudiante(codEstudiante):
    for reg in respuestasAlumnos:
        if reg[1] == codEstudiante:

```

```

        return reg[:10]
    return None

# getDatosEstudiante:
# Dado que previamente se ha recuperado un registro de las respuestas de un estudiante
# se puede recuperar información particular 1: Examen 2:Version 3:Respuestas
# Cualquier otro dato a solicitar retorna None
def getDatosEstudiante(tipoInfo, registro):
    if tipoInfo == InfEst.EXAMEN.value:
        return registro[0]
    elif tipoInfo == InfEst.VERSION.value:
        return registro[3]
    elif tipoInfo == InfEst.RESPUESTAS.value:
        return registro[8]
    else:
        return None

def getRespuestaEstudiante(codEstudiante, numeroPregunta):
    reg = getEstudiante(codEstudiante)
    if reg != None:
        respuestas = getDatosEstudiante(InfEst.RESPUESTAS.value, reg) #3 solicita las respuestas
        if respuestas != None:
            return respuestas[numeroPregunta]
        else:
            return None
    return None

def getFilaEstudianteEnMatriz(codEst, matriz):
    for i in range(len(matriz)):
        if matriz[i][0] == codEst:
            return i
    return -1

def getColumnaPreguntaEnMatriz(codPregunta, matriz):
    for j in range(len(matriz[0])):
        if matriz[0][j] == codPregunta:
            return j
    return -1

"""
getVersionesAplicadas construye un diccionario con las distintas versiones de examen que se han aplicado.
Para cada versión (key del diccionario), se asocia una lista con los códigos de preguntas que tiene esa versión
"""
def getVersionesAplicadas():
    versiones={}
    #crea un diccionario con todas las versiones distintas aplicadas
    for i in respuestasAlumnos:
        if i[3] not in versiones: #la posición 3 tiene el código de versión
            versiones[i[3]] = []
    #llena cada versión con los códigos de preguntas contenidas
    for key in versiones:
        datos = versiones[key]
        for examen in formatoExamen:

```

```

        if examen[1] == key:
            datos.append(examen[5])
        versiones[key] = datos
    del versiones["version"]
    return versiones

def marcaPreguntasAplicadas(versiones, matriz):
    for i in range(1, len(matriz)):
        codEstudiante = matriz[i][0] # ubicado en i-esimo estudiante
        #obtiene registro de las respuestas del estudiante
        estudActual = getEstudiante(codEstudiante)

        #identifica la version de examen realizada por el estudiante
        verRealizada = getDatosEstudiante(InfEst.VERSION.value, estudActual)

        #recupera la lista de preguntas contenidas en la version realizad
        preguntas = versiones[verRealizada]
        for codPregunta in preguntas:
            posPregunta = getColumnaPreguntaEnMatriz(codPregunta, matriz) #ubica la posicion de la pregunta en matriz
            matriz[i][posPregunta] = 1 # marca como evaluada al estudiante

def getCodigoPregunta(version, posicion):
    for pregunta in formatoExamen:
        if pregunta[1] == version and pregunta[2] == posicion:
            return pregunta[5] #codigo_pregunta
    return None

def getPosCorrectaEnExamen(version, codigoPregunta, respuestaNumero):
    for pregunta in formatoExamen:
        if pregunta[1] == version and pregunta[5] == codigoPregunta:
            if pregunta[6] == str(respuestaNumero):
                return "A"
            elif pregunta[7] == str(respuestaNumero):
                return "B"
            elif pregunta[8] == str(respuestaNumero):
                return "C"
            elif pregunta[9] == str(respuestaNumero):
                return "D"
            elif pregunta[10] == str(respuestaNumero):
                return "E"
    return None

def calificaRespuestasEstudiantes(version, posicion, codigoPregunta, posCorrectaVersion, matriz):
    for estudiante in respuestasAlumnos:
        if estudiante[3] == version: #el estudiante hizo esa version del examen
            datos= getEstudiante(estudiante[1])
            respuestaEstudiante = getRespuestaEstudiante(estudiante[1], posicion-1)
            enMatriz = 1
            if posCorrectaVersion != respuestaEstudiante:
                enMatriz = 0

        posPregunta = getColumnaPreguntaEnMatriz(codigoPregunta, matriz) #ubica la posicion de la pregunta en matriz

```

