

Diseño de actividades auto-evaluables personalizadas en Matlab Grader en el Grado en Ingeniería Telemática*

Design of personalized self-assessment activities in Matlab Grader for the Degree in Telematics Engineering

Antonio Soriano-Asensi¹, Carmen Botella-Mascarell², Sandra Roger³, Jaume Segura-García⁴, Santiago Felici-Castell⁵ y Martin Sanz-Sabater⁶

¹Departament d'Informàtica, ETSE, Universitat de València, antonio.soriano-asensi@uv.es

²Departament d'Informàtica, ETSE, Universitat de València, carmen.botella@uv.es

³Departament d'Informàtica, ETSE, Universitat de València, sandra.roger@uv.es

⁴Departament d'Informàtica, ETSE, Universitat de València, jaume.segura@uv.es

⁵Departament d'Informàtica, ETSE, Universitat de València, santiago.felici@uv.es

⁶Departament d'Informàtica, ETSE, Universitat de València, martin.sanz@uv.es

How to cite: Soriano, A.; Botella, C.; Roger, S.; Segura, J.; Felici, S. y Sanz, M. 2024. Diseño de actividades auto-evaluables personalizadas en Matlab Grader en el Grado en Ingeniería Telemática. En libro de actas: *X Congreso de Innovación Educativa y Docencia en Red*. Valencia, 11 – 12 de julio de 2024. Doi: <http://dx.doi.org/10.4995/INRED2024.2024.18432>.

Abstract

This paper presents an innovation action that focuses on developing customized self-assessment activities using the tool Matlab Grader in the Degree in Telematics Engineering at the University of Valencia. In this way, it is possible to implement an active methodology that benefits both the teaching staff, by allowing them to define personalized activities and sustainable correction, and the students, by receiving instantaneous correction and feedback. The results observed have been a greater involvement of the students during the semester, and a better academic performance in the final evaluation of the subject

Keywords: *Matlab grader; automatic grading and feedback; customized assessment activities*

Resumen

En este trabajo se presenta una acción de innovación que se centra en desarrollar actividades auto-evaluables personalizadas utilizando la herramienta Matlab Grader en el Grado en Ingeniería Telemática de la Universitat de València. De esta forma, se consigue implementar una metodología activa que beneficia tanto al profesorado, al permitirle definir actividades personalizadas y de corrección sostenible, como al alumnado, al recibir corrección y realimentación instantánea. Los resultados observados han sido una

*Este trabajo ha sido financiado por el “Vicerectorat d'Ocupació i Programes Formatius” de la Universitat de València a través del proyecto UV-SFPIE_PIEC-2736754.

mayor implicación de los estudiantes durante el cuatrimestre, y un mejor rendimiento académico en la evaluación final de la asignatura

Keywords: *Matlab grader; corrección y realimentación automática; actividades auto-evaluables personalizadas*

1 Introducción

La propuesta de actividades en clase requiere, por parte del profesorado, preparar la actividad, presentarla a los estudiantes, y tras su realización proporcionar en un tiempo relativamente corto una realimentación significativa para el estudiantado si se quiere que el aprendizaje sea efectivo. Este proceso conlleva una elevada carga de trabajo para el profesorado, especialmente en aquellos grupos con un número de estudiantes relativamente alto.

En este trabajo presentamos la metodología seguida para diseñar actividades de auto-evaluación y personalizadas dentro del laboratorio docente de la asignatura de señales y sistemas lineales (SSL) del Grado en Ingeniería Telemática (GIT), impartido en la Escola Tècnica Superior d'Enginyeria de la Universitat de València. La asignatura SSL del GIT se ubica en segundo curso, primer cuatrimestre, y además, es la primera asignatura de la materia Señales, Sistemas y Servicios de Telecomunicación que cursa el alumnado. El temario comprende una introducción a los sistemas lineales, revisión de la respuesta temporal y en frecuencia de los sistemas lineales, transformadas de Fourier y Laplace y transformada discreta de Fourier. En el curso académico 2023-2024 hay 86 estudiantes matriculados en la asignatura, que se reparten en 5 grupos de laboratorio formados por entre 15 y 20 estudiantes cada uno. Se trata de una asignatura que suele costar un poco más a los estudiantes de segundo curso de GIT. Hay un porcentaje considerable de estudiantes, del orden del 50 %, que requieren varias convocatorias para superar la asignatura. Como consecuencia el número de estudiantes matriculados suele ser elevado, superior a 80 estudiantes, por lo que la realización de actividades de evaluación continua supone una elevada carga de trabajo para el profesorado.

