Automated Grading with AI Using Rubrics in Computer Science 1

Inés Friss de Kereki; Ismael Garrido
Universidad ORT Uruguay, Uruguay, *kereki_i@ort.edu.uy*, ismael.garrido@fi365.ort.edu.uy

***Abstract*– A precise evaluation is essential to provide valuable information that allows for adjustments in teaching and improvements in the learning process. Rubrics are assessment tools that define clear and specific criteria for grading assignments. This study presents a detailed comparison of the evaluation of Computer Science 1 assignments using rubrics applied manually by different instructors and through Artificial Intelligence (AI) tools on the same assignments. The effectiveness and consistency of the corrections made by AI are analyzed in comparison to those made by instructors and among the instructors themselves. Additionally, surveys conducted with the instructors involved in the grading process are presented to gather their perceptions regarding the accuracy, usefulness, and impact of AI on their work, as well as its comparison with traditional assessment methodologies.**

***Keywords*—Computer Science 1, rubrics, assessment, Artificial Intelligence.**

Corrección automatizada con IA mediante el uso de Rúbricas en Programación 1

Inés Friss de Kereki; Ismael Garrido
Universidad ORT Uruguay, Uruguay, *kereki_i@ort.edu.uy*, ismael.garrido@fi365.ort.edu.uy

Resumen–Una evaluación precisa es fundamental para ofrecer información valiosa que permita ajustar la enseñanza y mejorar el proceso de aprendizaje. Las rúbricas son herramientas de evaluación que definen criterios claros y específicos para calificar trabajos. Este estudio presenta una comparación detallada sobre la evaluación de trabajos en Programación 1 utilizando rúbricas en forma manual por diferentes docentes y a través de herramientas de Inteligencia Artificial (IA) sobre los mismos trabajos. Se analiza la efectividad y coherencia de las correcciones hechas por IA frente a las de los docentes y entre los mismos docentes. Además, se presentan encuestas realizadas a los docentes involucrados en la corrección, con el fin de conocer sus percepciones sobre la precisión, utilidad y el impacto de la IA en su trabajo, así como su comparación con las metodologías tradicionales de evaluación.

Palabras clave—Programación I, rúbricas, evaluación, Inteligencia Artificial.

I. Introducción

Evaluar es importante porque permite medir el progreso y la comprensión de los estudiantes, identificar áreas de mejora, y ajustar la enseñanza para asegurar un aprendizaje efectivo. Además, una evaluación adecuada garantiza una retroalimentación objetiva y equitativa, orientando tanto a docentes como a estudiantes en el proceso educativo.

Una rúbrica es un elemento de evaluación que enumera los criterios para un trabajo y articula también los niveles de calidad para cada criterio [1].

Programación 1 (P1) es una asignatura del 1er semestre de carreras de Ingeniería en Sistemas y afines y presenta los conceptos básicos de la programación. Este trabajo es continuación y ampliación de [2]. En [2], se describe la herramienta "AutoGrade", donde se cargan las consignas de evaluaciones, las rúbricas, las fotos de las soluciones de los estudiantes que son transcriptas o el propio código y el sistema genera el reporte detallado de la aplicación de dichas rúbricas, o sea, por cada elemento de la rúbrica en cada solución se indica si se cumple o no y justifica la evaluación. Se analizan distintos modelos de Inteligencia Artificial (IA) a utilizar para la evaluación de un parcial específico de P1 a partir de rúbricas y en función de las comparaciones que realiza, recomienda el uso de Llama3-70B [3, 4] y se brindan además recomendaciones sobre el diseño de rúbricas. En particular, se analizó un ejercicio resuelto por 24 estudiantes y se obtuvo 85% de coincidencia global entre IA y el docente.

El aporte de este trabajo es ampliar el análisis a diferentes ejercicios típicos de P1 y sus rúbricas para tratar de identificar qué aspectos evaluaría mejor la IA. Se comparan correcciones de los mismos trabajos por diferentes docentes aplicando una rúbrica y luego las correcciones realizadas por medio de herramientas de IA con la misma rúbrica, con un análisis que profundiza el presentado en [2]. Se trata de evaluar la eficacia y consistencia de correcciones de IA en comparación con la evaluación de los docentes y entre los propios docentes.

Se extiende también el alcance del estudio y se incluye la perspectiva de los docentes. Las encuestas tienen como objetivo recoger sus opiniones y experiencias respecto al uso de rúbricas y de la IA en el proceso de evaluación. Se trata de conocer su percepción de la precisión y utilidad, su impacto en la carga de trabajo, y cómo se compara con las metodologías tradicionales de corrección.

El trabajo está organizado de la siguiente forma: se describe en primer lugar el curso de Programación 1, luego se presentan consideraciones sobre las rúbricas, el diseño de las rúbricas para ejercicios típicos de programación, y se presenta la experimentación realizada. Se realiza el análisis detallado y se incluyen las encuestas. Finalmente, se ofrecen conclusiones.

II. Curso de Programación 1

Programación I es una asignatura de 1er semestre de las carreras de Ingeniería y Licenciatura en Sistemas, Electrónica, Eléctrica y Telecomunicaciones en la Universidad ORT Uruguay. Tiene por objetivos desarrollar el pensamiento computacional y las habilidades básicas de programación para resolver problemas no triviales utilizando un lenguaje de programación ampliamente utilizado. Los principales temas son: variables, expresiones, estructuras de control, estructuras de datos simples, funciones y procedimientos. El lenguaje utilizado es JavaScript.