```

    matriz[int(estudiante[1])] [posPregunta] = enMatriz
    print ("Version ", version, " Estudiante: ", estudiante[1], "Esp:" ,posCorrectaVersion,"
Resp:",respuestaEstudiante," Matriz:",matriz[int(estudiante[1])] [posPregunta] )

def comprobarRespuestas(versiones, matriz):
    for version in versiones:
        print("Comprobando Respuestas de version ", version)
        totalPreguntas = len(versiones[version]) #cantidad de preguntas de la version
        for i in range (1,totalPreguntas+1):
            codigoPregunta= getCodigoPregunta (version, str(i))
            print("Obteniendo respuesta correcta de pregunta ", str(i),"-",codigoPregunta)

            respuestaLetra = getRespuestaAPregunta(codigoPregunta)
            respuestaNumero = decodificarLetra(respuestaLetra)
            posCorrectaVersion = getPosCorrectaEnExamen(version, codigoPregunta, respuestaNumero)
            print("Revisando respuestas de estudiantes a pregunta ", codigoPregunta, " version ", version)
            calificaRespuestasEstudiantes(version, i, codigoPregunta, posCorrectaVersion, matriz)

def crearMatriz (datosEstudiantes, datosPreguntas):
    global matriz
    matriz = []
    cantFilas = len(respuestasAlumnos)
    cantColumnas = len(respuestasCorrectas)
    for i in range (cantFilas):
        matriz.append([])
        for j in range (cantColumnas):
            matriz[i].append( -1 )

    #redefine los encabezados de filas
    matriz[0][0] = ""
    for i in range (1,cantFilas):
        matriz [i][0] = datosEstudiantes[i][1] # se almacenan codigos de estudiante
    for j in range (1,cantColumnas):
        matriz [0][j] = datosPreguntas[j][2] # se almacenan codigos de preguntas
    return matriz

def imprimirMatriz (matriz):
    for i in range (len(matriz)):
        for j in range (len (matriz[i])):
            print (matriz[i][j], sep = " ", end="")
        print("\n")

def guardarMatriz(matriz, nombreArchivo):
    file = open(nombreArchivo, "w")
    for i in range (len(matriz)):
        for j in range (len (matriz[i])):
            if j != len(matriz[i])-1:
                separador=";"
            else:
                separador = "\n"
            file.write (str(matriz[i][j]) + separador)
    file.close();

```

```

def menu():
    opcionSalir = False
    while opcionSalir != True:
        print("\n-----")
        print("1.Consultar respuesta correcta de una pregunta por código")
        print("2.Consultar datos de un estudiante por código")
        print("3.Consultar código y versión de exámenes realizados por un estudiante")
        print("4.Consultar respuesta a una pregunta de un examen ")
        print("5.Versiones aplicadas a los estudiantes")
        print("6.Crear matriz de estudiantes/preguntas ")
        print("7.Paso 1- Marca matriz con preguntas contestadas")
        print("8.Llenar matriz - paso 2: si las preguntas han sido contestadas correctamente o no")
        print("9.Guardar matriz en archivo")
        print("0.Terminar ")
        print("-----")
        opcion = int(input("Seleccionar una opcion:"))
        if opcion == 0:
            opcionSalir = True
        elif opcion == 1:
            codPregunta = input("Código de pregunta: ")
            #buscando la respuesta correcta de la pregunta indicada
            respAlfabetica= getRespuestaAPregunta(codPregunta)
            print ("\nCodigo Pregunta: ", codPregunta, " Respuesta: ", respAlfabetica,
                "(" , decodificarLetra(respAlfabetica), ")")
        elif opcion == 2:
            codEstudiante = input("Código de estudiante: ")
            #buscando un estudiante particular
            regEstudiante= getEstudiante(codEstudiante)
            print ("\nDatos del estudiante:\n", regEstudiante)
        elif opcion ==3:
            codEstudiante = input("Código de estudiante: ")
            #buscando un estudiante particular
            regEstudiante= getEstudiante(codEstudiante)
            examen = getDatosEstudiante (InfEst.EXAMEN.value, regEstudiante)
            version = getDatosEstudiante (InfEst.VERSION.value, regEstudiante)
            print ("\nExamen ", examen, " Version: ", version)
        elif opcion == 4:
            codEstudiante = input("Código de estudiante: ")
            numeroPregunta = int(input("Ingrese el número de pregunta a consultar: "))
            resultado = getRespuestaEstudiante(codEstudiante, numeroPregunta-1)
            print ("\nEstudiante ", codEstudiante, " contestó pregunta ", numeroPregunta, " con ", str(resultado))
        elif opcion == 5:
            versiones = getVersionesAplicadas()
        elif opcion == 6:
            matriz = crearMatriz(respuestasAlumnos, respuestasCorrectas)
        elif opcion == 7:
            marcaPreguntasAplicadas(versiones, matriz)
        elif opcion == 8:
            comprobarRespuestas(versiones, matriz)
        elif opcion == 9:
            guardarMatriz(matriz,"matriz.csv")

```

