



## **Epaile**

### Manual del Profesor

**Arkaitz Garro Elgueta**

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea



# Contenido

<b>Prólogo</b>	<b>5</b>	<b>Bienvenido al <i>Manual del profesor de Epaile</i></b>
<b>Capítulo 1</b>	<b>6</b>	<b>Introducción al módulo Tareas</b>
<b>Capítulo 2</b>	<b>7</b>	<b>Utilización de Epaile</b>
	7	Creación de una nueva tarea
	11	Gestión y revisión de una tarea
	13	Consulta de notas



# Bienvenido al

## *Manual del profesor de Epaile*

En este documento PDF se detallan las instrucciones de utilización de Epaile por parte del profesor.

En este documento se detallan las instrucciones necesarias para la utilización del plugin Epaile en el sistema por parte del profesor. Esto incluye la gestión de tareas y la evaluación manual de las soluciones enviadas por los alumnos.

Si necesita instrucciones detalladas para realizar tareas específicas, o cree que es necesario aclarar algún punto de este manual, no dude en contactar conmigo:

Arkaitz Garro: [arkaitz.garro@gmail.com](mailto:arkaitz.garro@gmail.com)

## Una pequeña introducción al módulo *Tareas*.

En Moodle una *Tarea* es cualquier trabajo, labor o actividad que asignamos a los estudiantes y que no está cubierta por otro módulo de Moodle. Típicamente los alumnos han de devolver el producto de su trabajo como un fichero de ordenador: un documento de texto con un trabajo escrito, una presentación con diapositivas, una imagen gráfica, un video, un programa de ordenador: cualquier cosa susceptible de ser encapsulada como un archivo informático.

También es posible utilizar Moodle para tareas que no tienen un producto informatizable (una actuación artística, una maqueta etc.). En este caso la *Tarea* será un lugar en la página del curso donde se especifican las **instrucciones** para realizar el trabajo asignado y donde, una vez evaluada la tarea consignaremos las **calificaciones** de la misma, de forma que sean accesibles vía el Campus virtual.

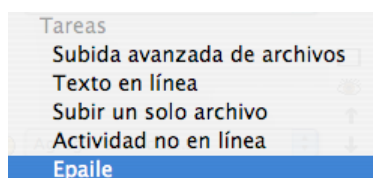
Concretamente para el plugin Epaile, el entregable se corresponde con un programa informático que debe ser evaluado por el propio sistema, ateniéndose a los criterios de evaluación definidos por el profesor. En el siguiente capítulo se muestra la forma de crear este tipo de tareas.

En este capítulo se explica como hacer uso de Epaile, tanto para crear tareas como para evaluarlas manualmente.

El primer paso para el uso de Epaile es la creación de una *Tarea*.

## Creación de una nueva tarea

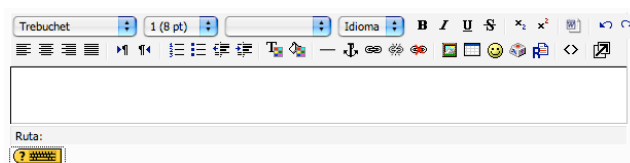
Como para todas las actividades, para crear una nueva *tarea* hay que pasar al modo de edición dentro del curso y seleccionar una *tarea* de tipo *Epaile* en la lista desplegable de la caja *agregar actividad* de un bloque temático. Esto nos llevará al formulario de configuración de la misma, donde completaremos una serie de campos. Los siguientes campos son comunes para todos los tipos de tareas:



- **Título:** es el nombre, corto e identificativo, con el que aparecerá la tarea en la página del curso. Se pueden incluir marcas HTML para lograr efectos de estilo en el texto.

Nombre de la tarea\*

- **Descripción:** es un texto que será presentado a los estudiante cuando entren en la página de la tarea. Este texto debe describir la actividad que deben realizar los alumno, especificando los puntos que deben cubrir y los objetivos de la tarea. Es muy recomendable que indique aquí las precondiciones y poscondiciones que debe cumplir el programa a realizar, así como el formato exacto de su salida.