El curso dura 15 semanas, con 4 horas de clases teóricas y 2 horas de trabajo en laboratorio cada semana. La evaluación incluye 2 parciales individuales (de 15 y 40 puntos respectivamente) y dos trabajos de varias semanas de duración en equipos de 2 estudiantes (de 10 y 35 puntos) Se requieren 70 puntos o más para aprobar el curso. Cada parcial tiene una duración de 2 horas, en papel, sin el uso de materiales adicionales, y consiste en ejercicios que deben resolverse en JavaScript. Los parciales implican el desarrollo de código, no hay preguntas teóricas.

El curso se dicta cada semestre, los grupos son de 25-30 alumnos cada uno y se distribuyen al azar. Habitualmente en el segundo semestre del año hay 4-6 grupos.

Al diseñar las evaluaciones se trata de asegurarse que los diferentes docentes interpreten el ejercicio de la misma forma e incluir ejemplos para ayudar a entender lo que se pregunta [5]. En el curso, todos los docentes revisan previamente las propuestas y se unifican las sugerencias que otorgan.

III. Sobre las Rúbricas

Las rúbricas son herramientas poderosas tanto para el aprendizaje como la evaluación, reducen el tiempo evaluando trabajos y son fáciles de usar y explicar [6]. Crear rúbricas es la parte difícil, usarlas es la parte fácil [6]. Usando rúbricas, el evaluador puede evaluar consistente y objetivamente [7].

Las rúbricas pueden ser construidas de diferentes formas: en forma holística (donde no se hace énfasis en cuánto deducir por un error individualmente) o en forma analítica (donde se dan un conjunto detallado de criterios, cada uno con su puntaje y se suman los puntajes para el resultado final) [8]. Pueden ser de uso general o específicas de la tarea. Las de uso general pueden ser usadas para un conjunto de tareas o un curso, mientras que las específicas están orientadas a los requerimientos particulares de una tarea [9]. Tienen generalmente 3 componentes: la dimensión ("indicador de desempeño"), descriptor y escala ("nivel de desempeño") [10].

Las rúbricas deben ser fiables, lo que significa que debe generar calificaciones similares cuando la utilicen diferentes personas [1]. Los docentes deben iterar para mejorar la rúbrica hasta que produzca resultados que sean consistentes con lo que espera el evaluador y ajustarla hasta que produzca resultados convincentes [8].

Son usadas frecuentemente para evaluar programación [7, 11]. Pueden diferir significativamente los docentes al calificar ejercicios de desarrollo de código en exámenes de Programación 1, entre los factores clave se señala la variabilidad en cómo se construyen las rúbricas [8]. Un aspecto recomendado por [6] para diseñar rúbricas es evitar el lenguaje no claro: todos los términos deben ser claramente definibles (ejemplo: "trabajo creativo" es difícil de definir). En [2] se recomienda, entre otros aspectos, evaluar los criterios en forma binaria: cumple/ no cumple y para las soluciones que difieren de lo "previsto" (por ejemplo, utilizan estructuras no vistas en el curso como "map" o "filter"), utilizar criterios explícitos para detectarlos. Este último punto está relacionado con lo indicado en [12], en cuanto señala que una dificultad con el enfoque tradicional para otorgar calificaciones según un "punto por declaración correcta" es que los estudiantes son evaluados en función de la similitud de su solución con el esquema de respuestas. En Programación, un mismo ejercicio puede ser resuelto de muchas formas diferentes, con el mismo resultado, por lo cual ese enfoque podría no ser aplicable [12]. Puede haber además inconsistencias dentro de la evaluación de la misma solución al evaluar por diferentes personas, pues podrían hacer énfasis en diferentes elementos, como la sintaxis o el diseño [12]. Para abordar este problema, es necesario un conjunto de rúbricas de evaluación con el fin de proporcionar flexibilidad para soluciones críticas y creativas entre los estudiantes, así como para mejorar la consistencia en la calificación entre los instructores y asistentes docentes [12].

En este trabajo se focaliza en rúbricas específicas con enfoque analítico, incluyendo los 3 componentes (indicador o ítem, descriptor y nivel o ponderación), y se construyeron realizando varias iteraciones. Inicialmente, cada rúbrica fue generada conjuntamente por un docente experimentado, incluyendo sugerencias brindadas por Chat GPT-4o [13].

IV. Parcial de Programación 1 y Rúbricas

El primer parcial del curso de Programación 1 del 2do semestre de 2024 fue utilizado para este trabajo. Ese parcial se realiza en la semana 6 del curso y trata sobre estructuras de control, variables simples, funciones y "strings", que son temas típicos para este tipo de curso.

Consiste en 3 ejercicios. Los ejercicios son elaborados por un docente experimentado y son ajustados y validados por los demás docentes de la Cátedra.

El primer ejercicio involucra lectura de datos, el uso de estructuras de control y variables simples. Se trata de solicitar datos, acumular, contar y emitir los resultados finales, estos temas se trabajan desde la semana 1. El segundo ejercicio implica el uso funciones y "strings", temas vistos habitualmente en la semana 3-4 del curso. El tercer ejercicio implica el uso de estructuras anidadas, visto a partir de la semana 5.

