

Guía del desarrollador de AssessMediaWiki

José Tomás Tocino García - `josetomas.tocino@uca.es`

Resumen

Este documento tiene como objetivo familiarizar al desarrollador con AssessMediaWiki (AMW), sus funcionalidades y la manera en la que ha sido desarrollado, de forma que le sea más fácil entender y modificar el código fuente.

1. Introducción a AssessMediaWiki

AssessMediaWiki - <http://assessmediawiki.forja.rediris.es/> es una aplicación web de código abierto capaz de conectarse a una instalación de MediaWiki y proporcionar una serie de herramientas para la evaluación de las aportaciones de los usuarios, ya sea en forma de autoevaluación, heteroevaluación o evaluación entre iguales. Todas esas evaluaciones se mantienen registradas y es posible supervisarlas.

1.1. Introducción al trabajo colaborativo en wikis

Un wiki es un entorno online de **trabajo colaborativo** en el que multitud de usuarios pueden **añadir y editar artículos** sobre una temática determinada, utilizando normalmente un navegador web como punto de acceso. El wiki más popular es **Wikipedia**¹, una enciclopedia online de libre acceso y edición.

Los wikis están compuestos de **artículos en formato texto** que pueden incluir a su vez elementos multimedia, como imágenes. Por defecto cualquier usuario puede crear y editar un artículo, aunque es posible limitar estas capacidades según diferentes criterios. Cada edición de un artículo se conoce como una **revisión** y es almacenada en el sistema, junto al resto de revisiones anteriores. Esto supone una potente herramienta para poder observar la **evolución de un artículo** desde su concepción.

Además, los artículos afines pueden **clasificarse en categorías**, enlazarse entre ellos y muchas otras opciones. El sistema de gestión de wikis más famoso y en el que se basa AssessMediaWiki es **MediaWiki**². Nació como base para la enciclopedia Wikipedia y es un proyecto de código abierto.

En el ámbito educativo se están utilizando los wikis como **herramienta de evaluación** de los conocimientos y el trabajo continuo de los alumnos. El problema surge por la falta de herramientas de análisis del contenido de los wikis, tanto de forma cuantitativa como cualitativa. En la Universidad de Cádiz se han desarrollado numerosos proyectos para satisfacer esta demanda, como **StatMediaWiki**³, que hace análisis estadístico de las aportaciones en un wiki, o el presente AssessMediaWiki.

1.2. Funcionalidades principales

1.2.1. Evaluación básica

La principal funcionalidad de AssessMediaWiki es permitir a los usuarios **evaluar revisiones concretas de los artículos de un wiki**, a través de la pestaña **Assess**. No se evalúa el estado final de cada artículo, sino las modificaciones particulares que cada usuario ha llevado a cabo. Cuando un usuario accede a AMW, se le presenta un enlace a la revisión del artículo que debe evaluar, y una lista de conceptos evaluables. Una vez lista la evaluación, se envía y queda registrada en el sistema.

¹<http://wikipedia.org>

²<http://mediawiki.org>

³<http://statmediawiki.forja.rediris.es/>

1.2.2. Conceptos evaluables

Los conceptos que el usuario tiene que tener en cuenta a la hora de evaluar se representan como **entregables**, con un título y una descripción, como por ejemplo “*Ortografía*”. Se les puede asignar una calificación numérica y un comentario textual a uno o varios de estos conceptos. El profesor puede, desde la propia aplicación, añadir, editar y borrar estos parámetros de evaluación. Esto se hace a través de la pestaña **Params**.

1.2.3. Revisión de evaluaciones

Los usuarios del wiki pueden ver las evaluaciones que otros usuarios han hecho sobre sus aportaciones mediante la pestaña **My Assessments**. Desde el panel es posible ver los detalles de cada evaluación y, si el usuario no está de acuerdo con la misma, es posible añadir una réplica a través del botón **Reply**, que añadiría una nueva evaluación asociada a la anterior.