```
#programa principal
global matriz
global formatoExamen
global respuestasCorrectas
global respuestasAlumnos
global versiones

print ("Cargando archivos...")
""" El archivo FormatoExamen.csv tiene la estructura:
examen;version;posicion;ejercicio;secuencia;codigo_pregunta;opcion_1;opcion_2;opcion_3;opcion_4;opcion_5"""
#se carga el archivo FormatoExamen en una lista de trabajo
formatoExamen=cargarArchivo("FormatoExamen.csv")

""" El archivo RespuestasCorrectas.csv tiene la estructura:
codigo_consecutivo_ejercicio; secuencia_pregunta;codigo_unico_pregunta;
tipo;creacion_month;creacion_day;dificultad_estimada;respuesta;numero_respuesta"""

#se carga el archivo RespuestasCorrectas en una lista de trabajo
respuestasCorrectas=cargarArchivo("RespuestasCorrectas.csv")

""" El archivo RespuestasAlumnos.csv tiene la estructura:
10 campos obligatorios
examen;estudiante;nombre;version;correctas;porcentaje;nota_ajustada;nota_final;respuestas;preguntas

Pueden haber Ri, para i=1..62 campos opcionales por registro - se cargan pero van a ser ignorados
porque se extrae de las respuestas"""

#se carga el archivo RespuestasAlumnos en una lista de trabajo
respuestasAlumnos=cargarArchivo("RespuestasAlumnos.csv")

matriz = []
versiones = {}

#paso 5
versiones = getVersionesAplicadas()
#paso 7
matriz = crearMatriz(respuestasAlumnos, respuestasCorrectas)
#paso 8
marcaPreguntasAplicadas(versiones, matriz)
#paso 9
comprobarRespuestas(versiones, matriz)
#paso 10
guardarMatriz(matriz,"matrizFinal.csv")
```

## ANEXO NO.6, PROCEDIMIENTO PARA ANÁLISIS DE EXÁMENES SIMULADOS CON ENFOQUE TCT.

```
#####
##### REALIZA EL ANALISIS DE 18 EXAMENES #####
##### (3 POR MATERIA), TOMANDO EN CUENTA #####
##### LAS PREGUNTAS CON MAYOR CANTIDAD DE #####
##### RESPUESTAS (MAYOR A 19 POR PREGUNTA) #####
#####

library(psych)
library(readxl)
library(ltm)
library(Cronbach)

rm(list=ls())

# Windows
setwd("C:/Users/felip/OneDrive/Documentos/MEGA/MEGAsync/PP-UCR")

# Linux
# setwd("~/MEGAsync/ESOLFER")

examenes <- c("AA_01.csv","AA_02.csv","AA_03.csv",
             "BCM_01.csv","BCM_02.csv","BCM_03.csv",
             "COM_01.csv","COM_02.csv","COM_03.csv",
             "FOC_01.csv","FOC_02.csv","FOC_03.csv",
             "INN_01.csv","INN_02.csv","INN_03.csv",
             "IO_01.csv","IO_02.csv","IO_03.csv")

# Calcula media real, desviaci?n e intervalos de confianza
medias_simulacion <- data.frame(examen= character(), Li_media = numeric(), media_real = numeric(),
Ls_media=numeric(), desvest =numeric())

for(i in 1:length(examenes))
{
  archivo <- read.csv(file=examenes[i], sep = ";",header=T)
  archivo <- archivo[,-1]
  prom_filas <- rowMeans(archivo)*100
  prom_exam <- mean(prom_filas)
  n <- length(prom_filas)
  var <- var(prom_filas)*(n-1)/n
  desv <- sqrt(var)
  me <- 1.96 * desv/sqrt(n)
  li_media <- prom_exam - me
  ls_media <- prom_exam + me
  medias_simulacion[i,1] <- examenes[i]
  medias_simulacion[i,2] <- li_media
  medias_simulacion[i,3] <- prom_exam
  medias_simulacion[i,4] <- ls_media
}
```