- **Calificación:** indica la puntuación máxima alcanzable en este actividad o la escala de evaluación aplicable a la misma, si no es numérica. Las escalas de calificación se definen en la sección de Administración del curso.

Calificación 

- **Fecha de inicio:** determina la fecha y la hora a partir de la cual los alumnos pueden comenzar a enviar sus soluciones.

Disponible en      ☐ Deshabilitar

- **Fecha de entrega:** determina sencillamente el día y hora en que acaba el plazo para entregar los archivos a través del Campus virtual. Moodle establece un "periodo de gracia" de 30 minutos tras la hora de cierre en el que continúa recibiendo archivos cuyo envío se iniciara antes del fin del plazo. Esta característica pretende evitar problemas debido al exceso de tráfico en la plataforma cuando un gran número de alumnos se conecta simultáneamente casi al límite del plazo y saturando el tráfico por la red. Esto puede evitar o reducir el número de "quejas" de estilo "le prometo que lo envié antes del cierre, pero el sistema iba muy lento y lo rechazó sin motivo".

Fecha de entrega      ☐ Deshabilitar

- **Impedir envíos retrasados:** permite o deniega el envío de soluciones más allá de la fecha de entrega. Por el momento no influye en la evaluación si la solución es enviada fuera de plazo.

Impedir envíos retrasados



Los siguientes campos definen el comportamiento propio del plugin Epaile:

- **Lenguaje de programación:** indica el lenguaje de programación a utilizar para la realización del ejercicio. Si desea añadir nuevos lenguajes de programación, contacte con su administrador de sistema.

Lenguaje de programación ? Java

- **Fecha Cron:** por defecto este campo se encuentra desactivado, lo que significa que las soluciones de los alumnos serán evaluadas en el momento de ser enviadas. Si desea que todas las soluciones se evalúen en una fecha y hora determinada, desactive el casillero *Deshabilitar* y seleccione una fecha y hora.

Fecha cron ? 3 septiembre 2007 12 15 ☒ Deshabilitar

- **Tiempo máximo de CPU:** tiempo máximo (en segundos) que puede estar en ejecución el programa enviado por el alumno. Muy útil para evitar bucles infinitos.

Tiempo máximo de CPU (segundos) ? 15 secs

- **Máximo uso de memoria:** tamaño máximo de memoria que puede ser asignado al programa en ejecución. Para el caso particular de Java, hay que tener en cuenta el consumo de la máquina virtual. Recomendado 512MB.

Máximo uso de memoria ? 500Kb

- **Reenvío:** indica sencillamente si se permite o no que los alumnos envíen varias versiones del archivo. Si está activada esta opción, usted como profesor podrá ver la última versión enviada por el alumno. En caso contrario sólo se permite un intento de envío y luego se desactiva. Las nuevas versiones pueden recibirse incluso después de que las anteriores hayan sido calificadas. Así pues, por este medio puede establecer un proceso iterativo para que los alumnos revisen y mejoren su trabajo.

Permitir reenvío ? No

- **Alertar a los profesores:** Si se activa, los profesores recibirán una alerta mediante un breve correo siempre que los estudiantes añadan o actualicen el envío de una tarea. Sólo se avisará a los profesores con permiso para calificar ese envío en particular. De este modo, si, por ejemplo, el curso usa grupos separados, entonces los profesores asignados a grupos específicos no recibirán información sobre los estudiantes que pertenecen a otros grupos.

Alertas de email a los profesores ? No

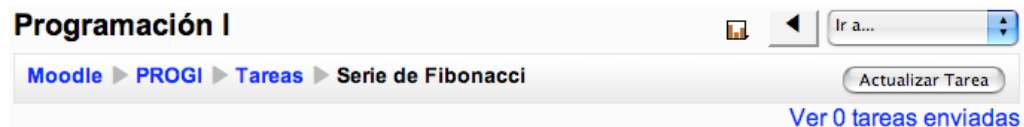
- **Tamaño máximo de fichero:** define el máximo tamaño de fichero que pueden enviar los alumnos como solución.