Las prácticas de laboratorio de SSL se realizan con el lenguaje de programación Matlab, y tradicionalmente consisten en la programación de pequeños scripts o en la resolución de ejercicios concretos que sirven de refuerzo al contenido impartido en las sesiones de teoría de la asignatura. En el curso 2021-2022, se planteó la introducción de la herramienta Matlab Grader para poder proporcionar mecanismos de autocorrección al estudiantado, que además proporcionarían realimentación instantánea, y que fueran sostenibles para el profesorado debido al elevado número de estudiantes que cursa la asignatura. Matlab Grader es una herramienta reciente, de manera que la literatura sobre el tema se centra en describir su introducción en distintas asignaturas de estudios de ciencias o ingeniería (Boada & Vignoni, 2021; Martínez Guardiola et al., 2021; Smith P.E., 2020; Tejado et al., 2023). En este curso 2023-2024, se planteó el objetivo de conseguir diseñar actividades de auto-evaluación que fueran personalizadas por cada estudiante, intentando así minimizar acciones de copia entre el estudiantado e intentando mejorar su motivación y su aprendizaje (*learning by doing*). En esta contribución, se presenta la metodología empleada para conseguir obtener actividades personalizadas por estudiante en Matlab Grader, que además, es sostenible para el profesorado puesto que permite una actualización sencilla de las actividades para posteriores cursos académicos. Los resultados obtenidos demuestran un impacto positivo en el aprendizaje del alumnado implicado.

2 Objetivos

El objetivo general de este trabajo es el de diseñar e implementar una metodología activa personalizada para cada estudiante en el laboratorio docente de la asignatura SSL (Grado en Ingeniería Telemática de la Universitat de València). Este objetivo general se puede desglosar en los siguientes objetivos específicos:

- O1. Definir e implementar la metodología para conseguir enunciados personalizados con la herramienta Matlab Grader.
- O2. Proporcionar una solución que permita actualizar las actividades del laboratorio de forma eficiente y sostenible para el profesorado.
- O3. Evaluar el impacto de la metodología en el aprendizaje del estudiantado.

3 Desarrollo de la innovación

SSL es la primera asignatura del bloque de teoría de la señal que se imparte en GIT, y también es la primera de la titulación en la que se utiliza Matlab. En la experiencia presentada en (Botella-Mascarell et al., 2022), se introdujo por primera vez el uso de Matlab Grader en la asignatura, con un uso puntual para reforzar contenidos concretos. Visto el éxito de esa experiencia previa con Matlab Grader en la asignatura de SSL, en esta ocasión se ha dado un paso más con el diseño de actividades personalizadas para cada estudiante, de forma que se fuerza a que cada estudiante necesariamente tenga que resolver la tarea que tiene asignada. Con el fin de minimizar la carga de trabajo del profesorado, se ha automatizado el proceso de generación de las actividades, la evaluación de las mismas y la realimentación que recibe el alumnado. Esta realimentación comprende tanto la evaluación de la actividad como indicaciones para resolverla en caso de que la solución no sea la correcta. Otros autores (Terauds & Smolaninovs, 2023) han propuesto alternativas similares a la presentada en este trabajo, en la que los estudiantes utilizan un script de Matlab previo para generar sus enunciados. Dado que se trata de la primera ocasión en la que utilizan Matlab, se ha estimado más adecuado emplear la alternativa presentada en este trabajo en la que se les proporciona el enunciado en pdf, y además se garantiza que no se asignará el mismo ejercicio a dos estudiantes.

Este apartado se ha organizado en dos subapartados. En el primero se explica el proceso de diseño de cada actividad personalizada. El segundo apartado se centra en cómo se ha implementado la experiencia en el laboratorio de SSL de GIT.

3.1 Diseño de las actividades personalizadas

Uno de los inconvenientes de plantear actividades en grupos de estudiantes numerosos es conseguir que todos ellos realicen la actividad sin sucumbir a la tentación de presentar la solución obtenida por otro compañero. Una forma de favorecer que todos los estudiantes resuelvan la actividad es personalizarla para cada uno de ellos, aunque ello suponga un trabajo adicional para el profesorado. Por otro lado una alternativa posible para reducir la carga de trabajo del profesorado responsable de asignaturas con un elevado número de estudiantes es el uso de actividades auto-evaluables.

