

Jornadas de Automática

Evaluación automática en robótica industrial mediante MATLAB Grader

Elvira, C. *, Nájera, S., Otálora, P., Gil-Martínez, M., Rico-Azagra, J.

*Grupo de Ingeniería de Control, Departamento de Ingeniería Eléctrica, Universidad de la Rioja, Edificio Departamental –
C/San José de Calasanz, 31, 26004, Logroño, España*

To cite this article: Elvira, C., Nájera, S., Otálora, P., Gil-Martínez, M., Rico-Azagra, J. 2026. Automated assessment in industrial robotics using MATLAB Grader. *Jornadas de Automática*, 47.
<https://doi.org/10.17979/ja-cea.2026.47.13859>

Resumen

Este artículo analiza la integración realizada de MATLAB Grader en la asignatura "Control y Programación de Robots" en la Universidad de La Rioja, desde un uso meramente formativo en entorno de laboratorio hacia la evaluación integral de la misma. Se diseñó un itinerario formativo y evaluativo en esta herramienta, formado por ejercicios compuestos de tareas fragmentadas con dificultades crecientes, intentos limitados, autoevaluadas y ponderadas, y se aplicaron durante el curso 2025-2026. Se analizan los resultados cuantitativos obtenidos (22 alumnos evaluados), correlacionando el progreso durante las prácticas automatizadas con el rendimiento observado en los exámenes finales tradicionales tanto de convocatoria ordinaria como extraordinaria. Los datos revelan que la rigurosidad técnica de la evaluación automática actúa como un duro filtro, provocando tasas críticas de abandono sobre todo en bloques avanzados, lo que abre un debate clave sobre la persistencia estudiantil actual y la forma de abordar la evaluación mediante scripts en grupos de alumnos de asignaturas relacionadas con control, automatización y robótica.

Palabras clave: Educación en ingeniería, Robótica industrial, Cálculo simbólico, Evaluación automática, MATLAB Grader, Analíticas de aprendizaje.

Automated Assessment in Industrial Robotics Using MATLAB Grader

Abstract

This article analyzes the integration of MATLAB Grader into the "Robot Control and Programming" course at the University of La Rioja, progressing from purely formative use in a laboratory setting to comprehensive course assessment. A formative and evaluative learning path was designed using this tool, consisting of exercises composed of fragmented tasks with increasing difficulty, limited attempts, self-assessment, and weighting. This path was implemented during the 2025-2026 academic year. The quantitative results obtained (22 students assessed) are analyzed, correlating progress during the automated practical exercises with performance observed in traditional final exams, both regular and resit. The data reveal that the technical rigor of the automated assessment acts as a harsh filter, leading to critical dropout rates, especially in advanced modules. This raises a crucial debate about current student persistence and how to approach assessment using scripts in groups of students in courses related to control, automation, and robotics.

Keywords: Engineering education, Industrial robotics, Symbolic computation, Automated assessment, MATLAB Grader, Learning analytics.

1. Introducción

La formación en robótica industrial dentro de las titulaciones de Ingeniería Automática requiere dotar al

alumno de capacidades analíticas complejas para resolver el modelado de matrices de transformación homogénea, la cinemática de la orientación (muñecas de Euler), la cinemática de todos los eslabones de un robot mediante la notación de

Denavit-Hartenberg, la formulación de matrices Jacobianas analíticas y geométricas y los interpoladores de movimiento aplicados al control cinemático.

El uso tradicional de la corrección manual de scripts oculta ineficiencias críticas: falta de objetividad, dilatación temporal en el *feedback* recibido por el estudiante y la imposibilidad de evaluar de manera granular la sintaxis algorítmica. De acuerdo con Boada et al. (2021), la evaluación automatizada del código en sesiones de programación no solo optimiza la gestión del tiempo, sino que estandariza los criterios de corrección bajo entornos objetivos. Para resolver estas deficiencias, se ha apostado por la automatización integral de la docencia de laboratorio y su posterior integración con el resto de los procesos de evaluación académica. Los objetivos fundamentales son: facilitar el aprendizaje autónomo, reforzar la precisión técnica en cálculo simbólico, automatizar la corrección sumativa garantizando una objetividad matemática total (Sangwin, 2013) y analizar el comportamiento y la usabilidad del sistema por parte del estudiantado frente a herramientas automatizadas de corrección inflexible (Gaona, 2020).

