

Creación de prácticas de programación con autocorrección usando la herramienta AUTOCORECTOR

Universidad Politécnica de Madrid

Grupo de Innovación Educativa CyberAula

2022

Autores

Álvaro Alonso González

Enrique Barra Arias

Esta guía ha sido elaborada por el **Grupo de Innovación Educativa CyberAula** en el marco del Proyecto de Innovación Educativa IE22.0903: “Desarrollo y evaluación de un sistema de corrección automatizada para la realización de experiencias de aprendizaje basado en retos en cursos de programación” financiado por la UPM en la convocatoria 2021-22 de “Ayudas a la innovación educativa y a la mejora de la calidad de la enseñanza”.

Madrid, España

1ª Edición. Noviembre 2022.

Índice

1.	Introducción	4
2.	Características de la herramienta AUTOCORECTOR	5
3.	Uso del autocorrector por parte del profesor	7
3.1	Acceso	7
3.2	Creando una tarea autocorregida	9
3.3	Creando el repositorio de GitHub asociado a la tarea	11
3.4	Indicando a los alumnos cómo usar el autocorrector	13
3.5	Ayuda y soporte	14

1. Introducción

Los sistemas de corrección automatizada de tareas de programación se utilizan cada vez más en las enseñanzas universitarias ya que la evaluación manual de este tipo de tareas es muy laboriosa para los profesores, resultando extremadamente costosa en tiempo o incluso inviable en asignaturas con un elevado ratio alumno-profesor. Estos sistemas de corrección automatizada pueden brindar varios beneficios tanto para profesores como para estudiantes. Por un lado, el uso de estos sistemas reduce drásticamente la carga de trabajo de los profesores, permitiendo de este modo la evaluación continua de tareas de programación incluso en cursos con un elevado ratio alumno-profesor. Además, estos sistemas permiten realizar evaluaciones de una forma más objetiva y consistente. Por otro lado, los sistemas de corrección automatizada de tareas de programación permiten a los alumnos obtener información sobre la calidad de sus soluciones y orientación sobre cómo mejorarlas de manera rápida y oportuna. Existen multitud de estudios en la literatura científica que muestran que los sistemas de evaluación automatizada de tareas de programación, cuando son utilizados correctamente, pueden producir impactos positivos en el rendimiento académico de los estudiantes. Según la forma en que se desencadena el proceso de evaluación, los sistemas de corrección automatizada pueden clasificarse en tres tipos: centrados en el instructor (cuando los instructores inician el proceso de evaluación), centrados en el estudiante (cuando los estudiantes inician el proceso de evaluación) e híbridos (cuando el sistema implementa una estrategia dirigida a explotar las fortalezas de los dos enfoques anteriores, buscando lograr el equilibrio entre la provisión de una retroalimentación más inmediata a los estudiantes y un mejor control de la evaluación por parte de los instructores).

Este manual o guía de uso describe un sistema de corrección automatizada de tareas de programación centrado en el estudiante llamado AUTOCORECTOR. El uso de este sistema de corrección automatizada centrado en el estudiante agilizará el proceso de desarrollo de las prácticas por parte de los alumnos haciéndolo a su vez más real, ya que se reducen los tiempos de espera para obtener retroalimentación y se aumenta su motivación a la vez que los hace protagonistas de su propio proceso de aprendizaje.

2. Características de la herramienta AUTOCORECTOR

La herramienta AUTOCORECTOR está disponible de modo totalmente gratuito como paquete npm en la URL <https://www.npmjs.com/package/autocorector>



Ilustración 1 Paquete npm del autocorector

A continuación, se detallan las principales características de esta herramienta autocorector. El autocorector realiza todas las pruebas en el ordenador del alumno y por lo tanto utilizará su mismo entorno de trabajo, evitando diferencias de configuración entre entornos que pueden causar problemas si la corrección fuese realizada en servidor y proporcionando al sistema una alta escalabilidad (ya que no requiere gran cantidad de recursos en un servidor para correr los programas de todos los alumnos y corregirlos). Al realizar cada prueba el sistema proporciona retroalimentación instantánea del resultado de la corrección. Una vez el alumno ha completado la práctica (total o parcialmente) el sistema permite la subida automática de las evidencias de aprendizaje a Moodle y la subsecuente publicación de la nota autogenerada sin necesidad de la intervención del profesor. El sistema permite la subida de la nota tantas veces como el alumno desee hasta que la práctica se cierre en Moodle, haciendo así que el alumno pueda ir progresando paulatinamente en el desarrollo y estando tranquilo por ir teniendo en Moodle notas parciales. Es

