

# moodleQ: una herramienta para facilitar y extender los cuestionarios de Moodle

José C. Chacón y Miguel Á. Castellanos<sup>1</sup>

**Resumen:** moodleQ es un paquete de R que nació como un Proyecto de Innovación con el objetivo de extender las posibilidades de los cuestionarios de Moodle al crear cuestionarios que ofrecieran a cada estudiante un diferente conjunto de datos y, por tanto, diferentes soluciones, todo ello de forma automatizada. Actualmente tiene dos principales funcionalidades: Por un lado, facilita la construcción de cuestionarios o exámenes simples mediante una aplicación web. Por otro, permite la construcción de cuestionarios complejos con documentos incrustados, o preguntas basadas en bases de datos incrustadas (para descargar y analizar), e individualizadas para cada alumno. La generación automática de las bases de datos y sus soluciones permiten que los estudiantes puedan trabajar autónomamente y descargarse tantos ejemplos como quieran. Estos modelos individualizados se han utilizado en las prácticas de tres asignaturas del área de Metodología de los que ofrecemos algunos resultados.

**Palabras clave:** Moodle; cuestionarios; aprendizaje autónomo; paquete de R.

## 1. Objetivo

El paquete de R moodleQ (Castellanos y Chacón 2021) nace a partir de un Proyecto de Innovación Educativa con un objetivo principal: facilitar a los estudiantes la práctica de actividades relacionadas con el análisis de datos, tanto en clase como de forma autónoma.

Con las herramientas disponibles en Moodle, construir un ejercicio de este tipo requiere generar una base de datos, analizarla y construir cada pregunta y sus soluciones manualmente. El proceso es tedioso, por lo que sólo se pueden construir unos pocos ejemplos y todos los alumnos tienen los mismos datos, lo que reduce su utilidad en actividades que puedan requerir evaluación. Con moodleQ podemos crear un número arbitrario de bases de datos con R, analizarlas, capturar las soluciones programáticamente e integrarlas en un conjunto de preguntas generales. El resultado generado puede importarse en Moodle como un banco de preguntas para construir posteriormente los cuestionarios. Así, cada estudiante recibe una base de datos distinta, asociada con sus soluciones, para un mismo conjunto de preguntas. El que la creación esté automatizada proporciona a los estudiantes la posibilidad de repetir la práctica tantas veces como quiera, siempre con datos y soluciones diferentes. Otra vertiente tratada es la incrustación de documentos que puedan contener información extensa, tablas, gráficos, notación matemática, etc., lo que evita tener que incluirla en las preguntas del cuestionario.

Como subproducto de la construcción de moodleQ también surgió la posibilidad de construir cuestionarios o exámenes simples, generalmente en la modalidad de multi-respuesta, de forma mucho más rápida y eficiente que con el editor de Moodle. La

---

<sup>1</sup> Departamento de Psicobiología y Metodología en CC. del Comportamiento. Email: mcastellanos@psi.ucm.es; jchacon@psi.ucm.es.

actual pandemia ha disparado el uso de los cuestionarios de Moodle, lo que hizo que esta utilidad fuera más solicitada por otros compañeros este último año, aunque la necesidad de instalar (y manejar) R dificultaba su uso. Ello nos llevó a construir una aplicación *web* que no requiere, por tanto, la instalación de R ni conocimientos del mismo.

Comenzaremos presentando la aplicación *web* por ser más simple tanto en uso como en concepción, y continuaremos presentando las características técnicas del paquete moodleQ y su aplicación a la creación de cuestionarios de mayor alcance.

## 2. Cuestionarios multi-respuesta simples mediante una aplicación *web*

Si bien algunas prácticas pueden requerir cuestionarios más sofisticados, en muchos casos simplemente se requiere construir un cuestionario simple, con frecuencia conteniendo un conjunto de preguntas con dos, tres o más alternativas. Ha sido el caso (extensivo) durante esta pandemia, y muchas veces orientado a la realización de exámenes.