2. Contexto de la asignatura e infraestructura

La experiencia llevada a cabo se contextualiza en la asignatura "*Control y Programación de Robots*", impartida en el primer cuatrimestre del tercer curso del Grado en Ingeniería Electrónica Industrial y Automática de la Universidad de La Rioja. La infraestructura tecnológica desplegada se compone de tres pilares integrados de manera transparente:

2.1. MATLAB Online y MATLAB Grader

Plataformas en la nube que permiten la ejecución de scripts y funciones sin instalación local, facilitando la estandarización del entorno informático del alumno. Como señala Terauds (2023), la versatilidad de MATLAB Grader para instanciar múltiples variantes de una misma tarea mitiga sustancialmente los problemas de copia cooperativa indeseada.

2.2. Symbolic Math Toolbox

Extensión utilizada para la definición de variables simbólicas (*sym*), funciones simbólicas (*symfun*), diferenciación analítica (*Jacobianos*) y simplificación de expresiones algebraicas complejas.

2.3. Integración Blackboard LTI 1.3

Protocolo de comunicación que permite la sincronización automática de las calificaciones obtenidas en MATLAB Grader directamente en la plataforma virtual, garantizando el suministro constante de *feedback* de aprendizaje en tiempo real (MathWorks, 2025).

3. Metodología y casos de estudio en prácticas

Durante el curso académico 2025-26, se evaluó a un grupo homogéneo de 22 alumnos matriculados activos. El plan metodológico consistió en la fragmentación de la asignatura en un itinerario secuencial de prácticas obligatorias corregidas automáticamente. Siguiendo las directrices

innovadoras de Vuksanovic et al. (2024), el entorno fue configurado para evaluar respuestas parcialmente correctas y evitar en lo posible el plagio, combinando esquemas de intentos limitados e ilimitados, así como evaluaciones tipo todo/nada y ponderaciones variables por complejidad de apartado.

El núcleo de aprendizaje automatizado se estructuró a partir de las siguientes unidades didácticas e hitos de implementación de software:

Práctica n.º 2:

Matrices de cambio de orientación en el plano (*Robot RR*) usando variables simbólicas y datos de tipo estructura (*struct*).

Práctica n.º 4:

Estudio analítico de una muñeca de Euler (*muñeca RRR*) y validación de propiedades de bases vectoriales ortonormales. Se fragmentó en 7 subtareas críticas (desde la definición simbólica simple hasta funciones numéricas integradas).

Práctica n.º 6:

Cinemática completa de un manipulador cilíndrico *RPP* (posición, velocidad y aceleración lineal/angular absoluta) utilizando tipos de datos complejos (*struct* y *dictionary*).

Práctica n.º 7:

Robot *SCARA* mediante notación *Denavit-Hartenberg* (DH). Construcción avanzada de objetos tipo tabla (*table*). Desarrollo de funciones para matrices de transformación homogénea parciales y globales, y resolución analítica del *Problema Cinemático Inverso* (PCI).

Práctica n.º 8:

Estudio de la matriz *Jacobiana* del robot *SCARA* mediante derivación analítica de velocidad lineal y geométrica, consolidada en una función global.

Práctica n.º 9:

Control cinemático mediante interpoladores lineales cúbicos y resolución de sistemas de ecuaciones lineales.

La práctica 1 se enfoca a un curso de familiarización con el entorno simbólico en Matlab (*Symbolic Toolbox*) y no se evalúa mientras que las prácticas 3 y 5 se enfocan al empleo real práctico de robots cartesianos y antropomórficos respectivamente, y tampoco se evalúan mediante MATLAB Grader ya que se desarrollan en los sistemas operativos de los robots físicos.