En particular, los ejercicios propuestos son:

a) Ejercicio 1: Un parking desea un programa para controlar lo cobrado en un día. Inicia sin dinero. Sólo opera en efectivo y tiene diferentes tarifas: la tarifa 1 es sin cargo, la tarifa 2 es de $80 la hora y la tarifa 3 es de $130 la hora. Al comienzo del programa se ingresa el total de dinero que reporta el cajero al fin del día. Se ingresa luego cada uno de los tiques del día. De cada tique se ingresa el tipo de tarifa (1, 2 o 3) y cantidad de horas de ese tique. El fin de ingreso es tipo de tarifa con valor -1. Al final de todos los ingresos, debe mostrarse con "alert" si la cantidad de dinero del cajero coincide con lo calculado a partir de los tiquetes (mostrar "CORRECTO"), si lo indicado por el cajero es menor a lo registrado en los tiquetes (mostrar "FALTA") o si lo indicado por el cajero supera a lo registrado en los tiquetes (mostrar "SOBRA"). Además, mostrar con "alert" el promedio de horas entre todos los tiquetes. No hay inconsistencias. Hacer un programa en JS para ser probado en un "snippet" que realice el proceso descripto.

b) Ejercicio 2: Implementar en JavaScript una función que recibe una frase que se asume contiene solamente los valores "0" y "1" y retorna un nuevo "string" según este proceso: por cada "1" de la frase original, en el nuevo "string" va "0" y por cada "0" de la frase original, va "11". La firma es: "function proceso(frase)". Además, ejemplificar el uso de

la función, solicitando la frase al usuario y mostrando el resultado por consola y

c) Ejercicio 3: Hacer un programa en JavaScript que lee un valor tope y muestre una línea por consola por cada número entre 1 y tope (ambos inclusive). Cada línea contiene: el número, el símbolo "*" y luego todos los números pares positivos menores a ese número en forma decreciente, separados por un espacio. (Tanto para el ejercicio 2 como para el 3, se incluyeron ejemplos concretos.)

Para el ajuste inicial de la rúbrica, los docentes generaron soluciones "ficticias" y también se utilizó Chat GPT-4o [13] para obtener soluciones, indicando como "prompt" que hiciera soluciones suponiendo que era estudiante de 1er semestre. Se generaron 15 soluciones diferentes para cada ejercicio y con ellos se ajustó la rúbrica.

La rúbrica final utilizada con cada uno de los criterios se presenta en las Tablas I, II y III, detallándose por cada criterio su nombre o dimensión, descripción y ponderación o peso (la suma de las ponderaciones es 1). Se incluyeron en todas las rúbricas un criterio "de restricción", que no lleva puntos, pero se usa para ubicar fácilmente soluciones que podrían no aplicar los criterios estándar, considerando [2].

TABLA I
RÚBRICA DEL EJERCICIO 1

Número y Nombre del Ítem		Descripción	Peso
1	Leer dato inicial	El programa solicita y captura correctamente al comienzo el total de dinero del cajero.	1/14
2	Uso y fin de iteración	El programa reconoce correctamente el valor de -1 como fin de ingreso de tiquetes y termina el bucle de entrada.	2/14
3	Lectura de cantidad	Dentro del bucle de iteración, debe leerse la cantidad del tique. No importan las demás lecturas de datos.	1/14
4	Lectura del tipo	Dentro del bucle de iteración, debe leerse el tipo del tique. No importan las demás lecturas de datos.	1/14
5	Total de dinero	El programa calcula correctamente el total de dinero acumulado según los tiquetes ingresados (tarifa 2 a $80 y tarifa 3 a $130). Puede ir contando la cantidad o calcular al final. Asumir que el valor inicial está disponible y es válido.	2/14
6	Conteo para promedio	El programa va contando la cantidad de tiquetes para calcular el promedio.	2/14
7	Validación del resultado	El programa compara correctamente el total de dinero calculado con el reportado por el cajero, mostrando "FALTA", "SOBRA" o "CORRECTO" según corresponda. Las condiciones son mutuamente excluyentes. Si se usan "if" anidados o "else" verificar especialmente las condiciones.	2/14
8	Promedio de horas	El programa muestra el promedio de horas entre todos los tiquetes ingresados, dividiendo la suma total de horas por la cantidad.	2/14
9	Salida	Los resultados deben mostrarse todos con "alert", no con "console.log".	1/14
10	Restricción	No se usa "map", "find", "forEach", "reduce", "replace", ni expresiones regulares.	0

TABLA II
RÚBRICA DEL EJERCICIO 2

Número y Nombre del Ítem		Descripción	Peso
1	Recorre caracteres del "string"	Recorre los caracteres del "string".	1/8
2	Acceso dentro del rango	Al recorrer no accede a posiciones fuera del string, esto es, accede sólo entre 0 y largo-1 del "string".	1/8
3	Acceso a elementos	Accede correctamente a los elementos de la frase utilizando la notación [] o invocando al método "charAt".	1/8
4	Conversión de caracteres	Convierte correctamente de "0" en "11" y de "1" en "0".	1/8
5	Generación de "string" resultante	El "string" resultante se genera correctamente, concatenando en el orden correcto y está correctamente inicializado.	1/8
6	Retorno del "string"	La función retorna el "string" generado.	1/8
7	Ejemplo de solicitud	Ejemplifica solicitando la frase al usuario.	1/8
8	Invocación y muestra	Invoca a la función con el dato leído y muestra el resultado por consola, no con "alert".	1/8
9	Restricción	La función no incluye "map", "find", "forEach", "reduce", "replace", ni expresiones regulares.	0