Para facilitar este objetivo construimos una aplicación *web* disponible en la dirección <https://moodleq.shinyapps.io/moodshiny> (Figura 1) y que no requiere, por tanto, tener instalados R y los paquetes necesarios ni conocer su funcionamiento. Para usar la aplicación simplemente se necesita un archivo de texto que contenga las preguntas y respuestas y, en el caso de que se trate de un examen donde los errores penalizan, indicar el porcentaje que resta cada error (que ha de ser el mismo para todas las preguntas). Por último, incluimos una opción para aquellos casos en que las preguntas no se pudieran desmarcar una vez marcadas, añadiendo automáticamente una "respuesta" ([Dejar en blanco]) equivalente a haber desmarcado, en tanto no suma ni resta. Es un problema que existió el año pasado en la instalación de Moodle disponible en la UCM y, aunque ha sido resuelto, hemos preferido mantener la posibilidad para otros usuarios que se encuentren con el mismo problema.

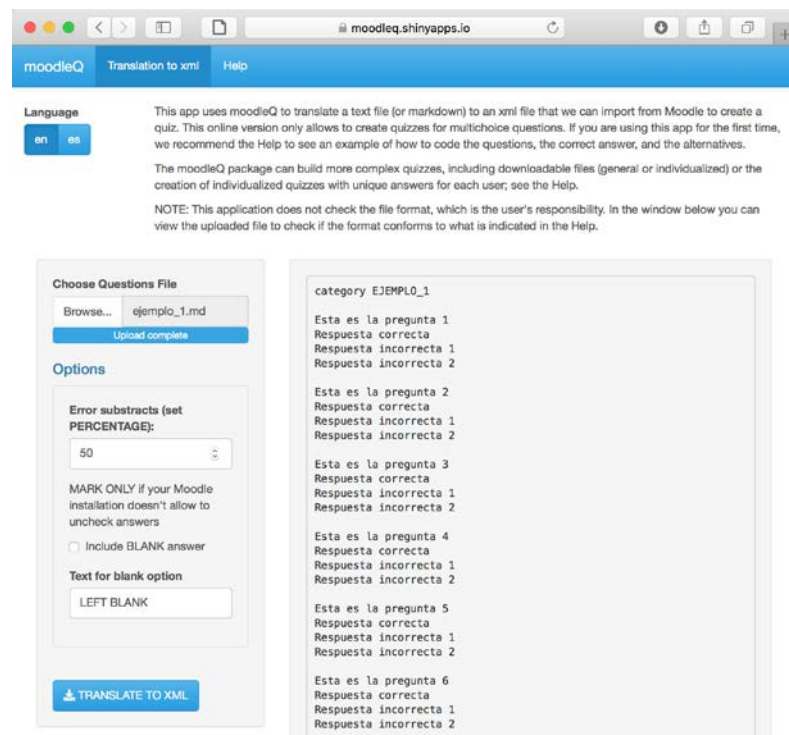


Figura 1. Aplicación *web* para la creación de un banco de preguntas. En el ejemplo se ha leído el archivo ejemplo\_1.md (en formato *markdown*) donde las preguntas se agruparán en la categoría "EJEMPLO\_1" (siempre sin espacios). Cada pregunta comienza con un enunciado y es seguida de la respuesta correcta. En el ejemplo cada error resta un 50% de la puntuación asignada a cada pregunta.

Sobre el archivo de texto es importante dar algunas indicaciones sobre su contenido y estructura para que moodleQ lo interprete adecuadamente. En particular, y como puede verse en la Figura 1, el archivo debe contener: (a) En la primera línea, la palabra clave *category* seguida de la categoría que queramos asignar. Aunque no es obligatoria, se recomienda para que las preguntas queden agrupadas bajo la etiqueta indicada aquí, lo que facilita posteriormente la importación de los bancos de preguntas y en general, la organización. El nombre de la categoría no debe contener espacios. (b) Las *preguntas*, cada una de ellas separada por dos saltos de línea del contenido anterior. (c) las *respuestas*, comenzando siempre por la respuesta correcta y seguida del resto de alternativas.

Por defecto, la aplicación permite utilizar el lenguaje de marcación *markdown*; en cuanto a anotaciones matemáticas en TeX (la sintaxis usada en LaTeX), el resultado depende de la versión de Moodle y de los filtros instalados, y se recomienda leer la información disponible en <https://moodleQ-man.zip> si se quiere utilizar.