3.1. El reto de la programación del evaluador simbólico

La implementación técnica de los scripts de evaluación (*Assessment Scripts*) en MATLAB Grader para cálculo simbólico presenta una complejidad exponencial frente a la corrección puramente numérica. El evaluador automático debe ser programado de forma exhaustiva para comprobar secuencialmente: el nombre exacto de la variable, la dimensión de la estructura (escalar, vector o matriz multidimensional), la clase de dato estricta (*sym*, *symfun*, *struct*, *table* o *dictionary*) y el contenido algebraico exacto.

Este último punto representa el desafío principal: las operaciones de simplificación matemática en MATLAB (*simplify*, *expand*) pueden arrojar expresiones formalmente distintas, pero algebraicamente equivalentes. La programación docente del evaluador debe contemplar el uso de funciones de validación basadas en la diferencia entre la solución del alumno y la solución patrón, y la posterior verificación de que el rango simbólico resultante es igual a cero, evitando falsos negativos desmotivadores.

3.2. Ejemplo desarrollo de caso

A modo de ejemplo ilustrativo se detalla el desarrollo de las 4 partes de las que se compone cualquier tarea en Matlab Grader, el enunciado, la plantilla de estudiante, la solución de referencia y el evaluador:

3.2.1. Enunciado

Práctica 4. Ejercicio 4F: Cálculo del modelo matemático de orientación girando en torno a ejes móviles X, Y y X respectivamente.

La Figura 1 muestra la representación simbólica de una muñeca Euler con tres articulaciones rotacionales, y en una posición genérica. Los parámetros constantes que caracterizan las dimensiones del robot son: a y b.

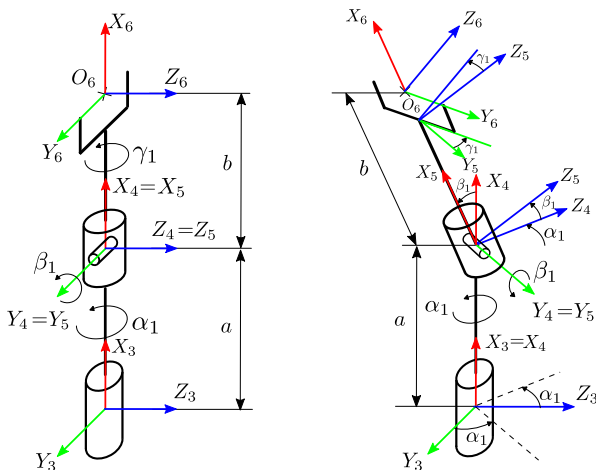


Figura 1: Representación simbólica muñeca de Euler XYZ.

Se desea estudiar los modelos matemáticos matriciales simbólico tridimensionales que describan una orientación de un ángulo $\alpha_1(t)$ en torno al eje X principal, un ángulo $\beta_1(t)$ en torno al eje Y principal, y un ángulo $\gamma_1(t)$ en torno al eje X.

Comprobar el modelo matemático simbólico obtenido en la celda de salida. El alumno realizará las siguientes tareas:

1. Guardar en la variable R34 la función simbólica con la matriz de cambio de orientación

$${}^3R_4(\alpha_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1(t)) & -\sin(\alpha_1(t)) \\ 0 & \sin(\alpha_1(t)) & \cos(\alpha_1(t)) \end{pmatrix}$$

2. Guardar en la variable R45 la función simbólica con la matriz de cambio de orientación

$${}^4R_5(\beta_1) = \begin{pmatrix} \cos(\beta_1(t)) & 0 & \sin(\beta_1(t)) \\ 0 & 1 & 0 \\ -\sin(\beta_1(t)) & 0 & \cos(\beta_1(t)) \end{pmatrix}$$

3. Guardar en la variable R56 la función simbólica con la matriz de cambio de orientación

$${}^5R_6(\gamma_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma_1(t)) & -\sin(\gamma_1(t)) \\ 0 & \sin(\gamma_1(t)) & \cos(\gamma_1(t)) \end{pmatrix}$$