TABLA III
RÚBRICA DEL EJERCICIO 3

Número y Nombre del Ítem		Descripción	Peso
1	Lectura correcta del valor tope	El programa debe solicitar y capturar correctamente el valor tope ingresado por el usuario.	1/8
2	Generación de la secuencia numérica	El programa debe generar una línea por cada número entre 1 y el valor tope, ambos inclusive, asegurándose de que el incremento sea de 1 en 1.	2/8
3	Cálculo de la línea	El programa debe construir cada línea correctamente, incluyendo el número actual, el símbolo "*", y todos los números pares positivos menores que el número actual, en orden decreciente. Si el programa genera todos los números menores al actual, debe asegurarse de verificar que sólo los números pares se incluyan en la línea final. El número de línea debe ser creciente.	2/8
4	Formato correcto de la salida	El programa debe mostrar cada línea de la salida en el formato correcto, con los números pares y sin errores en el formato. No importa si hay espacios adicionales al final.	2/8
5	Uso de "console"	El programa debe utilizar "console.log" para imprimir cada línea en la consola.	1/8
6	Restricción	No se usa "map", "find", "forEach", "reduce", "replace", ni expresiones regulares.	0

Tanto el sistema "AutoGrade" [2] como los docentes evalúan cada parcial en función de idénticos criterios, indicando "0" si no lo cumple o "1" si lo cumple.

V. EXPERIMENTACIÓN

La experimentación se realizó en setiembre de 2024, con 4 grupos matutinos de 25-30 alumnos distribuidos al azar. Varios días antes del parcial se hizo una reunión con los docentes y se les informó de todos los detalles del proceso (tomado de fotos de los parciales, planillas a completar, cómo revisar transcripciones de las fotos y evaluaciones, y la encuesta final a rellenar).

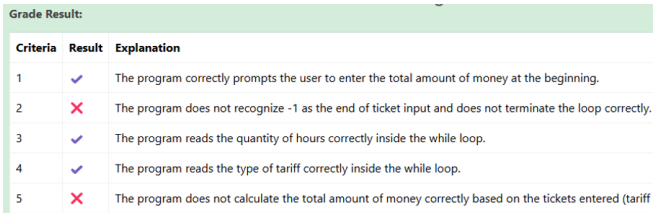
Para la recolección de datos de las correcciones se dispuso una planilla compartida, donde cada docente registró sus puntuaciones y comentarios adicionales.

En el parcial, se les pidió a los alumnos escribir con la mayor claridad posible y resolver un ejercicio por hoja, de forma de facilitar la posterior corrección. Se les indicó que se corregiría manualmente y también en forma automática en modo de testeo.

El mismo día del parcial se tomaron y subieron las fotos de todos los parciales (105 parciales, total 315 evaluaciones). El tiempo total de tomar las fotos y subirlas a "AutoGrade" [2] fue de unas 2 horas.

Se amplió el sistema "AutoGrade" [2] al que se le agregó la opción de "transcribir en lote", la que permite seleccionar un conjunto de fotos y solicitar la transcripción. Se realizó al día siguiente del parcial una revisión de las transcripciones. Cada uno de los 5 ayudantes de Cátedra revisó y ajustó un lote de transcripciones, corrigiendo errores de la lectura, como omisiones de "}" o ";". Al finalizar esta revisión, se realizó la evaluación automatizada de todos los parciales con la rúbrica, o sea, los docentes ya disponían de la corrección de IA poco tiempo después de realizado el parcial. Los resultados se visualizan en forma resumida (ver Fig. 1), donde se indica por cada resolución de ejercicio y por cada criterio si es correcto o no y en forma detallada, con la explicación (ver Fig. 2).

Fig. 1 Presentación (vista parcial) de resumen de aplicación de rúbrica en "AutoGrade"


Grade Result:

Criteria	Result	Explanation
1	✓	The program correctly prompts the user to enter the total amount of money at the beginning.
2	✗	The program does not recognize -1 as the end of ticket input and does not terminate the loop correctly.
3	✓	The program reads the quantity of hours correctly inside the while loop.
4	✓	The program reads the type of tariff correctly inside the while loop.
5	✗	The program does not calculate the total amount of money correctly based on the tickets entered (tariff

Fig. 2 Presentación (vista parcial) de detalle de aplicación de rúbrica en "AutoGrade"

Cada docente evaluó sus parciales y puso en la planilla compartida sus resultados. En la revisión final de las 315 evaluaciones, se detectaron en total 5 casos de fotos muy borrosas, 13 casos de ejercicios que no fueron resueltos, y un caso que quedó faltante de subir la corrección del docente en la planilla. Estos casos fueron descartados de la evaluación general. Se totaliza así 296 evaluaciones de ejercicios. Hubo tres casos donde el docente indicó puntaje "0.5" en vez de "0" o "1" en un criterio (que fueron ajustados conversando con el docente).

VI. ANÁLISIS

La fiabilidad [12] refiere a la consistencia de los puntajes de evaluación. En una prueba fiable, un estudiante esperaría obtener la misma calificación independientemente de cuándo realizó la evaluación, cuándo fue corregida o quién la calificó.