Cabe notar que los alumnos no pueden ver quiénes han sido los autores de las evaluaciones para evitar posibles conflictos sociales. Sin embargo, el administrador sí tiene esta capacidad.

1.2.4. Capacidades de administración

AssessMediaWiki implementa dos roles de usuario distintos: supervisores y estudiantes. Los supervisores tienen ciertas capacidades administrativas con las que los alumnos no cuentan, como son:

- Edición de los conceptos evaluables y la categoría de artículos a evaluar (pestaña *Params*).
- Ver la lista de todos los alumnos y las evaluaciones de sus revisiones (pestaña *Students*).
- Ver información oculta, como los autores de las evaluaciones.

1.2.5. Financiación

Este trabajo ha sido parcialmente financiado por el Proyecto de Innovación Docente “**La Heteroevaluación como Apoyo a la Sostenibilidad en Evaluaciones Complejas de Trabajos Colaborativos en Wikis**” (Código PI2.12.029) de la Universidad de Cádiz

2. Instalación

La instalación de AssessMediaWiki parte de una **instalación previa de MediaWiki**. Es importante que el usuario conozca los datos de acceso a la base de datos de MediaWiki. Para más información sobre cómo instalar MediaWiki u obtener las credenciales de la base de datos, visite la web de MediaWiki⁴.

⁴<http://www.mediawiki.org/wiki/MediaWiki>

El primer paso es crear una base de datos para AMW, y un usuario que tendrá todos los privilegios sobre ella. Se reflejarán los pasos a seguir desde la línea de comandos de un sistema GNU/Linux, aunque el procedimiento también puede hacerse de manera visual mediante asistentes como *PhpMyAdmin*. Para acceder a la terminal de mysql utilizaremos:

```
% mysql -u root -p
```

Una vez dentro, ejecutaremos el siguiente comando para crear la base de datos `datos_amw`.

```
CREATE DATABASE datos_amw;
```

Seguidamente, crearemos el usuario `'usuario_amw'` y le daremos permisos sobre la base de datos.

```
GRANT ALL ON datos_amw.* TO usuario_amw@localhost IDENTIFIED BY 'clave_amw';
```

Hecho esto, iremos al fichero `application/config/database.php` y cambiaremos los parámetros de acceso, de forma que coincidan con los que hemos utilizado:

```
$db['default']['username'] = 'usuario_amw';  
$db['default']['password'] = 'clave_amw';  
$db['default']['database'] = 'datos_amw';
```

El siguiente paso es configurar el acceso a la base de datos de MediaWiki. Para ello, editamos el fichero `application/config/amw.php` con los datos de acceso:

```
$config["database_mw"] = "bd_mw";  
$config["username_mw"] = "usuario_mw";  
$config["password_mw"] = "clave_mw";
```

El siguiente paso es indicar qué usuarios tendrán permisos de administrador. Por regla general, los profesores de la asignatura y el desarrollador deberían tener acceso de administrador. Previamente es necesario conocer los ID numéricos de cada uno de los usuarios administradores, que podremos encontrar en la tabla `users` de MediaWiki. Seguidamente, iremos al fichero `application/config/amw.php` e incluiremos en la clave `usuarios_admin` los ID de los usuarios administradores:

```
// Administradores los usuarios con ID = 1 y ID = 2  
$config["usuarios_admin"] = array(1, 2);
```

En AMW solo se tendrán en cuenta las ediciones realizadas en un cierto periodo de tiempo. Éste se definirá indicando la fecha de inicio y la fecha de fin en el fichero de configuración previamente indicado:

```
$config["fecha_inicio"] = "20120301000000";  
$config["fecha_fin"]    = "20120605000000";
```

Con esto ya podremos acceder a la aplicación, para lo que habrá que usar las credenciales de acceso al wiki. Inicialmente AMW se encontrará en *modo desarrollo*, de forma que al hacer login no hará falta

meter la contraseña exacta del usuario con el que queramos acceder, cualquier contraseña valdrá. Esto nos permitirá probar los diferentes roles de usuario a la hora de hacer pruebas.