En este trabajo se propone el uso de actividades auto-evaluables personalizadas para cada estudiante para favorecer el que cada estudiante se centre en resolver los ejercicios que tiene asignados. Para reducir la carga de trabajo del profesor en el planteamiento de este tipo de actividades se

ha empleado el flujo de trabajo mostrado en la [Figura 1](#) para el diseño de dichas actividades. Dicho flujo de trabajo se basa en dos scripts Python, uno para crear los problemas personalizados para cada estudiante y otro para generar los enunciados de la actividad que se presentará a los estudiantes.

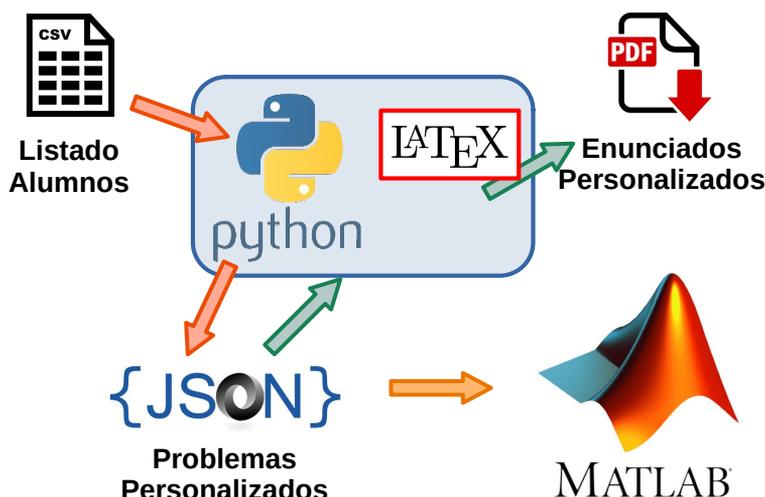


Fig. 1: Flujo de trabajo empleado para la creación de cada actividad.

La funcionalidad del script encargado de generar los problemas se representa en la [Figura 1](#) mediante flechas rojas. Parte de un fichero CSV (Comma Separated Values) descargado de aula virtual con el listado de estudiantes que componen el curso como el mostrado en el [Listado 1](#), en el que se aprecia el nombre, apellidos y dirección de correo electrónico de cada estudiante. La función de este primer script es crear un problema personalizado para cada estudiante y almacenar toda la información necesaria para corregir dicho problema en un fichero JSON (JavaScript Object Notation) como el mostrado en [Listado 2](#). La estructura JSON de cada estudiante está formada por los campos “nom”, “cognoms” que se corresponden con el nombre y apellidos del estudiante, “task” es la información de la tarea asignada a cada estudiante. En el campo “codi” se almacena un código HASH obtenido a partir de la información de la tarea que será el que deberá indicar el estudiante en el momento de presentar su solución. En cada uno de los problemas que pueden formar una tarea se incluye un campo “hash” que se corresponde con un código HASH de la información del problema. Dicho campo se emplea para evitar que se asigna el mismo problema a dos estudiantes. En caso de que el campo “hash” de un problema coincida con el de un problema asignado previamente a otro estudiante se genera un nuevo problema aleatorio.

Listado 1: Ejemplo de fichero *csv* descargado de aula virtual.

```
1 "Nom";"Cognoms";" correo "  
2 "Nombre Primer";"Apellido Alumno";"primalu@alumni.uv.es "  
3 "Nombre Segundo";"Apellido Estudiante";"segest@alumni.uv.es "
```

Listado 2: Ejemplo de fichero *json* de un problema personalizado.

```

1 {
2   "primalu": {
3     "nom": "Nombre Primer",
4     "cognoms": "Apellido Alumno",
5     "task": {
6       "1": {
7         "duration": 24,
8         "T": 4,
9         "nArmonics": 6,
10        "a0": -0.24250538384758702,
11        "an": [
12          -0.3183098861837907,
13          ...
14        ],
15        "bn": [
16          0.0,
17          ...
18        ],
19        "hash": "P848A1930"
20      }
21    },
22    "codi": "K67754BF9"
23  },
24  "segest": {
25    "nom": "Nombre Segundo",
26    "cognoms": "Apellido Estudiante",
27    "task": {