McHugh [14] indica que para medir la confiabilidad entre evaluadores se puede utilizar el porcentaje de acuerdo y el Kappa de Cohen (*K*).

El porcentaje de acuerdo Pr(a) calcula el porcentaje de coincidencias entre evaluadores. Se realiza una matriz con los diferentes evaluadores y cada fila representa las variables de las que se recolectó información. El número de coincidencias sobre el número de variables medidas da el porcentaje de acuerdo entre los evaluadores. Es una medida directa, refleja el nivel real de acuerdo sin considerar el azar. Permite identificar variables que pueden ser problemáticas [14].

El Kappa de Cohen según refiere [14], es un estadístico útil para verificar la confiabilidad entre evaluadores. Los resultados se interpretan según [15] en: <0 acuerdo pobre, entre 0.00 y 0.20 acuerdo bajo, entre 0.21 y 0.4 es acuerdo regular, entre 0.41 y 0.60 es acuerdo moderado, entre 0.61 y 0.80 es acuerdo sustancial y entre 0.81 y 1 es acuerdo casi perfecto. En [14] se indica cómo realizar el cálculo del estadístico *K*. La fórmula básica referida por [14, 15] es (fórmula 1):

$$K = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (1)$$

siendo: Pr(a) es el acuerdo observado (la proporción de veces en que los evaluadores coinciden en sus evaluaciones) y Pr(e) la probabilidad de acuerdo esperada por azar, calculada en función de las frecuencias marginales de cada categoría de evaluación. Se utiliza una tabla que muestra la frecuencia con que cada evaluador asigna cada categoría. Para ese cálculo, a partir de los ejemplos que brinda [14], se elaboró una tabla similar a la Tabla IV, donde "a" es la cantidad de veces que el evaluador 1 indicó que ese criterio era falso y el evaluador 2 indicó que ese criterio era falso, "b" es la cantidad de veces que el evaluador 1 indicó verdadero y el evaluador 2 indicó falso, "c" es para la cantidad de casos del evaluador 1 que indicó falso y el evaluador 2 indicó verdadero y "d" es la cantidad de veces que el evaluador 1 indicó verdadero y el evaluador 2 también. "n" es la suma de a, b, c, y d. Se calculan los valores de columna marginal ("cm1": a+c, "cm2": b+d) y fila marginal ("rm1": a+b, "rm2":c+d). Luego se aplica la fórmula (2) para calcular Pr(e).

TABLA IV
CÁLCULO DE VALORES MARGINALES POR CRITERIO

		Evaluador 1		
		0	1	fila marginal (rm)
Evaluador 2	0	a	b	a+b rm1
	1	c	d	c+d rm2
	columna marginal (cm)	a+c cm1	b+d cm2	a+b+c+d n

$$Pr\,(e) = \frac{\left(\frac{cm1*rm1}{n} + \frac{cm2*rm2}{n}\right)}{n} \quad (2)$$

Se tuvo en cuenta que el tamaño de la muestra es mayor o igual a 30, según refiere [14].

El análisis del porcentaje de acuerdo se realiza en distintos enfoques: global (considerando el total de coincidencias entre todas las correcciones de todos los ejercicios), por ejercicio individual y por criterio individual.

En el enfoque global, en total se evaluaron 296 ejercicios, correspondiendo a 103 del primero, 100 del segundo y 93 del tercero. El ejercicio 1 tiene 10 criterios, el ejercicio 2 tiene 9 y el ejercicio 3 tiene 6. En total, son 2488 criterios evaluados, donde coinciden lo asignado por el docente con lo asignado por la IA en 1968 criterios (79%) y difieren en 520 (21%).

Analizando por ejercicio, en el caso del Ejercicio 1, se obtuvo 85% de coincidencia entre el docente y la IA, en el Ejercicio 2, se tuvo 76% y en el Ejercicio 3, se obtuvo 75% (ver Tabla V). Estos valores son cercanos a los reportados en [2]: 85%.

TABLA V
ACUERDO GLOBAL Y POR EJERCICIO

	Resultados		
Enfoque	Criterios evaluados	Coinciden	Difieren
Global	2488	1968 79%	520 21%
Ejercicio 1 (muestra: 103) 10 criterios	1030	871 85%	159 15%
Ejercicio 2 (muestra: 100) 9 criterios	900	680 76%	220 24%
Ejercicio 3 (muestra: 93) 6 criterios	558	417 75%	141 25%

Analizando por cada criterio de cada ejercicio, se ve en la Tabla VI que el ejercicio 1 tiene los niveles más altos de acuerdo, con la mayoría de los criterios (8 de 10) alcanzando porcentajes por encima del 75%. Hay cuatro criterios que superan el 90%. El criterio con menor acuerdo es el 7, que refiere a comparar para determinar si "sobra dinero", si es "correcto" o si "falta". Implica "ifs" anidados o chequeos independientes escritos cuidadosamente para no confundir los casos.

En el ejercicio 2, los criterios están en un rango más bajo, de 70-80% aproximadamente, indicando un acuerdo moderado, sin criterios que alcancen el 90%. El más bajo es el criterio 7 relativo a invocar la función.