ES INDISPENSABLE que al poner la aplicación a disposición de los usuarios se desactive el modo desarrollo. Para ello, accederemos al fichero `application/config/amw.php` y pondremos el valor a `FALSE`.

```
$config["modo_desarrollo"] = FALSE;
```

3. Información de desarrollo

3.1. Introducción

AssessMediaWiki está desarrollado en el lenguaje de programación **PHP**⁵ y utilizando el framework **CodeIgniter**⁶. Se trata de un framework de desarrollo web que utiliza el patrón **Modelo-Vista-Controlador**⁷, que divide las aplicaciones web en tres partes, de forma que el mantenimiento y las actualizaciones posteriores sean más sencillas:

- Los **modelos** representan los datos de la aplicación. Por regla general un modelo corresponde a una entidad (por ejemplo, un *usuario*), y se le suele asignar una tabla en la base de datos. Los modelos también albergan métodos para el trabajo con los datos, abstrayendo los pormenores del acceso al usuario.
- Las **vistas** son las formas en las que se presentan los datos al usuario. En la mayoría de los casos, las vistas son plantillas web hechas en HTML y CSS, pero también es posible representar los datos utilizando vistas en otros formatos, como XML o JSON.
- Los **controladores** se sitúan entre los modelos y las vistas, y ejecutan la lógica de la aplicación. Reciben las peticiones del navegador, deciden qué operaciones deben realizar y qué datos necesitan, y cargan las vistas apropiados con los datos necesarios. La mayor parte de la programación se ubica en los controladores.

3.2. Orígenes de datos

Para poder trabajar, AssessMediaWiki hace uso de dos orígenes de datos. En primer lugar, la aplicación tiene su **propia base de datos**, en la que se guardan las evaluaciones de las aportaciones del wiki, las revisiones de las evaluaciones y los comentarios, así como diversos parámetros de evaluación.

⁵<http://www.php.net>

⁶<http://codeigniter.com>

⁷http://es.wikipedia.org/wiki/Modelo_vista_controlador

Por otro lado, la aplicación debe tener acceso a la **base de datos del wiki** sobre el que se van a hacer las evaluaciones. Además, toda la autenticación se hace sobre la base de usuarios del wiki, por lo que evitamos que haya que hacer dos registros.

3.3. Modelos

Se detallan ahora los diferentes modelos de datos con los que cuenta la aplicación para la gestión de la información.

3.3.1. Acceso

El modelo *Acceso* (`application/models/acceso_model.php`) se encarga de establecer la conexión a la base de datos de MediaWiki. Al utilizar este modelo, automáticamente establecerá una conexión con la base de datos MySQL de MediaWiki que se haya configurado en la aplicación (en el fichero `application/config/amw.php`).

Otra de las tareas de este modelo es obtener la información de login a la aplicación, haciendo uso de la tabla de usuarios de MediaWiki. El **controlador de acceso** hará uso de estas capacidades para gestionar el login.

3.3.2. Category

AssessMediaWiki trabaja sobre un subconjunto de los artículos presentes en el wiki MediaWiki. En particular, la revisión se produce sobre los artículos de una categoría en particular, que el administrador puede configurar desde la pestaña *Params* de la aplicación.

El modelo *Category* (`application/models/category_model.php`) lee y gestiona esta categoría, que se encuentra almacenada en la BD de AMW como un registro de la tabla `config` cuyo campo `parameter` tiene el valor `category`.

3.3.3. CSV

Un fichero CSV (*Comma separated values*) es un fichero de texto plano con datos tabulados mediante comas u otros caracteres que se utiliza, normalmente, para el análisis en masa de datos con fines estadísticos o de automatización.

En AMW, el modelo *CSV* (`application/models/csv_model.php`) ofrece funciones que procesan y devuelven datos sobre las evaluaciones de los usuarios, de forma que el controlador de *Feedback* pueda generar listados CSV para descargar.

3.3.4. Entregable

A la hora de evaluar una edición en un artículo se definen una serie de **conceptos** a evaluar. Cada uno de estos conceptos es representado como un *entregable*, con un título y una descripción.