```

El segundo de los scripts Python emplea el fichero JSON, mostrado en [Listado 2](#), con el listado de problemas y genera un documento pdf¹ con los enunciados de todos los problemas asignados a cada estudiante. En la [Figura 2](#) se aprecia un ejemplo del enunciado correspondiente a la tarea mostrada en [Listado 2](#) asignada al primer estudiante. La funcionalidad implementada por este segundo script Python se representa mediante flechas verdes en la [Figura 1](#). El script Python lee la información de cada problema y emplea la herramienta L^AT_EX para generar el documento pdf con los enunciados de las tareas asignadas a cada estudiante.

3.2 Implementación de la innovación en el laboratorio de SSL.

El laboratorio tiene un peso del 30% en la calificación de la asignatura SSL. La mitad de esa calificación se corresponde con la asistencia al laboratorio, en la que se evalúan las actividades planteadas en cada una de las sesiones de laboratorio y que los estudiantes realizan en parejas. Esta calificación de asistencia al laboratorio es una evaluación continua en la que se valora principalmente el trabajo realizado por los estudiantes durante el cuatrimestre. La otra mitad de la calificación del laboratorio se evalúa mediante un examen escrito realizado de forma individual, más orientado a evaluar los conocimientos adquiridos por los estudiantes en relación a los aspectos prácticos de la

¹Portable Document Format

ALUMNO/A: Apellido Alumno, Nombre Primer

Tarea: K67754BF9

Implementa la función `fourierSeries` en Matlab a la que se le pasarán como argumentos el vector de tiempos (t), la señal (s), el periodo fundamental (T) y el número de armónicos que la componen, y proporcionará como resultado la señal reconstruida. Utiliza dicha función para determinar el periodo fundamental y el número de armónicos de todas las señales de esta tarea.

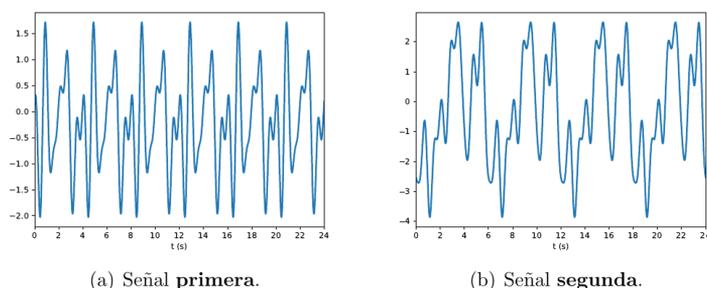


Figure 1: Señales de la tarea **K67754BF9**.

Fig. 2: Ejemplo del enunciado de la tarea **K67754BF9** asignada al estudiante *Nombre Primer Apellido Alumno*.

asignatura. El laboratorio de SSL se organiza en un total de 8 sesiones de 2.5 horas, cuyo contenido se describe en la [Tabla 1](#). De los 5 grupos de laboratorio en que se reparten los 86 estudiantes matriculados en la asignatura solo se han empleado las actividades con Matlab Grader en dos de ellos (grupo de test), manteniendo los tres restantes como grupos de control. Las actividades con Matlab Grader se han introducido como ejercicios adicionales al final de las sesiones 2, 4 y 6. En particular, la actividad de ejemplo presentada en este documento se corresponde con la realizada al final de la práctica 4: Series de Fourier. En la actividad adicional se pide a los estudiantes que utilicen una función, previamente desarrollada durante la sesión de laboratorio, para realizar la descomposición en serie de Fourier del conjunto de señales que cada estudiante tiene asignado. Y que determinen el periodo fundamental y el número de armónicos que componen cada una de ellas. Los últimos 15 minutos de las mencionadas sesiones de laboratorio se dedicaban a explicar a los estudiantes la actividad a realizar, y se les daba la opción de completar dicha actividad hasta el final del día.

Tabla 1: Sesiones de laboratorio y contenido en SSL

Sesión	Contenido
1	Introducción a Matlab.
2	Representación de señales continuas y discretas.
3	Respuesta de un sistema lineal. Convolución.
4	Series de Fourier.
5	Respuesta en frecuencia.
6	Transformada Discreta de Fourier
7	Muestreo y reconstrucción de señales.
8	Respuesta temporal de sistemas continuos.