El ejercicio 3 tiene niveles mixtos de acuerdo: si bien algunos criterios tienen valores de acuerdo alto (como por ejemplo el criterio 1: leer datos), otros son mucho más bajos (por ejemplo el criterio 2 relativo a generar la secuencia, por debajo del 60%).

TABLA VI
ACUERDO POR CRITERIO

	Coincidencias		
	Ejercicio 1 103 casos	Ejercicio 2 100 casos	Ejercicio 3 93 casos
Criterio 1	101 98%	71 71%	85 91%
Criterio 2	81 79%	78 78%	54 58%
Criterio 3	97 94%	78 78%	66 71%
Criterio 4	89 86%	74 74%	63 68%
Criterio 5	78 76%	75 75%	65 70%
Criterio 6	92 89%	81 81%	84 90%
Criterio 7	65 63%	64 64%	-
Criterio 8	69 67%	72 72%	-
Criterio 9	100 97%	87 87%	-
Criterio 10	99 96%	-	-

En cada ejercicio, se calculó el valor *K* para cada criterio según la fórmula presentada (fórmula (2)). Respecto al ejercicio 1 se observa en la Tabla VII que los criterios 7 y 8 son los de menor valor, esto sugiere que sólo una pequeña parte del acuerdo observado supera lo que podría esperarse por azar (acuerdo "regular" según [15]). El criterio 7 del ejercicio 1, como se indicó, es el de ver el resultado para establecer si sobra, es correcto o falta dinero. El criterio 8 refiere al cálculo correcto del promedio. Los de mayor coincidencia del ejercicio 1 son el criterio 1 (que refiere al ingreso inicial del total de dinero) y el 3 (que es sobre el ingreso de cantidad), que corresponden a los pasos iniciales de muchos ejercicios. Esos criterios parecen ser los más fáciles de evaluar de manera consistente.

De este ejercicio, los criterios con mayor desacuerdo están relacionados a elementos que implican mayor comprensión del problema. No se observaron valores en el rango de concordancia pobre o baja de concordancia, lo que sugiere un desempeño positivo en términos de coherencia (ver Tabla VIII).

En relación al ejercicio 2, los criterios con valores más bajos de Kappa son: el criterio 9 (que refiere al uso de otras estructuras), el criterio 7 (invocar la función) y el criterio 1 (recorrer toda la frase). En particular, en el criterio 9 (que tiene un porcentaje de acuerdo alto), Kappa es negativo, lo que

podría interpretarse que esas coincidencias podrían ser sólo por azar, no por un verdadero consenso. Revisando las diferencias, corresponden a soluciones con muchos errores o muy incompletas, y quizás esto generó una falta de consistencia en la valoración, ya que, al no encontrar elementos válidos para analizar, los docentes optaron por no detallar si incluía o no otras estructuras. Los demás criterios tuvieron concordancia moderada (41%-60%) con cinco observaciones y sustancial (61%-80%) con una observación (ver Tabla VIII), lo que parece indicar un nivel intermedio de consistencia entre los evaluadores.

En cuanto al ejercicio 3 (el de mayor complejidad pues requiere estructuras anidadas), el criterio más bajo es el 4, relacionado al formato adecuado (respondido correctamente por 34 de 93 casos). Los valores de concordancia (ver Tabla VII) son bajos, ya que predominan los rangos de concordancia baja (0%-20%) y regular (21%-40%), con muy pocos casos en los niveles superiores, lo que sugiere una baja consistencia entre los evaluadores en este caso específico.

TABLA VII
ANÁLISIS *K* POR CRITERIO

	Resultados Kappa		
	Ejercicio 1	Ejercicio 2	Ejercicio 3
Criterio 1	0.878	0.347	0.518
Criterio 2	0.542	0.560	0.254
Criterio 3	0.814	0.447	0.169
Criterio 4	0.536	0.480	0.158
Criterio 5	0.503	0.520	0.440
Criterio 6	0.744	0.611	0.166
Criterio 7	0.224	0.287	-
Criterio 8	0.397	0.449	-
Criterio 9	0.713	-0.069	-
Criterio 10	0.485	-	-

TABLA VIII
RANGOS DE CONCORDANCIA

	Cantidad de criterios		
Concordancia	Ejercicio 1	Ejercicio 2	Ejercicio 3
Rango hasta 20% (pobre o baja concordancia)	0	1	3
Rango 21%-40% (regular)	2	2	1
Rango 41%-60% (moderada)	4	5	2
Rango 61%-80% (sustancial)	2	1	0
Rango mayor igual 81% (casi perfecta)	2	0	0

A modo de verificación, se volvieron a corregir los parciales del grupo del docente más novato por parte de un docente experimentado y se aplicaron los mismos cálculos de *K* entre ambas correcciones, sobre el mismo subconjunto al azar, obteniéndose valores similares en todos los casos.

Como cierre del análisis, se observa que las tareas relacionadas con cálculos simples y estructuras condicionales, como en el ejercicio 1, resultan más manejables para los estudiantes y para el diseño de rúbricas. Por otro lado, los problemas que implican transformaciones complejas de datos o la generación de patrones, como en los ejercicios 2 y 3, presentan mayores desafíos. Estos desafíos no sólo afectan a los estudiantes en el proceso de resolución, sino que también requieren que los docentes presten especial atención al momento de diseñar rúbricas que evalúen de manera precisa y justa estas habilidades más avanzadas.