importante señalar que como el sistema pasa las pruebas en el ordenador del alumno, este puede verse tentado a modificar el programa de pruebas para facilitar el desarrollo y obtener la máxima nota, pero el sistema previamente a la publicación de la nota, realiza las comprobaciones pertinentes para asegurar que el alumno no ha modificado el programa de corrección automática (sino que ha usado el provisto por el profesor), así como para verificar la identidad del alumno utilizando su token de seguridad de Moodle y la autenticidad de la nota obtenida.

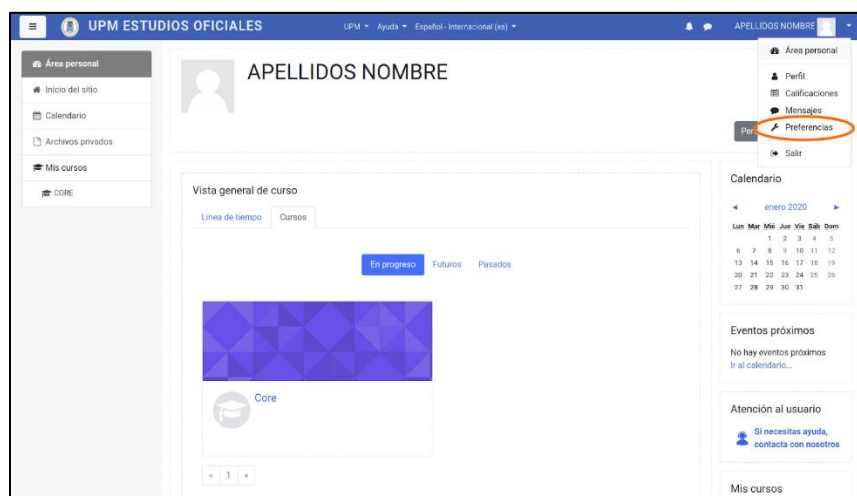
Adicionalmente, el sistema registra los datos generados por los alumnos al hacer uso del mismo, permitiendo calcular analíticas de aprendizaje que permiten a los profesores conocer el tiempo empleado por los alumnos en realizar la práctica, cuántas veces hacen uso del sistema de corrección y la evolución de sus calificaciones.

3. Uso del autocorrector por parte del profesor

3.1 Acceso

Para utilizar el servicio de autocorrección el docente debe acceder a la siguiente dirección mediante un navegador web: <http://aat.dit.upm.es>. Puede emplear cualquier navegador web compatible con HTML5.

Para hacer login en dicha URL debe usar su usuario de Moodle y su token de autenticación (no su contraseña). El token de autenticación se obtiene de la siguiente manera en el Moodle UPM:



UPM ESTUDIOS OFICIALES UPM Ayuda Español - Internacional (es) APELLIDOS NOMBRE

Claves de seguridad

Las claves son empleadas para acceder a su cuenta Moodle desde aplicaciones externas.

Clave	Servicio	Válido hasta	Creador	Operación
e222a37d226e4772ce895529a0d1623c08	Attendance		NOMBRE APELLIDO1 APELLIDO2	Reiniciar
4d2b7a3f375a68f29443dad03712d58f01	Moodle Mobile additional features service		NOMBRE APELLIDO1 APELLIDO2	Reiniciar
ba40327d7abd36c858daa20e6d9f9e4bd1	Moodle mobile web service		NOMBRE APELLIDO1 APELLIDO2	Reiniciar
2242c542dc48e3f2783980adfb65f31902	Entregar Tarea		NOMBRE APELLIDO1 APELLIDO2	Reiniciar
32692562337cf9f9443daa03f97a5763fd	AutoCorrección		NOMBRE APELLIDO1 APELLIDO2	Reiniciar

Ilustración 2 Pasos para obtener el token de autocorrección

Si por alguna situación el profesor pierde su token de Autocorrección o por error lo comparte con los alumnos, este debe ser reiniciado con el botón que aparece en la parte de la derecha de la última pantalla de la Ilustración 2. Esto generará un nuevo token de autenticación e invalidará el antiguo.