Tamaño máximo de fichero 2Mb

## Gestión y revisión de una tarea

La gestión de las tareas asignadas es muy simple informáticamente (sin entrar en los contenidos pedagógicos). Cuando usted entre como profesor en la página de la tarea encontrará las instrucciones para su realización. Obviamente, no tiene un interfaz para subir un archivo al servidor pues el profesor no se auto-envía una tarea.

Encontrará el botón de edición en la barra de navegación para reconfigurar la tarea y un enlace, arriba a la derecha, *ver tareas enviadas*, para revisar los trabajos ya enviados por los alumnos.









Siguiendo este enlace llegaremos a la página de revisión de los trabajos. Aquí encontraremos una lista con todos los alumnos del curso. En esa lista mostrará el nombre y apellidos del alumnos, la calificación obtenida, las salidas de compilación y ejecución (si existen), comentarios del profesor y las fechas de modificación de la solución tanto por parte del profesor como del alumno. Además se incluye un enlace a una ventana de calificación y consulta de resultados detallada.

	Nombre / Apellidos	Calificación	Salida compilación	Salida de ejecución	Comentario	Estado
	Arkaitz Carbajo	100 / 100				Actualizar
	Arkaitz Garro	67 / 100	Fibonacci.java:31: ...		Comentarios para ...	Actualizar
	Joserra Lopez	33 / 100				Actualizar
	Juantxo Morcillo	100 / 100				Actualizar
	Ibon Pikabea	-	Fibonacci.java:31: ...		Existen errores de ...	Actualizar
	Iñaki Rodriguez	0 / 100		Error en tiempo de ejecución		Actualizar
	Iñigo Sola	-				Calificación

Esta lista de personas/trabajos se puede ordenar por cualquiera de las columnas existentes en la tabla. En todos los casos se puede ordenar de forma ascendente o descendente haciendo clic en el nombre de las columnas.


También disponemos de la posibilidad de una evaluación rápida, que se puede activar bajo la tabla:

Nombre / Apellidos ↓	Calificación	Comentario	Estado
 Arkaitz Carbajo	100 / 100		Actualizar
 Arkaitz Garro	67 / 100	Comentarios para el alumno.	Actualizar
 Joserra Lopez	33 / 100		Actualizar
 Juanxo Morcillo	100 / 100		Actualizar
 Ibon Pikabea	No hay calificación	Existen errores de compilación no solucionados.	Actualizar
 Iñaki Rodriguez	0 / 100		Actualizar
 Iñigo Sola	No hay calificación		Calificación

Los cuadros de textos para introducir los comentarios al trabajo están pensados para escribir textos relativamente simples. No disponen del editor de texto HTML. Si necesita escribir comentarios más elaborados, con formatos de caracteres y listas o tablas para estructura el texto, puede usar marcas HTML explícitas para obtener estos efectos.

Finalmente, pulsando en *Actualizar* o *Calificación* podemos acceder a la ventana de calificación de la solución enviada por el alumno. En esta ventana se muestran los siguientes datos:

- El **nombre** completo del **alumno**, el fichero enviado (se puede visionar su contenido pulsando en el nombre) y la fecha de envío de la solución.



Arkaitz Garro

lunes, 3 de septiembre de 2007, 17:34 (19 días 23 horas antes de la fecha límite)

Fibonacci.java

- **Errores de compilación**, si se han producido.

Salida compilación

```

Fibonacci.java:31: cannot find symbol
symbol : variable i
location: class Fibonacci
    if (i == 0) System.out.print("Error de compilacion");
        ^
1 error

```

- | <input type="checkbox"/> | Status <input type="checkbox"/> | Expected output <input type="checkbox"/> | Program output <input type="checkbox"/> | Error <input type="checkbox"/> | Score <input type="checkbox"/> |
|--------------------------|---------------------------------|--|---|--------------------------------|--------------------------------|
| Test #1                  | Correcto                        | 1  | 1                                       |                                | 33,33                          |
| Test #2                  | Correcto                        | 1 1 2 3 5                                | 1 1 2 3 5                               |                                | 33,33                          |
| Test #3                  | Respuesta incorrecta.           | 1 1 2 3 5 8                              | 1 1 2 3 5 5                             |                                | 0,00                           |

- Juanan Pereira

lunes, 3 de septiembre de 2007, 18:07





Calificación:


















67 / 100

Trebuchet

1 (8 pt)



Idioma

**B** *I* U ~~S~~  $\times_2$   $\times^2$     

Comentarios para el alumno.