VII. ENCUESTAS A DOCENTES

Se les consultó a los 4 docentes participantes sobre su experiencia y opiniones respecto al uso de rúbricas y la IA. Se exploraron diversos aspectos, incluyendo la utilidad percibida de las rúbricas, la precisión y eficiencia de la IA a través de "AutoGrade" [2], así como los cambios en la confianza hacia el uso de IA tras su uso.

Tres de los cuatro docentes indicaron que les resulta muy útil el uso de rúbricas y el otro indicó "útil". En general, la opinión sobre el uso de rúbricas fue positiva.

En relación con cómo fue su proceso de corrección, todos indicaron que comenzaron corrigiendo de manera manual antes de comparar o verificar con la IA. Hubo variaciones en el proceso de comparación y ajuste: tres docentes lo hicieron en paralelo (de a un estudiante por vez) y el otro lo hizo en etapas: comparando de a 4 estudiantes por vez. En general, se destacó una valoración positiva sobre la flexibilidad del proceso de corrección al contar con la IA.

La corrección por IA fue evaluada como "Precisa" por tres de ellos y "Muy precisa" por el cuarto docente. Las otras opciones disponibles no seleccionadas eran: "neutral", "imprecisa", "muy imprecisa" o "prefiero no responder". Todos indicaron que la corrección por IA detectó errores que no habían visto. Uno de ellos señaló que todos estaban bien indicados y los otros tres refirieron además que "algunos pocos errores estaban mal indicados". Las otras opciones eran "varios mal indicados", "mayoría mal indicados" o "prefiero no responder". Uno de los docentes comentó: "Considero valiosa la IA dentro de un proceso de doble validación, para verificar posibles errores no detectados en forma manual". A pesar de alguna leve discrepancia en los errores, el uso de la IA fue vista como un apoyo valioso para mejorar la precisión en la calificación.

Respecto a la rapidez en el proceso de calificación, tres de ellos indicaron que se ahorró tiempo de proceso de calificación y uno indicó que igual. Un docente además comentó: "La validación de la corrección como una segunda pasada me parece que ahorra mucho tiempo". La percepción fue positiva respecto a la eficiencia del proceso con IA.

Previamente al uso de IA para la corrección, la IA les daba "confianza" (dos docentes), les resultaba "neutral" (un docente) y "poca confianza" (un docente), y luego de aplicarla, se transformaron en "mucha confianza" (un docente) y "confianza" (tres docentes), es decir, mejoró la apreciación en cuanto a la confianza. Todos coincidieron que se mejora el proceso de calificación utilizando la herramienta de IA en comparación con el método tradicional e indicar recomendarla para las demás instancias de evaluación, como parcial o

examen. El uso de IA generó un aumento en la confianza de los docentes en su incorporación.

Algunos aspectos destacados del uso de IA en el sistema "AutoGrade" [2] fueron: "Poder visualizar el detalle de cada corrección" y "La claridad con la que indica la IA las correcciones cuando el ejercicio no cumple alguno de los criterios predefinidos". Como mejoras sugieren que se debe prever tiempo para revisar las transcripciones para chequear más a fondo por posibles diferencias.

Los docentes indicaron también que incorporan un aspecto humano basado en su experiencia, que les permite evaluar no sólo los resultados objetivamente, sino también el esfuerzo y las circunstancias de los estudiantes y que no está contemplado en la corrección por IA. Esto es especialmente evidente en el caso de trabajos muy incompletos, donde las herramientas de IA suelen asignar una calificación de 0 al no cumplir con los criterios mínimos establecidos en las rúbricas. Sin embargo, los docentes, considerando factores como la motivación estudiantil y la prevención de posibles conflictos, tienden a otorgar un puntaje mínimo, como 0.5 o 1 punto, reconociendo el esfuerzo realizado e intentando fomentar una actitud positiva hacia el aprendizaje. Este mínimo ajuste no afecta significativamente el resultado final, ya que el estudiante igualmente no alcanzará la aprobación en el caso planteado, pero refleja un enfoque empático que la IA no puede ofrecer, al carecer del contexto de situaciones individuales y valorar aspectos subjetivos del proceso educativo que trascienden los parámetros estrictamente técnicos.

En resumen, la encuesta mostró que tanto el uso de rúbricas como el uso de IA son valorados positivamente por los docentes participantes. La IA se percibe como un recurso útil para mejorar la precisión y la eficiencia en la calificación.

VIII. Conclusiones y Trabajo Futuro

El uso de rúbricas por parte de los docentes en combinación con herramientas de IA resultó ser una estrategia eficaz para una evaluación coherente y precisa. En términos generales, se obtuvo una coincidencia del entorno del 79%, con valor mayor en el ejercicio 1 (85%).

Utilizar rúbricas proporciona una estructura uniforme y clara para la evaluación, fomentando que todos los docentes apliquen los mismos criterios de manera consistente. Por otro lado, al integrar una herramienta de IA, se añadió un nuevo punto de vista automatizado, que aplicó esos mismos criterios, proporcionando una perspectiva adicional, una corrección más precisa y un doble chequeo. Esto permitió validar y enriquecer el proceso de corrección. Los docentes recomendaron continuar con esta metodología, ya que perciben que optimiza el proceso de corrección y promueve una evaluación más justa y rigurosa.