Si el acceso funciona correctamente el profesor obtiene una lista de todas sus asignaturas en Moodle. Como ejemplo ver Ilustración 3:

autoCOREctor Salir

Mis asignaturas de Moodle

- Bases de Datos (Bases de Datos_GIB)
- Bases de Datos no Relacionales y Distribuidas (Bases de Datos no Relacionales y Distribuidas)
- Big Data: Fundamentos e Infraestructura (BDFI)
- Ciberseguridad y privacidad: gestión y operación (CPGO / GOCI)
- Core (CORE)
- Gestión de Riesgos y Operaciones en Ciberseguridad (GROC)
- Gestión y operación de la ciberseguridad y privacidad (GOCI)
- Ingeniería Web (IWEB)
- Máster Universitario en Ciberseguridad - General (Máster Universitario en Ciberseguridad)
- Sistemas de Información y Bases de Datos (Sistemas de Información y Bases de Datos)

Ilustración 3 Lista de asignaturas

3.2 Creando una tarea autocorregida

Para poder crear una nueva tarea autocorregida en primer lugar hay que crear la tarea en el Moodle de la asignatura correspondiente.

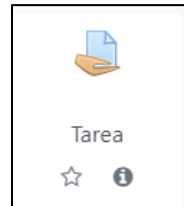


Ilustración 4 Crear tarea en el Moodle

La tarea la tenemos que configurar con entrega de archivos enviados (el autocorrector sube los archivos que ha desarrollado el alumno) y la tenemos que poner oculta. Pero en el desplegable de opciones de la tarea marcamos “hacer disponible”, así quedará no visible para los alumnos pero será una tarea abierta que el autocorrector podrá entregar y el alumno no podrá entrar a hacer entregas manuales.

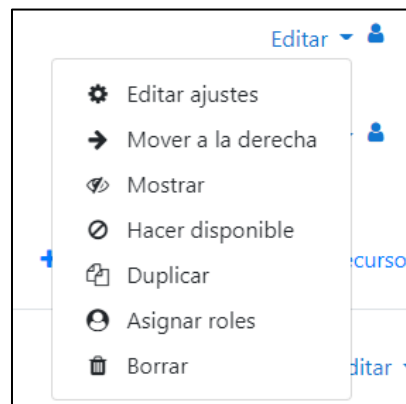


Ilustración 5 hacer disponible la tarea

Una vez hecho esto ya podemos volver a nuestra lista de asignaturas en <https://aat.dit.upm.es/> y hacemos click en la asignatura que queremos. Así accedemos a la página de tareas de dicha asignatura:

Big Data: Fundamentos e Infraestructura (BDFI)

ID	Tarea
12468	Renuncia a la Evaluación continua.
12469	Entrega de la práctica final
12470	Entrega de la práctica final (Enero)
12471	Nota Examen segundo parcial (NoSQL)
12472	Ejercicios de LISP
12473	OPCIONAL : entrega de los ejemplos de LISP en lenguaje CLOJURE
12474	Ejercicio de bases de datos de grafos
12475	Infraestructura como código
12476	Kafka y NIFI
12477	Ejercicios sobre Scala
12478	Tratamiento de datos

Ilustración 6. Lista de tareas de la asignatura

Ahora hacemos click en la tarea que hemos configurado en Moodle y en la que queremos autocorrección. Entonces tendremos que asociar un repositorio de github donde está el enunciado de la tarea y el código para descargarse los test (y que explicaremos más adelante cómo configurar):

Mis asignaturas / BDFI / Ejercicios de LISP

Repositorio de la entrega "Ejercicios de LISP"

Repositorio de GitHub

[Añadir](#)

Ilustración 7 Asociar repositorio de github a la tarea

Una vez pulsado el botón añadir el servicio ofrece varias opciones:

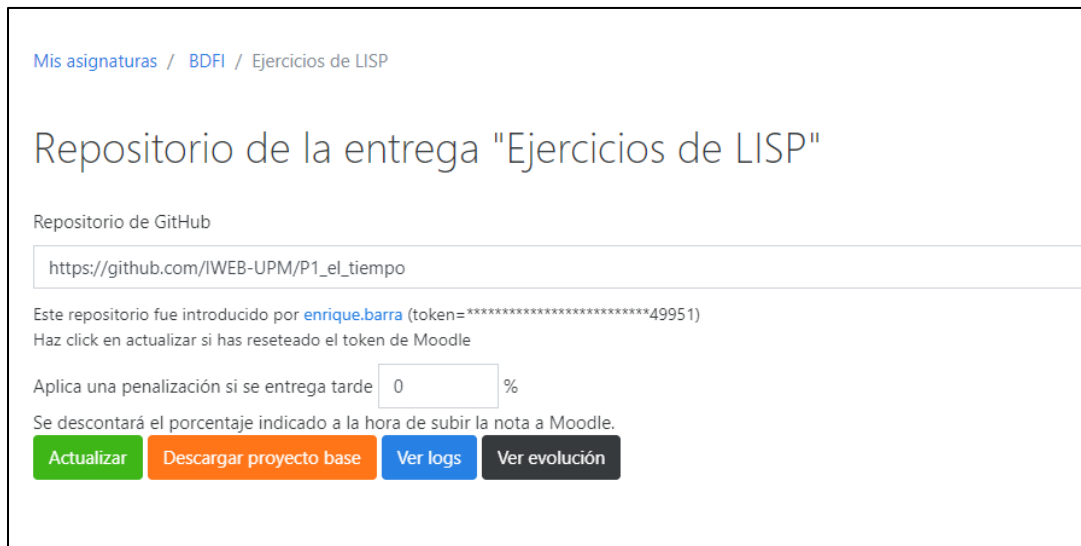


Ilustración 8 Opciones una vez añadido el repositorio de GitHub

Podemos descargar un proyecto base que nos descargará un zip con el package.json y todo bien configurado para subir a github, y ahí solo tendremos que añadir los tests que pasa el autocorrector.

También podemos ver los logs de intentos de entrega y la evolución de notas de los alumnos, siendo estas acciones muy útiles en cuanto a learning analytics.

3.3 Creando el repositorio de GitHub asociado a la tarea

El paso siguiente es crear un repositorio en GitHub vacío y en él subir el zip descargado en el paso anterior. Una vez hecho esto tenemos que empezar a crear los tests que se pasarán en el autocorrector.

Lo más sencillo es basarse en test existentes para ver qué se puede hacer, hay test sobre páginas html que comprueban que existen determinadas etiquetas o determinados ids o clases y en la cantidad en la que aparecen (https://github.com/CORE-UPM/P1_calculadora_HTTP), hay tests sobre funciones o programas que debe desarrollar el alumno (https://github.com/CORE-UPM/P4_Clima), hay tests sobre backend (https://github.com/CORE-UPM/P8_Autores), hay test sobre bases de datos relacionales y no relacionales (<https://github.com/orgs/BBDD-ETSIT/repositories>), hay test sobre el framework de cliente React (<https://github.com/IWEB-UPM>).

Cada uno de ellos es ligeramente diferente al resto con sus particularidades, pero todos plantean en general la misma idea, es una batería de tests sobre el código que ha desarrollado el alumno.

Para entender la estructura del zip que nos descargamos y que debemos subir veamos la práctica de ejemplo de la asignatura IWEB:

https://github.com/IWEB-UPM/P1_el_tiempo

En ella hay una carpeta “autocorrector” que contiene a su vez dos carpetas:

1. “tests” con los tests que se pasa al código del alumno
2. “utils” que contiene utilidades para dejar los tests lo más limpios y entendibles posibles.

Cada uno de los tests debe indicar una configuración inicial como la que sigue:

```

{
    name: "La aplicación tiene un componente Header con el logo y el mensaje de bienvenida con tu nombre",
    score: 1,
    msg_ok: "Header encontrada",
    msg_error: "Header no encontrada o no es como se esperaba, revise el enunciado"
}
    
```

El nombre del test, la nota que da si se pasa, el mensaje si va bien y el mensaje si hay algún error. En su caso además el autocorrector imprimirá el mensaje (stack trace) en caso de que se produzca una excepción.

También se puede ver en el anterior repositorio que al alumno se le puede entregar un esqueleto de código con el hola-mundo de la aplicación que se pide desarrollar para que sepa donde empezar, esto ya queda a decisión del docente que elabora la práctica.

Conviene también ver en las anteriores URLs los enunciados de cada práctica para ver cómo se enuncian este tipo de prácticas y como se detallan los pasos a seguir, ya que muchas veces ocurre que el alumno afirma “en mi pantalla se ve” o “mi función ya da la solución para el caso X” pero resulta que no ha puesto los ids o clases adecuados o no ha considerado todos los casos y el autocorrector no le da los puntos.