Como resultado, la aplicación devuelve un archivo xml que podemos importar desde Moodle para crear un banco de preguntas, como se ve en la Figura 2. Posteriormente bastará crear un cuestionario y añadir todas las preguntas de la categoría EJEMPLO\_1 recién importada.

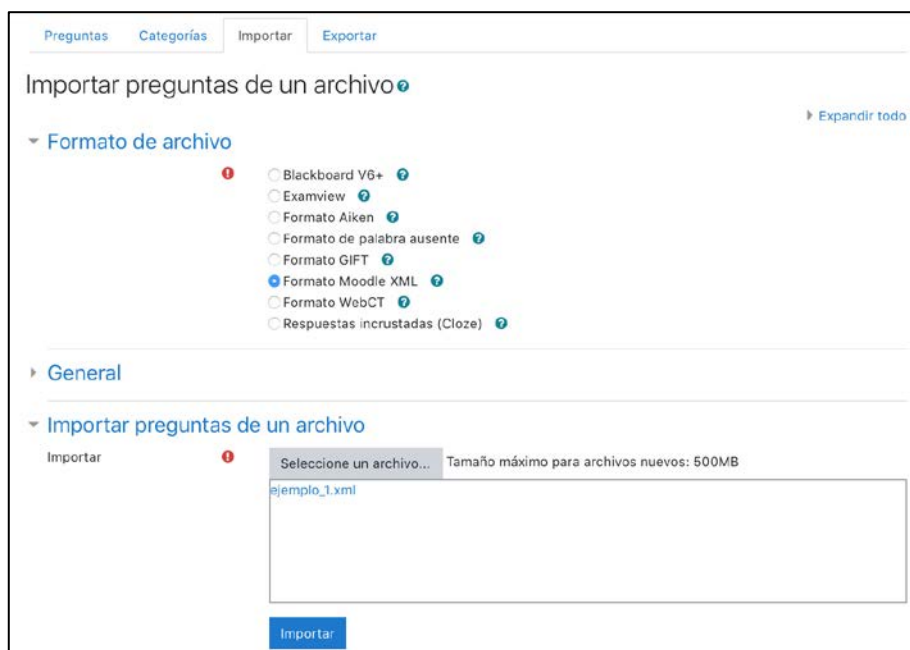


Figura 2. Importación del archivo ejemplo\_1.xml desde Moodle.

### 3. El paquete moodleQ

moodleQ es un paquete de R para la creación de bancos de preguntas de Moodle, desde una perspectiva basada en las clases R6 (Chang 2020) y la programación orientada a objetos. Permite la creación de los principales tipos de preguntas disponibles en Moodle

y las traslada, como hemos visto, a un archivo xml que puede ser importado desde Moodle al banco de preguntas.

El paquete moodleQ está disponible en GitHub (<https://mcstllns/moodleQ>) junto con la documentación y ejemplos en los enlaces (<https://mcstllns.github.io/moodleQ> y <https://moodleQ-man.zip>). La instalación se realiza a través del paquete devtools (Wickham, Hester y Chang 2021) y posteriormente se carga en memoria como se observa a continuación:

```
> install.packages("devtools") # si no está instalado
> devtools::install_github("mcstllns/moodleQ")
> library(moodleQ)
```

A la hora de construir un banco de preguntas, moodleQ se basa en la definición de dos clases R6: moodleQQuestion para las preguntas y moodleQQuiz para el cuestionario (o, más exactamente, para el conjunto de preguntas que constituirán un banco). Una definición mínima se realizaría de la siguiente forma:

```
> question <- R6::R6Class("moodleQQuestion",
  public = list(
    type = NA,
    prop = list())
> quiz <- R6::R6Class("moodleQQuiz",
  public = list(
    quiz = list())
```

Las preguntas se crean invocando al constructor question\$new() y especificando en los argumentos de la función los parámetros que definen una pregunta. Muchos de esos parámetros están definidos por defecto y pueden consultarse en la documentación indicada previamente. En el ejemplo siguiente se crea una pregunta multi-respuesta con tres alternativas donde la primera respuesta es la correcta (puntuía como un 100% del valor de la pregunta).

```
> q1 <- question$new(
  type = "multichoice",
  'name' = "Q001",
  'question' = "Este es el enunciado de la pregunta",
  'answer' = list("Respuesta 1", 100),
  'answer' = list("Respuesta 2"),
  'answer' = list("Respuesta 3"))
> Q <- quiz$new(q1)
> Q$save_xml('mi_cuestionario.xml')
```

Las dos últimas líneas integran la pregunta q1 en el cuestionario Q y graban el resultado en el archivo "mi\_cuestionario.xml", que podremos importar posteriormente desde Moodle.