Se tiene en cuenta que los resultados dependen en gran medida de la calidad del diseño de las rúbricas y de la interpretación de los errores en las soluciones de los estudiantes, así como del uso de herramientas de IA que están en constante desarrollo y su efectividad puede variar significativamente según sus versiones y capacidades específicas.

El análisis de los resultados por ejercicio y los valores de Kappa, desglosados por criterio, muestran que las rúbricas son efectivas (tanto para los docentes como para la IA) para evaluar ejercicios simples como el ejercicio 1, que involucra lectura de datos, el uso de estructuras de control sencillas y variables. Para ejercicios más avanzados, como el ejercicio 3 que requiere lógica más compleja, las rúbricas pueden presentar desafíos adicionales tanto en su diseño como en su aplicación. Los estudiantes enfrentaron mayores dificultades para la resolución, lo que puede deberse a la necesidad de integrar múltiples habilidades, como lógica no trivial y manipulación de estructuras complejas. Los docentes podrían beneficiarse de un diseño más refinado de las rúbricas para esos casos, asegurando que estas desglosen de manera clara y precisa los pasos necesarios para resolver problemas complejos y permitan evaluar enfoques diversos con mayor flexibilidad.

Una opción para analizar a futuro podría ser considerar enfoques de corrección por nivel, utilizando la IA para los ejercicios más simples y que los docentes se focalicen en los ejercicios de lógica más compleja.

Como otros trabajos futuros, se evaluará la posibilidad de poner a disposición de los estudiantes para practicar, el uso de "AutoGrade" [2] para los ejercicios simples, lo que permitiría realizar evaluación automatizada y retroalimentación instantánea por el propio estudiante, para colaborar en la mejora del proceso de aprendizaje.

También se procurará identificar y analizar los errores recurrentes en la resolución de ejercicios complejos, esto es, examinar sistemáticamente las soluciones de los estudiantes para detectar patrones de errores que puedan señalar dificultades conceptuales, lo que puede contribuir a diseñar mejores rúbricas para esos casos en futuras evaluaciones.

Otro elemento para considerar será realizar el parcial en computadora, según la disponibilidad de equipos, lo que facilitaría la recolección de datos para la corrección automatizada.

Agradecimientos

Este trabajo se hizo en el marco del "Fondo de Innovación en Proyectos de Enseñanza utilizando Inteligencia Artificial" de la Universidad ORT Uruguay, siendo uno de los proyectos seleccionados para recibir financiamiento.

Referencias

[1] H. Goodrich, "Teaching with Rubrics: the Good, the Bad, and the Ugly", College Teaching, January 2005, Vol 53, No. 1

[2] I. Friss de Kereki e I. Garrido, "AutoGrade: an AI-Based Assessment Tool for Computer Science 1", IEEE Educon 2025, marzo 2025, Uruguay

[3] Meta Llama3-70B, https://github.com/meta-llama/llama-Models/blob/main/models/llama3/MODEL_CARD.md

[4] Llama-3-Groq-70B, https://console.groq.com/docs/models

[5] Simon, J. Sheard, D. D'Souza, M. Lopez, A. Luxton-Reilly, I. H. Putro, P. Robbins, D. Teague, y J. Whalley, "How (not) to write an introductory

programming exam", 17th Australasian Computing Education Conference (ACE 2015), Sydney, Australia
[6] H. Goodrich, "Understanding Rubrics", Educational Leadership, Dic 1996
[7] D. Saito, R. Yajima, H. Washizaki, y Y. Fukazawa, "Validation of Rubric Evaluation for Programming Education", Educ. Sci., vol. 11, p. 656, 2021, doi: 10.3390/educsci11100656
[8] I. Albluwi, "A closer look at the differences between graders in Introductory Computer Science exams", IEEE Transactions on Education, vol. 61, no. 3, pp. 253-259, Aug. 2018
[9] M. Stegeman, E. Barendsen y S. Smetsers, "Designing a Rubric for feedback and code quality in Programming Courses", Proc. 16th Koli Calling Int. Conf. Computing Education Research, Koli, Finland, Nov. 2016, pp. 160-164, doi: 10.1145/2999541.2999555
[10] M. Dawood, K. A. Buragga, A. R. Khan y N. Zaman, "Rubric based assessment plan implementation for Computer Science program: A practical approach", 2013 IEEE Int. Conf on Teaching, Asessment and Learning for Engineering, Indonesia, 2013
[11] D. Saito, S. Kaieda, H. Washizaki y Y. Fukazawa, "Rubric for Measuring and visualizing the effects of learning Computer Programming for Elementary School Students", Journal of Inf. Technology Education: Innovation in Practice, 19, 203-227, 2020, https://doi.org/10.28945/4666
[12] A. Mustapha, N. A. Samsudin, N. Arbaiy, R. Mohamed e I. R. Hamid, "Generic Assessment Rubrics for Computer Programming Courses", The Turkish Online Journal of Educ. Technology, vol. 15, no. 1, Jan. 2016
[13] Chat GPT-4o, https://openai.com/index/chatgpt/
[14] M. L. McHugh, "Interrater reliability: the kappa statistic", Biochem. Med. (Zagreb), vol. 22, no. 3, pp. 276–282, Oct. 2012
[15] R. Landis y G. Koch, "The Measurement of Observer Agreement for Categorical Data", Biometrics, vol 33, No. 1, 1977, pp 159-174