Implementa la función **fourierSeries** en Matlab que realizará la descomposición en series de Fourier de una señal periódica. La función devolverá la función reconstruida con el número de armónicos que se indique como argumento. Se le pasarán como argumentos:

- el vector de tiempos (t).
- la señal a descomponer (s).
- el período fundamental (T).
- el número de armónicos que componen la señal.

Utilizad la función **fourierSeries** para realizar la descomposición en series de Fourier de todas las señales indicadas en la tarea. Representad en una misma gráfica la señal del enunciado, identificada como "original" y la señal reconstruida, identificada como "Fourier". Para cada una de las cuatro señales, determinad el periodo fundamental y el número de armónicos necesarios para reconstruir la señal original. Representad cada una de las cuatro señales en una figura diferente.

Indicad en la variable **taskCode** el código correspondiente al enunciado que pretendéis resolver. Así, para resolver el primer ejercicio:

```
taskCode = "K0906B66E";
```

Indicad en la variable **usuarioUV** vuestro nombre de usuario de la UV. (Dirección de correo UV sin @alumni.uv.es).

```
usuarioUV= "perca5";
```

En el párrafo destacado anterior se muestran las instrucciones de la actividad y en [Listado 3](#) se muestra el código que se presenta a cada estudiante en la plantilla de Matlab Grader. Todas las líneas que se muestran en [Listado 3](#) están bloqueadas para forzar que los estudiantes sólo

implementen la función **fourierSeries**, y sólo se les permite modificar el periodo fundamental y el número de armónicos de cada una de las señales.

Listado 3: Presentación de la tarea en Matlab Grader.

```
1 taskCode = "K0906B66E";
2 usuarioUV = "usuarioUV";
3
4 jsonFile = "fourierInRed.json";
5
6
7 %% Senal primera
8 [t, s1] = getSignal(usuarioUV, taskCode, 1);
9 T1 = 1; numArmonics1=1;
10 r1 = fourierSeries(t, s1, T1, numArmonics1);
11
12 figure(1)
13 plot(t, s1, "DisplayName", "original");
14 xlim([min(t) max(t)]);
15 xticks(0:2:max(t));
16 title("Senal primera");
17
18
19 %% Senal segunda
20 [t, s2] = getSignal(usuarioUV, taskCode, 2);
21 T2 = 1; numArmonics2=1;
22 r2 = fourierSeries(t, s2, T2, numArmonics2);
23
24 figure(2)
25 plot(t, s2, "DisplayName", "original");
26 xlim([min(t) max(t)]);
27 xticks(0:2:max(t));
28 title("Senal segunda");
29
30
31
32 function r = fourierSeries(t, s, T, numArmonics);
33
34 % Determina el paso temporal
35
36
37 % Selecciona las muestras correspondientes a un periodo T.
38
39
40 % Calcula a0
41
42
43 % Calcula an, bn, y reconstruye la senal con los primeros
  numArmonics
44 r = zeros(size(t));
```

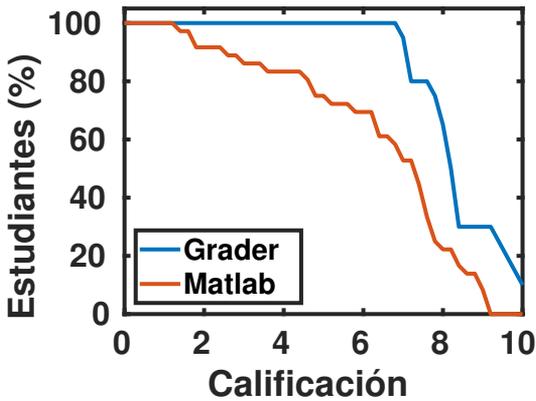


Fig. 3: Comparación de la calificación de asistencia al laboratorio entre los estudiantes que realizaron las actividades adicionales con Grader (Grader) y quienes no (Matlab).