Ruta:

Guardar cambios

Cancelar

Guardar y mostrar siguiente

Siguiente

Gracias a la gestión de calificaciones que incorpora el propio Moodle, podemos mantener organizadas las calificaciones de los alumnos sin ningún tipo de coste adicional para el profesor. Podemos acceder a ellas desde el menú de administración que se encuentra dentro del curso.



Se muestra una tabla que contiene los alumnos inscritos en el curso y los ejercicios existentes para ese curso, mostrando la calificación obtenida por cada alumno en dicho ejercicio. Además, disponemos de las notas medias del curso para cada alumno, para cada ejercicio y la media global de todo el curso.

Programación I <input type="checkbox"/>			
Nombre / Apellidos	Total del curso 	Serie de Fibonacci 	Factorial 
 Arkaitz Garro	33,50	67,00	-
 Iñigo Sola	-	-	-
 Arkaitz Carbajo	50,00	100,00	-
 Juan txo Morcillo	50,00	100,00	-
 Joserra Lopez	16,50	33,00	-
 Iñaki Rodriguez	0,00	0,00	-
 Ibon Pikabea	-	-	-
Promedio	21,43	42,86	0,00

Podemos ordenar las columnas ascendente o descendientemente, además de poder acceder al listado de soluciones de cada ejercicio.

## Formato en el envío de soluciones

A continuación se muestran algunos códigos de ejemplo con el formato que deben cumplir al ser enviados por los alumnos. Estos ejemplos son soluciones para el siguiente problema: dado un entero  $n$  y tantos *strings* como sea definido en  $n$ , escribir por la salida estándar “Hola <nombre>!”.

Entrada	Salida
3	Hola Mundo!
Mundo	Hola Arkaitz!
Arkaitz	Hola Juanan!
Juanan	

## Solución en C

```
#include <stdio.h>

int main()
{
    int ntests, test;
    char name[100];

    scanf("%d\n", &ntests);

    for (test=1; test <= ntests; test++) {
        scanf("%s\n", name);
        printf("Hello %s!\n", name);
    }

    return 0;
}
```

NOTA: los programas escritos en C siempre deben devolver 0 como valor de retorno.

## Solución en C++

```
using namespace std;

#include <iostream>
#include <string>

int main()
{
    int ntests, test;
    string name;

    cin >> ntests;

    for (test=1; test <= ntests; test++) {
        cin >> name;
        cout << "Hello " << name << "!" << endl;
    }

    return 0;
}
```

NOTA: los programas escritos en C++ siempre deben devolver 0 como valor de retorno.

## Solución en Pascal

```
program example(input, output);

var
    ntests, test : integer;
    name : string[100];

begin
    readln(ntests);

    for test := 1 to ntests do
        begin
            readln(name);
            writeln('Hello ', name, '!');
        end;
    end.
```



## Solución en Java

```
import java.io.*;

class Main
{
    public static BufferedReader in;

    public static void main(String[] args) throws
        IOException
    {
        in = new BufferedReader(new
            InputStreamReader(System.in));

        int nTests = Integer.parseInt(in.readLine());

        for (int test=1; test <= nTests; test++) {
            String name = in.readLine();
            System.out.println("Hello "+name+"!");
        }
    }
}
```

Y finalmente, el código para Haskell

## Solución en Haskell

```
import Prelude

main :: IO ()
main = do input <- getContents
        putStr.unlines.solveAll.tail.lines $ input

solveAll :: [String] -> [String]
solveAll = map (\x -> "Hello " ++ x ++ "!" )
```