```
medias_simulacion[i,5] <- desv
}

# Calcula alpha para 18 exámenes simulados
alphas_simulacion <- data.frame(examen= character(), Li_alpha = numeric(), alpha = numeric(), Ls_alpha=numeric())

for(i in 1:length(examenes))
{
  archivo <- read.csv(file=examenes[i], sep = ";",header=T)
  archivo <- archivo[,-1]
  a <- cronbach.alpha(archivo)
  it <- cron.ci(archivo, conf = 0.95, type = "logit", B = 1000)
  alphas_simulacion[i,1] <- examenes[i]
  alphas_simulacion[i,2] <- it[[1]]
  alphas_simulacion[i,3] <- a$alpha
  alphas_simulacion[i,4] <- it[[2]]
}
```

## ANEXO NO.7, PROCEDIMIENTO PARA ANÁLISIS DE EXÁMENES SIMULADOS CON ENFOQUE TRI.

```
#####
##### REALIZA EL ANALISIS #####
##### TRI #####
##### DE LOS 50 EXAMENES #####
##### CON MAYOR CANTIDAD DE ALUMNOS #####
##### REPORTADOS EN LA BASE DE DATOS #####
#####

library(readxl)
library(eRm)
library(mirt)
library(ltm)
library(Cronbach)
rm(list=ls())

# Windows
setwd("C:/Users/felip/OneDrive/Documentos/MEGA/MEGAsync/PP-UCR")

examenes <- c("ex00002.csv", "ex00004.csv", "ex00008.csv", "ex00009.csv", "ex00039.csv", "ex00044.csv",
             "ex00048.csv", "ex00050.csv", "ex00060.csv", "ex00065.csv", "ex00068.csv", "ex00075.csv",
             "ex00080.csv", "ex00081.csv", "ex00083.csv", "ex00085.csv", "ex00089.csv", "ex00090.csv",
             "ex00091.csv", "ex00092.csv", "ex00095.csv", "ex00096.csv", "ex00098.csv", "ex00101.csv",
             "ex00102.csv", "ex00103.csv", "ex00105.csv", "ex00106.csv", "ex00108.csv", "ex00110.csv",
             "ex00112.csv", "ex00113.csv", "ex00117.csv", "ex00120.csv", "ex00121.csv", "ex00122.csv",
             "ex00124.csv", "ex00125.csv", "ex00126.csv", "ex00127.csv", "ex00128.csv", "ex00130.csv",
             "ex00131.csv", "ex00135.csv", "ex00137.csv", "ex00138.csv", "ex00139.csv", "ex00140.csv",
             "ex00146.csv", "ex00147.csv")

modelos <- data.frame(examen= character(),
                      bondad_ajuste = numeric(),
                      informacion = numeric(),
                      AIC_01 = numeric(),
                      AIC_02 = numeric())

#### ANALISI TRI
#### Informacion + bondad de ajuste

for(i in 1:length(examenes))
{
  datos <- read.csv(file=examenes[i], sep = ";", header=T)
  datos <- datos[,-1]
  modelos[i,1] <- examenes[i]

  modelo <- rasch(datos, IRT.param = T)

  inf <- information(modelo, c(-4,4))
  modelos[i,3] <- inf$InfoTotal

  bondad_ajuste <- GoF.rasch(modelo)
```

```
modelos[i,2] <- bondad_ajuste$p.value

}

### TRABAJANDO CON 1 EXAMEN PARA POBLAR EL "FOR"

datos <- read.csv(file="ex00146.csv", sep = ";",header=T)
datos <- datos[,-1]
#modelos[50,1] <- c("ex00147.csv")
  modelo <- rasch(datos, IRT.param = T)
summary(modelo)
  inf <- information(modelo, c(-4,4))
#modelos[50,3] <- inf$InfoTotal
  bondad_ajuste <-GoF.rasch(modelo)
# modelos[49,2] <- bondad_ajuste$p.value

for(i in 1:length(examenes))
{
  datos <- read.csv(file=examenes[i], sep = ";",header=T)
  datos <- datos[,-1]
  modelo <- rasch(datos, IRT.param = T)
  inf <- information(modelo, c(-4,4))
  bondad_ajuste <-GoF.rasch(modelo)

  modelos[i,1] <- examenes[i]
  modelos[i,2] <- bondad_ajuste$p.value
  modelos[i,3] <- inf$InfoTotal
}
}
```