El modelo *Entregable* (`application/models/entregable_model.php`) se encarga de leer los entregables establecidos en la base de datos.

3.3.5. Evaluación

Cuando un usuario hace una evaluación sobre una revisión de un artículo, por cada uno de los conceptos evaluables que ha definido el profesor (representados por el modelo *Entregable*) el usuario hace una evaluación numérica y tiene la posibilidad de dejar un comentario escrito. Para representar esta información en la base de datos existen dos tablas.

En primer lugar, la tabla **evaluaciones** indica quién está haciendo la evaluación, qué revisión se está evaluando y cuál fue su autor. Seguidamente, la tabla **evaluaciones.entregables** asocia cada una de esas evaluaciones a los conceptos que han sido evaluados junto a la nota numérica y el comentario que el evaluador puede haber añadido.

El modelo *Evaluaciones* (`application/models/evaluaciones_model.php`) se encarga de gestionar toda esta información en la base de datos.

- El método `consultar_entregables` devuelve la información sobre los entregables evaluados relacionados con una evaluación en particular, cuya ID se debe proveer.
- El método `listado` devuelve el conjunto de revisiones de artículos que ya han sido evaluados.
- El método `evaluados`.

3.3.6. Reply

Cuando un usuario no está conforme con una evaluación que han hecho sobre una aportación suya, o cuando un evaluador quiere modificar a posteriori una evaluación ya enviada, es posible añadir una réplica, que se añade como una evaluación más. Para representar la relación entre una evaluación y sus evaluaciones *hijas* o *replies*, se utiliza el modelo *Reply* (`application/models/reply_model.php`).

Este modelo se enlaza con la tabla **replies**, que simplemente modela una relación 1:N entre evaluaciones *padre* y sus réplicas. El modelo ofrece varias funciones para guardar replies y obtener las replies asociadas a una evaluación.

3.3.7. Revisión

AssessMediaWiki no trabaja sobre artículos directamente, sino sobre las distintas ediciones, o **revisiones**, que han ido haciéndose sobre cada artículo. El modelo *Revisiones* (`application/models/revisiones_model.php`) permite obtener información sobre revisiones de los artículos. En particular:

- El método `articulos` devuelve las revisiones de artículos relacionadas por un usuario particular, cuyo ID se debe proveer.
- El método `usuarioArticulo` devuelve el ID del usuario autor de una revisión particular, cuyo ID se debe proveer.
- El método `listado_validas` devuelve un listado de revisiones que podrían ser sujeto de evaluación. Este método recibe un conjunto de revisiones ya evaluadas y un usuario, y busca entre todas las revisiones no evaluadas y que no pertenezcan al usuario indicado. Además, filtra las revisiones para que solo se incluyan los artículos de la categoría indicada, y que se hayan realizado en el periodo de tiempo indicado.

3.3.8. Usuarios

Este modelo (`application/models/usuarios_model.php`) establece una conexión con la BD de MediaWiki y almacena el ID y nombre de todos los usuarios registrados.

Además, dado un ID de un usuario se encarga de comprobar, mediante su método `admin`, si el usuario indicado es administrador o no.

3.4. Controladores

Como se ha comentado previamente, los controladores dirigen la lógica de la aplicación, haciendo uso de los datos contenidos en los *modelos* y mostrándolos mediante las *vistas*. Los controladores de los que dispone la aplicación son los siguientes:

3.4.1. Acceso

El controlador de acceso (`application/controllers/acceso.php`) se encarga de controlar el acceso de los usuarios a la aplicación.

Su método `index` muestra y gestiona el formulario de login a la aplicación. Los usuarios que se utilizan en el acceso a AMW son los presentes en la tabla de usuarios de MediaWiki, y esta lectura se hace a través de los métodos del modelo de acceso. Una vez superado el login, el usuario es redirigido a la acción principal del controlador `Evaluar`.

Su método `salir` borra la información de login y redirige el usuario al panel de acceso inicial.