Los tipos de preguntas disponibles en moodleQ son, según los códigos utilizados en Moodle, *multichoice*, *truefalse*, *shortanswer*, *essay*, *numerical*, *calculated* y *description*, y pueden combinarse en cualquier número al crear un banco de preguntas. Sobre los formatos aceptados, moodleQ trabaja por defecto con HTML, aunque también acepta *markdown* (que es el formato por defecto en la aplicación *web*). Ambos formatos soportan anotaciones matemáticas en TeX, como ya se indicó, aunque con particularidades dependientes de la instalación de Moodle (véase la ayuda del paquete).

No obstante, esta forma de crear bancos de preguntas no es particularmente mejor que usar el editor de Moodle. Su verdadera utilidad se manifiesta cuando necesitamos cuestionarios más complejos.

## 4. Construcción de cuestionarios complejos

Las preguntas pueden definirse completamente mediante `question$new()`, o sólo inicialmente, de forma que la definición se pueda completar mediante otras funciones. Ello permite mucha más flexibilidad y la posibilidad de añadir interesantes funcionalidades.

### 4.1. Incrustación de archivos

Es posible insertar documentos en los enunciados de las preguntas, habitualmente en una primera pregunta de tipo *description*, pero también en otras. Para ello se puede utilizar la propiedad *file* en el constructor o directamente el método *addfile()* tras haber creado la pregunta. En el ejemplo se insertan dos archivos en el enunciado de una pregunta de tipo *multichoice*, uno ya existente (A.csv) y otro que es creado junto con la pregunta (B.csv).

```
> q1 <- question$new(
  type="multichoice",
  'name' = "P001",
  'question' = 'Choose:
    <a href="@@PLUGINFILE@@/A.csv">download A.csv</a>
    <a href="@@PLUGINFILE@@/B.csv">download B.csv</a>',
  'answer' = list("Answer 1", 100),
  'answer' = list("Answer 2"))
> q1$addfile("./file.csv", "A.csv")
> df <- data.frame(a = rnorm(10),
  b = rnorm(10))
> tmp <- tempfile()
> write.table(df, file = tmp, sep = ",", dec = ".")
> q1$addfile(tmp, "B.csv")
> unlink(tmp)
```

Los detalles del código para la inserción de archivos pueden encontrarse en la documentación de ayuda del paquete.

### 4.2. Creación de bases de datos y soluciones individualizadas

Las preguntas de tipo *calculated* permiten trabajar con bancos de respuestas. Estas son preguntas que aceptan *wildcards* o *tokens* que pueden ser utilizados tanto en el enunciado como en las alternativas de respuesta, y funcionan como variables o identificadores que aceptan distintos valores. Podemos ver su uso en el siguiente ejemplo, con los *tokens* {A} y {B}:

```
> q1 <- question$new(
  type="calculated",
  'name' = "P001",
  'question' = "
    Con los números: A = {A} y B = {B}
    <p>Calcula: A + B</p> ",
  'answer' = list("{C}", 100),
  'dataset' = list(data.frame(A = c(1, 4))),
```

```
'dataset' = list(data.frame(B = c(3, 2))),  
'dataset' = list(data.frame(C = c(4, 6)))
```

Moodle aleatoriamente presentará una de las 2 filas disponibles y utilizará los valores de A, B y C (por ejemplo, 1, 3 y 4) para la pregunta, de forma que al contestar con el número 4 la respuesta es considerada correcta. Además moodle permite sincronizar los bancos de distintas preguntas con la opción *synchronize* (activa por defecto en moodleQ), esto es: una vez se asigna una fila a un usuario, se asignará esa misma fila en todas las preguntas. Esto permite que varias preguntas (y sus respuestas) se puedan enlazar a un mismo conjunto de datos, lo que podemos usar para ampliar las posibilidades.