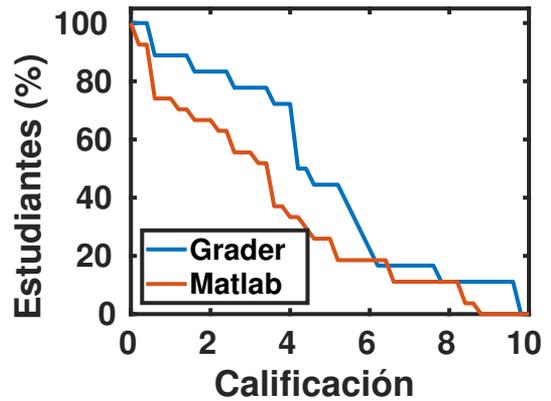


Fig. 4: Comparación de la calificación del examen de laboratorio entre los estudiantes que realizaron las actividades adicionales con Grader (Grader) y quienes no (Matlab).

```
45
46
47 end
```

4 Resultados

Para evaluar el impacto de la innovación en el proceso de aprendizaje del estudiantado se han comparado las calificaciones obtenidas por los alumnos que han realizado las actividades adicionales con Matlab Grader y las calificaciones de los estudiantes del grupo de control. En particular se han comparado las calificaciones de asistencia al laboratorio, Figura 3, del examen de laboratorio, Figura 4 y del examen de teoría, Figura 5. A los estudiantes que realizaron la asignatura en cursos anteriores tienen la posibilidad de no asistir al laboratorio y realizar únicamente el examen de laboratorio. Es por ello que la comparación de calificaciones sólo se han considerado las de los 56 estudiantes que han asistido al laboratorio en el curso 2023-2024. De esos 56 estudiantes, 20 realizaron las actividades con Matlab Grader, mientras que los 36 restantes se han empleado como grupo de control.

Para evaluar el rendimiento de ambos grupos de estudiantes, en la Figura 3 se compara el porcentaje de estudiantes que han obtenido una calificación igual o superior a un determinado valor. Entendiendo que se obtiene un mejor rendimiento si el porcentaje de estudiantes que han conseguido una determinada calificación es mayor. La línea de color azul se corresponde con el rendimiento de los estudiantes que han realizado las actividades con Matlab Grader. El trabajo en el laboratorio se realiza en parejas, la forma de evaluar cada práctica de laboratorio está más ligada al trabajo realizado por los estudiantes en cada práctica, que no a los conocimientos que se han adquirido. De la comparación de las calificaciones de asistencia al laboratorio que se observan en Figura 3, se aprecia que los estudiantes de los grupos de laboratorio que han realizado las actividades personalizadas han obtenido notas superiores a un 6 en la asistencia al laboratorio. Mientras que algunos de los estudiantes del grupo de control han obtenido calificaciones inferiores al 4. Esas bajas calificaciones se deben a una caída de rendimiento de los estudiantes, o en la no asistencia al laboratorio. En el caso de calificaciones inferiores a 5 en la asistencia al laboratorio se debe a que

esa situación se ha producido en varias sesiones de laboratorio. De los resultados observados en [Figura 3](#) se concluye que aquellos grupos en los que se han introducido las actividades personalizadas han mantenido mejor el interés en las prácticas de laboratorio, y se han esforzado más durante el cuatrimestre en realizar las actividades que se les han propuesto en las diferentes prácticas.

Con el fin de evaluar el impacto de la actividad llevada a cabo en los conocimientos prácticos adquiridos por los estudiantes se han comparado las calificaciones obtenidas por los estudiantes en el examen de laboratorio, [Figura 4](#). A diferencia de asistencia al laboratorio, en la que cada profesor evalúa a los estudiantes de su grupo de prácticas, en el caso del examen de laboratorio es el profesor que coordina la asignatura quien ha preparado el examen y quien se ha encargado de la corrección de todos los ejercicios. Para la comparativa sólo se ha considerado aquellos estudiantes que han realizado el laboratorio en el curso 2023-2024 y que también han realizado el examen de laboratorio. De los 20 estudiantes que componían el grupo de test sólo 18 realizaron el examen de laboratorio. En caso del grupo de control 27 estudiantes realizaron el examen de laboratorio, del total de 36 estudiantes que realizaron el laboratorio. De la comparación de las calificaciones en el examen de laboratorio, ver [Figura 4](#), se observa también un mejor rendimiento de los estudiantes de los grupos en los que realizaron las actividades con Matlab Grader. Si bien la diferencia entre ambas curvas es menor que en el caso de la calificación de la asistencia al laboratorio, [Figura 3](#).