El controlador de acceso está definido en el fichero de configuración de rutas (`application/config/routes.php`) como el **controlador por defecto**, de forma que al acceder a la aplicación sin indicar ningún parámetro, será este controlador el que se haga cargo de la petición.

3.4.2. Alumnos

El controlador *Alumnos* (`application/controllers/alumnos.php`) muestra una lista de todos los alumnos presentes en el wiki (y, por extensión, en AMW), y permite acceder a información sobre las acciones de estos. En concreto, muestra enlaces a las revisiones hechas por cada alumno así como a un documento CSV⁸ con información que puede ser usada para minería de datos.

Este controlador solo permite el acceso a los usuarios que sean administradores. El resto de usuarios es redirigido al panel principal de evaluación.

3.4.3. Evaluar

El controlador *Evaluar* (`application/controllers/evaluar.php`) es el más importante de toda la aplicación, ya que gestiona la funcionalidad principal de la misma: la emisión de evaluaciones.

Método index El método `index` es el primero al que accede el usuario. Se encarga de presentarle al usuario la revisión que debe hacer, o de informarle de que no tiene evaluaciones pendientes. Para ello:

1. Carga la lista de revisiones pendientes de evaluación.
2. Comprueba si el usuario tiene evaluaciones pendientes de completar o si por el contrario ya ha completado todas sus evaluaciones.
3. Si hay revisiones por evaluar, elige una de ellas aleatoriamente, carga la información y finalmente muestra al usuario el formulario de evaluación.
4. Si no hay revisiones por evaluar, muestra un mensaje informativo.

Método procesar Este método recibe los datos procedentes del formulario de evaluación, los introduce de forma correcta en la base de datos y redirige al usuario a la revisión de la evaluación recién terminada.

Método mostrar_evaluacion Se encarga de mostrar la información relacionada con una evaluación, como los conceptos evaluados, la calificación y comentarios realizados, así como las posibles respuestas (*replies*) que se han añadido a la evaluación.

⁸CSV - Comma Separated Values

Método `reply` Muestra el formulario para añadir observaciones sobre una evaluación en forma de meta-evaluación.

Método `reply_submit` Se encarga de trasladar a la base de datos los datos recibidos de un formulario de introducción de *reply* (al que se accede desde el método anterior).

3.4.4. Feedback

El controlador *Feedback* (`application/controllers/feedback.php`) se encarga de mostrar información sobre las acciones que han realizado los usuarios en la aplicación. En su acción por defecto (tanto `index` como `informe`), muestra una lista de todas las evaluaciones que se han hecho en forma de tabla, con una columna para el número de revisión evaluada y otra columna con un enlace a la información de la revisión.

Además, tiene métodos para la generación de informes en formato CSV sobre las evaluaciones recibidas por un usuario.

3.4.5. Params

El controlador *Params* (`application/controllers/params.php`) se encarga de gestionar el CRUD⁹ de varios parámetros de configuración de la aplicación, en particular:

- Lista de conceptos evaluables (entregables).
- Categoría a la que pertenecen los artículos a evaluar.

⁹Create, Read, Update, Delete

4. Tareas pendientes

Actualmente existe una serie de mejoras del software pendientes de hacer. Se incluyen consejos para abordar estas tareas.

4.1. Subcategorías

Enlace https://forja.rediris.es/pm/task.php?func=detailtask&project_task_id=2861&group_id=1135&group_project_id=2120

Descripción Cuando se indica la categoría de los artículos evaluar, podría incluir la opción de indicar *“incluir subtareas”*.

El filtrado de artículos por categoría se realiza en el modelo *Revisiones*, al hacer la consulta SQL de las revisiones pendientes de evaluación. Sería posible añadir más opciones de filtrado de artículos, así como incluir diversas categorías.

Unificar acceso a lista de de usuarios, en algunos lados se accede directamente al array usuarios y en otros lados se utiliza la función users del modelo usuarios

Añadir comprobaciones cuando se pidan objetos inexistentes