4. Guardar en la variable R36 la función simbólica con la matriz de cambio de orientación global girando en los ejes móviles X, Y y Z (en ese orden)

$${}^3R_6 = \begin{pmatrix} \cos(\beta_1(t)) & \sin(\alpha_1(t))\sin(\gamma_1(t)) & \sin(\beta_1(t))\cos(\gamma_1(t)) \\ \sin(\alpha_1(t))\sin(\beta_1(t)) & \cos(\alpha_1(t))\cos(\gamma_1(t)) - \sin(\alpha_1(t))\cos(\beta_1(t))\sin(\gamma_1(t)) & -\cos(\alpha_1(t))\sin(\gamma_1(t)) - \sin(\alpha_1(t))\cos(\beta_1(t))\cos(\gamma_1(t)) \\ -\cos(\alpha_1(t))\sin(\beta_1(t)) & \sin(\alpha_1(t))\cos(\gamma_1(t)) + \cos(\alpha_1(t))\cos(\beta_1(t))\sin(\gamma_1(t)) & -\sin(\alpha_1(t))\sin(\gamma_1(t)) + \cos(\alpha_1(t))\cos(\beta_1(t))\cos(\gamma_1(t)) \end{pmatrix}$$

5. Guardar en la variable R36_2 la matriz con las nueve componentes simbólicas de R36_2 sustituyendo el símbolo $\alpha_1(t)$ por α_1 , $\beta_1(t)$ por β_1 y $\gamma_1(t)$ por γ_1 .

$${}^3R_6 = \begin{pmatrix} \cos(\beta_1) & \sin(\alpha_1)\sin(\gamma_1) & \sin(\beta_1)\cos(\gamma_1) \\ \sin(\alpha_1)\sin(\beta_1) & \cos(\alpha_1)\cos(\gamma_1) - \sin(\alpha_1)\cos(\beta_1)\sin(\gamma_1) & -\cos(\alpha_1)\sin(\gamma_1) - \sin(\alpha_1)\cos(\beta_1)\cos(\gamma_1) \\ -\cos(\alpha_1)\sin(\beta_1) & \sin(\alpha_1)\cos(\gamma_1) + \cos(\alpha_1)\cos(\beta_1)\sin(\gamma_1) & -\sin(\alpha_1)\sin(\gamma_1) + \cos(\alpha_1)\cos(\beta_1)\cos(\gamma_1) \end{pmatrix}$$

6. Guardar en la variable R36_num la matriz con las nueve componentes numéricas de tipo *double* de R36_2, sustituyendo el símbolo $\alpha_1(t)$ por $\pi/4$, $\beta_1(t)$ por $\pi/3$ y $\gamma_1(t)$ por $\pi/6$.

3.2.2. Plantilla de estudiante

```
syms t alphas(t) betas(t) gammas(t);
assume(alphas(t), 'real');
assume(betas(t), 'real');
assume(gammas(t), 'real');
```

```
% 1. Guardar en la variable R34 la función
simbólica con la matriz de cambio de orientación
girando en el eje X
```

```
% 2. Guardar en la variable R45 la función
simbólica con la matriz de cambio de orientación
girando en el eje Y
```

```
% 3. Guardar en la variable R56 la función
simbólica con la matriz de cambio de orientación
girando en el eje X
```

```
% 4. Guardar en la variable R36 la función
simbólica con la matriz de cambio de orientación
girando en los ejes móviles X, Y y X (en ese orden)
```

```
% 5. Guardar en la variable R36_2 la con las nueve
componentes simbólicas de R35
```

```
% 6. Guardar en la variable R36_num la matriz con
las nueve componentes numéricas de tipo double de
R36_2 sustituyendo alpha1 por pi/4, beta1 por
pi/3 y gamma1 por pi/6
```