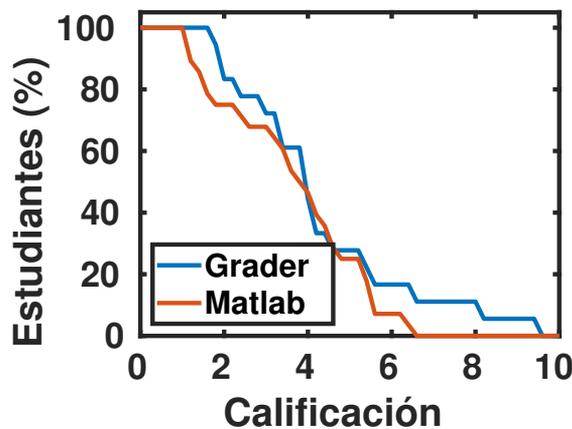


Fig. 5: Comparación de la calificación del examen de teoría entre los estudiantes que realizaron las actividades adicionales con Grader (Grader) y quienes no (Matlab).

Finalmente, en [Figura 5](#) se comparan las calificaciones de los estudiantes en el examen de teoría de la asignatura. En esta ocasión 18 de los 20 estudiantes del grupo de test realizaron el examen de teoría, mientras que en el grupo de control fueron 28 de los 36. En [Figura 5](#) no se aprecia una diferencia significativa entre las calificaciones obtenidas por los estudiantes que realizaron las actividades con Matlab Grader y los estudiantes del grupo de control. Este resultado era previsible debido a que las actividades planteadas tienen un marcado carácter práctico. Es por ello previsible que el impacto de las mismas en el examen de teoría sea menor que en el examen de laboratorio.

5 Conclusiones

En esta contribución se presenta el flujo de trabajo que hemos seguido para el diseño de actividades auto-evaluables personalizadas con Matlab Grader en la asignatura SSL del GIT. Este tipo de actividades reduce considerablemente la carga de trabajo del profesorado y proporciona al estudiantado una realimentación inmediata del trabajo que ha realizado. Además, permite personalizar las tareas asignadas a cada estudiante con lo que se favorece que necesariamente cada uno de ellos tenga que realizar la actividad. El flujo de trabajo presentado se basa en dos scripts de Python, el primero se emplea para generar las tareas personalizadas para cada estudiante, asegurando que no se asigna un mismo problema a dos estudiantes. El segundo de los scripts Python se emplea para generar el documento en formato pdf con los enunciados de las tareas asignadas a cada estudiante.

De los resultados observados se puede concluir que el uso de actividades personalizadas con Matlab Grader contribuye principalmente a mantener el interés de los estudiantes en la asignatura durante el cuatrimestre. Fruto de esa mayor implicación en las actividades de laboratorio favorece que los estudiantes tengan un mejor rendimiento en el examen de laboratorio, de lo que se infiere que contribuyen a mejorar las habilidades prácticas adquiridas por los estudiantes durante el curso.

Referencias bibliográficas

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.

Botella-Mascarell, C., Soriano-Asensi, A., Sanz-Sabater, M., Roger, S., & Segura, J. Metodologías activas de auto-evaluación basadas en la herramienta Matlab Grader en el Grado en Ingeniería Telemática. En: *IN-RED 2022. VIII Congreso de Innovación Educativa y Docencia en Red*. Valencia (Spain). 2022, julio.

Martínez Guardiola, F. J., Alavés Baeza, V., Romero Puig, N., Gimeno Nieves, E., & Francés Monllor, J. (2021). Utilización de Matlab Grader y Matlab Live Scripts para la docencia en asignaturas técnicas de Ingeniería. *Redes de Investigación e Innovación en Docencia Universitaria. Volumen 2021*, 709-722.

Smith P.E., N. (2020). Integration of instructional technology tools including Matlab Grader to enhance learning in a hybrid vibrations course. *2020 ASEE Virtual Annual Conference Content Access*.

Tejado, I., Nuevo-Gallardo, C., de la Encarnación-Sama, R., Pérez, E., & Vinagre, B. (2023). Using Matlab Grader for formative feedback in Engineering degrees. *INTED2023 Proceedings*, 5563-5573. <https://doi.org/10.21125/inted.2023.1456>

Terauds, M., & Smolaninovs, V. (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 (RTUCON)*, 1-6. <https://doi.org/10.1109/RTUCON60080.2023.10413075>