3.4 Indicando a los alumnos cómo usar el autocorrector

Una vez desarrollada la batería de tests que pasamos al código del alumno tenemos que preparar la guía para indicar al alumno como autocorregirse las prácticas.

Para ello se proporciona una presentación que puede servir de base personalizándola para la asignatura en concreto. Está disponible en la siguiente URL:

https://docs.google.com/presentation/d/1gnHxxzefHa-ix4m-IV1Z0Y3_Nuom_5pDX9TfbbQI-GA/edit?usp=sharing

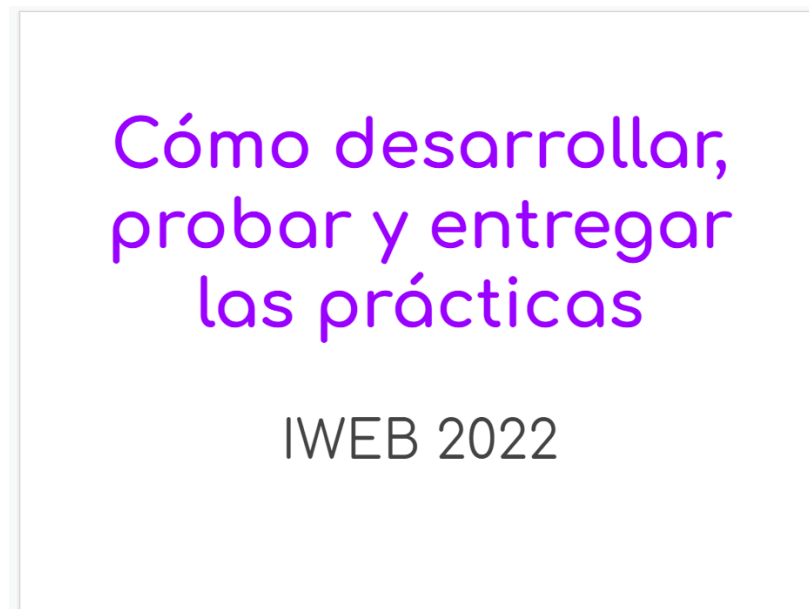


Ilustración 9 - Presentación que se provee a los alumnos en la asignatura

También hay disponibles videotutoriales en el canal de Youtube del Grupo de Internet de Nueva Generación (GING): <https://www.youtube.com/@ging-upm>

Y por último hay una serie de preguntas frecuentes (FAQ) que recomendamos subir a Moodle al comienzo de la asignatura (se hace mediante la opción restaurar, ya que es una actividad de Moodle tradicional):

<https://drive.upm.es/s/iUqcorpQFYs6kLr>

3.5 Ayuda y soporte

La primera vez que se usa el autocorrector es la más complicada. Por ello hemos preparado también varios videotutoriales que se pueden encontrar disponibles en el canal de Youtube del Grupo de Internet de Nueva Generación (GING): <https://www.youtube.com/@ging-upm>

Y también prestamos soporte desde el grupo de innovación educativa CyberAula. Para ello os podéis poner en contacto con enrique.barra@upm.es

Tenemos realizada una publicación en revista indexada donde se puede consultar la arquitectura del autocorrector, todas sus funcionalidades desde el punto de vista técnico y el caso de uso que realizamos durante el COVID para hacer un examen online de programación utilizando esta herramienta. Este artículo está disponible en abierto en:

<https://www.mdpi.com/2071-1050/12/18/7451/htm>

Barra, E.; López-Pernas, S.; Alonso, Á.; Sánchez-Rada, J.F.; Gordillo, A.; Quemada, J. Automated Assessment in Programming Courses: A Case Study during the COVID-19 Era. *Sustainability* **2020**, *12*, 7451. <https://doi.org/10.3390/su12187451>

Agradecimientos

Esta guía ha sido elaborada por el **Grupo de Innovación Educativa CyberAula** en el marco del Proyecto de Innovación Educativa IE22.0903: “Desarrollo y evaluación de un sistema de corrección automatizada para la realización de experiencias de aprendizaje basado en retos en cursos de programación” financiado por la UPM en la convocatoria 2021-22 de “Ayudas a la innovación educativa y a la mejora de la calidad de la enseñanza”. Agradecemos a todos los miembros del Grupo de Innovación Educativa CyberAula su ayuda y colaboración.