3.2.3. Solución de referencia

```
syms t alpha1(t) beta1(t) gamma1(t);
assume(alpha1(t),'real');
assume(beta1(t),'real');
assume(gamma1(t),'real');

% 1. Guardar en la variable R34 la función
simbólica con la matriz de cambio de orientación
girando en el eje X

R34 = [1 0 0; 0 cos(alpha1) -sin(alpha1); 0
sin(alpha1) cos(alpha1)]

% 2. Guardar en la variable R45 la función
simbólica con la matriz de cambio de orientación
girando en el eje Y

R45 = [cos(beta1) 0 sin(beta1); 0 1 0; -
sin(beta1) 0 cos(beta1)]

% 3. Guardar en la variable R56 la función
simbólica con la matriz de cambio de orientación
girando en el eje X

R56 = [1 0 0; 0 cos(gamma1) -sin(gamma1); 0
sin(gamma1) cos(gamma1)]

% 4. Guardar en la variable R36 la función
simbólica con la matriz de cambio de orientación
girando en los ejes móviles X, Y y Z (en ese
orden)

R36 = R34*R45*R56

% 5. Guardar en la variable R36_2 R36_2
sustituyendo el símbolo alpha1(t) por alpha1,
beta1(t) por beta1 y gamma1(t) por gamma1

R36_2 = subs(R36(0),{alpha1(0) beta1(0)
gamma1(0)},{alpha1 "beta1" "gamma1"})

% 6. Guardar en la variable R36_num la matriz con
las nueve componentes numéricas de tipo double de
R36_2 sustituyendo alpha1 por pi/4, beta1 por
pi/3 y gamma1 por pi/6

R36_num = double(subs(R36(0),{alpha1(0)
beta1(0) gamma1(0)},{pi/4 pi/3 pi/6}))
```

3.2.4. Evaluador - Ejemplo Prueba de evaluación

Código de Matlab

```
% Obtener solución de referencia.
R34Reference = referenceVariables.R34;

% Comparar con solución de estudiante.
assessVariableEqual('R34', R34Reference);
```

Comentarios si la prueba es incorrecta:

- Sigue intentándolo. Te recomiendo realizar las siguientes verificaciones:
 - Comprueba el nombre exacto de las variables.
 - Comprueba el tipo/clase (*class*) de las variables.
 - Comprueba la dimensión de las variables.
 - No hay errores sintácticos en las instrucciones.

4. Experiencias, resultados y discusión

El análisis cuantitativo de los datos de interacción con MATLAB Grader proporciona una visión profunda y crítica sobre el comportamiento del alumnado ante sistemas automáticos inflexibles, a lo largo de la asignatura.

4.1. Estudio analítico del progreso en prácticas

La Tabla 1 resume los resultados de participación, éxito y fracaso académico para la muestra total de 22 alumnos a lo largo del itinerario de tareas automatizadas.

Tabla 1: Métricas de seguimiento continuo en la plataforma MATLAB Grader (Curso 2025-26)

Identificador de Tarea / Práctica	Éxito (Aprobado)		Fracaso (Suspense)		Sin Acceder (Abandono)	
Práctica 2: Robot RR Plano	15	68,2%	1	4,5%	6	27,3%
MEDIA GLOBAL PRÁCTICA 2	15	68,2%	1	4,5%	6	27,3%
Práctica 4. Tarea 4.1: Orientación simbólica	14	63,6%	3	13,6%	5	22,7%
Práctica 4. Tarea 4.2: Función orientación	11	50,0%	1	4,5%	10	45,5%
Práctica 4. Tarea 4.3: Función numérica	6	27,3%	5	22,7%	11	50,0%
Práctica 4. Tarea 4.4: Tipos complejos (cell/struct)	6	27,3%	2	9,1%	14	63,6%
Práctica 4. Tareas 4.5, 4.6 y 4.7: Muñeca Euler	6	27,3%	1	4,5%	15	68,2%
MEDIA GLOBAL PRÁCTICA 4	8,6	39,1%	2,4	10,9%	11	50,0%
Práctica 6. Tarea 6.1: Vectores y bases	8	36,4%	3	13,6%	11	50,0%
Práctica 6. Tareas 6.2, 6.3 y 6.4: Pinza RPP	7	31,8%	0,7	3,2%	14,3	65,0%
MEDIA GLOBAL PRÁCTICA 6	7,5	34,1%	1,85	8,4%	12,6	57,5%
Práctica 7. Tarea 7.1: Tabla DH SCARA	7	31,8%	1	4,5%	14	63,6%
Práctica 7. Tarea 7.2 y 7.3: Funciones TH	6,5	29,5%	0,5	2,3%	15	68,2%
Práctica 7. Tareas 7.4 a 7.7: Diccionarios y PCI	4,5	20,5%	0	0,0%	17,5	79,5%
MEDIA GLOBAL PRÁCTICA 7	6	27,3%	0,5	2,3%	15,5	70,5%
Práctica 8. Tarea 8.1: Subjacobiano analítico	4	18,2%	0	0,0%	18	81,8%
Práctica 8. Tareas 8.2 y 8.3: Jacobiano global	3,5	15,9%	1	4,5%	17,5	79,5%
Práctica 8. Tarea 8.4: Velocidades de la pinza	0	0,0%	5	22,7%	17	77,3%
Práctica 8. Tarea 8.5: Aceleración diferencial	4	18,2%	1	4,5%	17	77,3%
MEDIA GLOBAL PRÁCTICA 8	2,85	13,0%	1,75	8,0%	17,4	79,1%
Práctica 9. Tarea 9.1: Interpolador cúbico	5	22,7%	1	4,5%	16	72,7%
Práctica 9. Tarea 9.2: Control multipunto	1	4,5%	3	13,6%	18	81,8%
MEDIA GLOBAL - PRÁCTICA 9	3	13,6%	2	9,1%	17	77,3%
MEDIA TOTAL SEGUIMIENTO CONTINUO	6,4	29,1%	1,7	7,7%	13,9	63,2%