En la documentación de moodleQ podemos encontrar un ejemplo (el tercero, demasiado extenso para incluirlo aquí) que incluye la generación de una lista que contiene 10 bases de datos sobre las que se calculan varios valores de interés. A partir de ahí se generan 10 archivos que son incrustados en una pregunta (uno por usuario) y un conjunto de respuestas vinculadas a los datos (y enlazadas para cada usuario) que toman los valores válidos de los cálculos previos.

Este procedimiento abre nuevas posibilidades pero también tiene sus limitaciones, que conviene conocer. Por un lado, sólo las preguntas de tipo *calculated* permiten asociar una entrada de la base de datos con un usuario; pueden transformarse en preguntas de elección múltiple, pero a través de un pequeño truco (que puede verse en la ayuda). Además, las variables sólo pueden ser de tipo numérico, y sólo se pueden utilizar los *tokens* que han sido incluidos en la misma pregunta.

Con este procedimiento se puede generar cualquier número de bases de datos para que los estudiantes puedan practicar cuantas veces quieran, ya que cada vez que vuelven al cuestionario se les asigna un nuevo conjunto de datos.

## 5. Aplicación en las prácticas de las asignaturas

Desde la primera versión de moodleQ, éste ha sido utilizado en las prácticas de varias asignaturas del área de Metodología (Estadística I y II y Psicometría). En todos los casos se ha seguido una estructura similar a la mostrada aquí en los ejemplos: un enunciado más o menos extenso planteando el problema a resolver (bien en una pregunta de tipo *description* o en un PDF incrustado), un archivo de datos incrustado que cada estudiante ha de descargar, y una serie de cuestiones a resolver que requieren realizar los análisis pertinentes e indicar los resultados.

Consultados mediante un cuestionario *online*, los alumnos de este curso que han realizado este tipo de prácticas valoraron algunas afirmaciones acerca de su utilidad. Los resultados pueden verse en la Tabla 1.

Tabla 1. Grado de acuerdo de los estudiantes con las siguientes afirmaciones (N = 240)

Pregunta	M (DT)
Las prácticas que hice durante el curso me ayudaron a comprender	5,67 (1,58)

los contenidos teóricos	
Poder repetir los análisis con nuevos datos me ha ayudado a mejorar el rendimiento y la comprensión de la asignatura	5,89 (1,45)
En las prácticas, al tener que resolver un problema creado para mí sentí una mayor implicación con los contenidos teóricos del curso	5,25 (1,65)
Los Cuestionarios de Moodle son, en general, una buena herramienta a utilizar en estas asignaturas	5,92 (1,35)

Notas: 1. M representa la media; DT la desviación típica. 2. En todas las preguntas se utilizó una escala de 1 (totalmente en desacuerdo) a 7 (totalmente de acuerdo).

Si bien la valoración general es alta, se observa que los valores mayores se encuentran en la valoración general de los cuestionarios de Moodle como herramienta de utilidad y específicamente en la posibilidad de repetir los ejercicios con nuevos datos.

## 6. Prospectiva

Actualmente, el paquete moodleQ cubre perfectamente las necesidades para las que se construyó, por lo que seguiremos usándolo en las prácticas de estas asignaturas como hasta ahora.

Otra cuestión es la aplicación *web*, cuya funcionalidad es muy limitada y que queremos ampliar en un futuro, para poder crear cuestionarios más complejos sin necesidad de instalar (y conocer) R y los paquetes necesarios para su uso.

## Referencias bibliográficas

Castellanos, Miguel A. y José C. Chacón. 2021. *moodleQ: Creación de preguntas en xml moodle*. R package version 0.5.3. <https://mcstllns.github.io/moodleQ/>

Chang, Winston. 2020. *R6: Encapsulated Classes with Reference Semantics*. R package version 2.5.0. <https://CRAN.R-project.org/package=R6>

Wickham, Hadley, Jim Hester y Winston Chang. 2021. *devtools: Tools to Make Developing R Packages Easier*. R package version 2.4.2. <https://CRAN.R-project.org/package=devtools>