Los datos de la Tabla 1 demuestran un fenómeno de deserción progresiva y drástica a medida que el cuatrimestre avanza. En los estadios iniciales (Práctica 2), el 68.1% de la clase completa la actividad con éxito. Sin embargo, cuando la abstracción matemática del cálculo simbólico aumenta y las estructuras de datos requeridas se vuelven complejas, el volumen de alumnos en estado "Sin acceder" escala severamente.

En la Práctica 7 (Modelado SCARA) la media de alumnos que interactúa cae al 26.3%, y toca un mínimo crítico en la Práctica 8 (Estudio del Jacobiano) donde la media de alumnos con éxito se reduce a tan solo 3 estudiantes (13.6% del total). Un caso singularmente anómalo es la Tarea 8.4 (Función para velocidades de la pinza): se registraron 0 éxitos y 5 fracasos activos, lo que evidenció un pequeño error de sintaxis y tolerancia algebraica en la programación del script evaluador por parte del equipo docente, provocando la imposibilidad técnica de aprobado.

4.2. Impacto directo en las convocatorias de examen final

Para determinar la correlación entre el comportamiento en la plataforma continua y el rendimiento en exámenes presenciales tradicionales sumativos, se analizaron los resultados pormenorizados de cada problema del examen escrito en la convocatoria ordinaria (Tabla 2) y extraordinaria (Tabla 3). Los resultados de ambas tablas se expresan tanto numéricamente (representando el número de alumnos de forma absoluta), como porcentualmente con respecto al total de alumnos de la asignatura.

Tabla 2: Rendimiento académico en examen final (Convocatoria Ordinaria)

Problema examen final (Ordinaria)	Éxito (Aprobado)		Fracaso (Suspense)		Sin Acceder (Abandono)	
P.1.A: Vector de posición y matriz de cambio de orientación	3	13,6%	10	45,5%	9	40,9%
P.1.B: Matrices de transformación homogénea	5	22,7%	7	31,8%	10	45,5%
P.1.C: Matriz antisimétrica y vector de velocidad angular	0	0,0%	11	50,0%	11	50,0%
P.1.D: Vectores velocidad y aceleración lineal	0	0,0%	9	40,9%	13	59,1%
P.2.A: Matriz homogénea de un eslabón con notación Denavit-Hartenberg	0	0,0%	6	27,3%	16	72,7%
P.2.B: Matriz Denavit-Hartenberg intermedias en un robot Scara	1	4,5%	8	36,4%	13	59,1%
P.2.C: Cinemática de velocidad de un robot Scara	0	0,0%	2	9,1%	20	90,9%
P.3.A: Interpolador lineal de grado quinto	0	0,0%	9	40,9%	13	59,1%
MEDIA TOTAL EXAMEN	1,1	5,1%	7,7	35,2%	13	59,7%

Tabla 3: Rendimiento académico en examen final (Convocatoria Extraordinaria)

Problema examen final (Extraordinaria)	Éxito (Aprobado)		Fracaso (Suspense)		Sin Acceder (Abandono)	
P.1.A. Brazo RRR: Transformación Homogénea de Denavit-Hartenberg de un brazo robot RRR	1	4,5%	7	31,8%	14	63,6%
P.1.B. Brazo RRR: Vectores de posición relativos y velocidad lineal de eslabones	0	0,0%	9	40,9%	13	59,1%
P.1.C. Brazo RRR: Matriz de rotación y vector de posición relativo de un eslabón	0	0,0%	2	9,1%	20	90,9%
P.1.D. Brazo RRR: Matrices antisimétricas de los eslabones	2	9,1%	3	13,6%	17	77,3%
P.1.E. Brazo RRR: Vectores velocidad lineal y angular	1	4,5%	2	9,1%	19	86,4%
P.1.F. Brazo RRR: Velocidad lineal y angular del efector terminal	1	4,5%	3	13,6%	18	81,8%
MEDIA TOTAL EXAMEN	0,8	3,8%	4,3	19,7%	17	76,5%

Los resultados de las evaluaciones presenciales reflejan con exactitud el abandono detectado en la plataforma virtual. En la convocatoria ordinaria, problemas esenciales de modelado cinemático como el P.2.A o el cálculo del Jacobiano analítico P.3.A presentaron tasas de respuestas en blanco del 72.7% y 59.0% respectivamente, con un 0% de éxito global. El problema P.2.C (almacenamiento en diccionarios simbólicos) fue omitido por 20 de los 22 alumnos, evidenciando una resistencia estructural insalvable hacia el manejo avanzado de datos complejos.

En las Figuras 2, 3 y 4 se aprecia la evolución porcentual de las tasas de éxito, fracaso y abandono de los alumnos a medida que avanzaban en las prácticas. Existe una correlación lineal directa ($R > 0.85$) entre el número de tareas abandonadas en MATLAB Grader y el número de ejercicios entregados en blanco en el examen final, lo que constata que la herramienta actúa como un predictor inequívoco del fracaso o éxito académico de la asignatura.

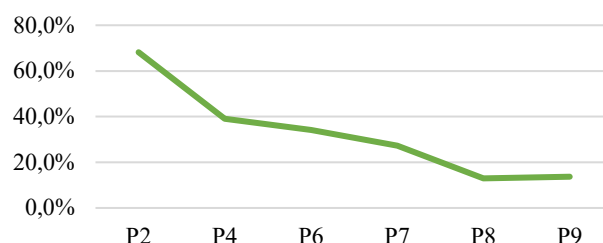


Figura 2: Tasa de éxito.

Por otro lado, se puede apreciar que la tasa de fracaso se mantiene aproximadamente constante y contenida en una banda inferior al 10% lo que denota que la complejidad creciente no es un problema para el alumnado que realiza habitualmente las prácticas.

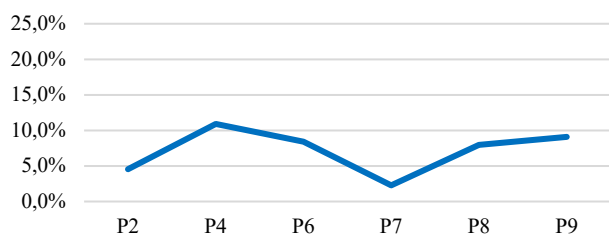


Figura 3: Tasa de fracaso.

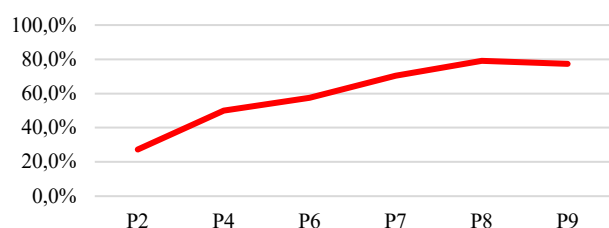


Figura 4: Tasa de abandono.

5. Conclusiones y trabajos futuros

La transición en la utilización de la plataforma MATLAB Grader desde un mero complemento formativo de prácticas, hacia un modelo de evaluación integral de la asignatura en cuestión, permite extraer conclusiones pedagógicas críticas:

La tecnología no supe la falta de persistencia: El *feedback* inmediato e interactivo que proporciona la evaluación automática es una condición necesaria pero no suficiente para mitigar la frustración ante problemas conceptuales complejos.

El "Filtro de Rigurosidad" técnico: La herramienta obliga al alumno a una precisión milimétrica (nombres de variables, clases de datos, dimensiones). El estudiante demuestra una baja tolerancia a esta rigidez algorítmica, optando por el abandono prematuro antes que por la iteración y depuración de su código.

Vulnerabilidad del diseño docente: Los pequeños errores en la programación de los scripts de testeo del profesor (como el detectado en la tarea 8.4) penalizan injustamente al alumno resiliente y destruyen la confianza en la objetividad del sistema.

Curva de aprendizaje hacia el sistema de autoevaluación MATLAB Grader: Dado que es el primer año trabajando en la asignatura con este tipo de formación y evaluación de las prácticas, se prevee que, en futuros cursos, los alumnos estén más adaptados a esta nueva herramienta formativa.

Como líneas de trabajo futuro inmediatas para el curso posterior, el grupo de investigación propone un rediseño de la acción formativa estructurado en la macrofragmentación de tareas (dividir los bloques avanzados en subtarefas todavía más elementales) y la implementación de un itinerario docente dinámico (*scaffolding*) mediante pistas sintácticas automatizadas dentro del *feedback* de error de MATLAB Grader para guiar al alumno sobre si su error es de tipología de dato o puramente algebraico. También se prevee el estudio del uso de la herramienta MATLAB Course Designer, preparada para crear cursos en línea, añadiendo actividades de aprendizaje como texto, vídeos y ejercicios interactivos al actual cuerpo de prácticas para mejorar la comprensión y fomentar la realización de las prácticas por parte del alumnado.

Agradecimientos

Este trabajo ha sido desarrollado en el marco del proyecto PID51 – Programación en Matlab. Autoaprendizaje y autoevaluación, financiado por la Universidad de La Rioja a través de la convocatoria de Proyectos de Innovación Docente (PID) 2025-26 del Vicerrectorado de Ordenación Académica y Profesorado. También ha sido realizado gracias al apoyo de la Universidad de la Rioja a través de la Ayuda a Grupos de investigación REGI 25/69.

Referencias

- Boada, Y., Vignoni, A., 2021. Automated code evaluation of computer programming sessions with MATLAB Grader. 2021 World Engineering Education Forum/Global Engineering Deans Council (WEEF/GEDC), 500–505. DOI: 10.1109/WEEF/GEDC53299.2021.9657355
- Gaona, J., 2020. Panorama sobre los sistemas de evaluación automática en línea en matemáticas. Revista Paradigma Extra 2, 53–81. DOI: 10.37618/PARADIGMA.1011-2251.0.p53-80.id853
- MathWorks, 2025. Using MATLAB Grader for formative feedback. University of Oxford, Centre for Teaching and Learning.
- Sangwin, C. J., 2015. Computer Aided Assessment of Mathematics Using STACK. Selected Regular Lectures from the 12th International Congress on Mathematical Education, 695–713. DOI: 10.1007/978-3-319-17187-6_39
- Terauds, M., et al., 2023. The MATLAB Grader: Expanding Possibilities with Various Task Versions. 2023 IEEE 64th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUON), 1–6. DOI: 10.1109/RTUON60080.2023.10413075
- Vuksanovic, B., et al., 2024. Innovative Application of Matlab Grader in Addressing Plagiarism and Assessing Partially Correct Answers in Engineering Education. Proceedings of the 2024 13th International Conference on Software and Information Engineering (ICSIE '24). DOI: 10.1145/3708635.3708646