

## ACTIVIDADES UD6 - SERVIDOR

Mediante las tareas que se detallan en este documento vamos a conseguir la parte de los criterios de evaluación cubiertos por la UD6:

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN
RA5. Desarrolla aplicaciones Web identificando y aplicando mecanismos para separar el código de presentación de la lógica de negocio.	<p>a) Se han identificado las ventajas de separar la lógica de negocio de los aspectos de presentación de la aplicación.</p> <p>b) Se han analizado y utilizado mecanismos y frameworks que permiten realizar esta separación y sus características principales.</p> <p>e) Se han identificado y aplicado los parámetros relativos a la configuración de la aplicación web.</p> <p>f) Se han escrito aplicaciones web con mantenimiento de estado y separación de la lógica de negocio.</p> <p>g) Se han aplicado los principios y patrones de diseño de la programación orientada a objetos.</p> <p>h) Se ha probado y documentado el código.</p>
RA6. Desarrolla aplicaciones web de acceso a almacenes de datos, aplicando medidas para mantener la seguridad y la integridad de la información	<p>a) Se han analizado las tecnologías que permiten el acceso mediante programación a la información disponible en almacenes de datos.</p> <p>b) Se han creado aplicaciones que establezcan conexiones con bases de datos.</p> <p>c) Se ha recuperado información almacenada en bases de datos.</p> <p>d) Se ha publicado en aplicaciones web la información recuperada.</p> <p>e) Se han utilizado conjuntos de datos para almacenar la información.</p>

Consideraciones adicionales:

- Los criterios de evaluación a) y b) del RA5, y el a) del RA 6 se dan por superados (calificación de 10) al hacer el ejemplo guiado de portfolio de la parte de introducción-exposición. La verificación de que se ha realizado el ejemplo guiado se llevará a cabo durante las clases presenciales.
- Se ha de continuar comentando el código mediante el código de tarea.
- Cualquier copia (ya sea de fuentes externas, literal de los apuntes...) que no sea una respuesta original será calificada con 0. El código no estructurado, que presente dificultades para ser leído no podrá ser evaluado.

- Cada actividad va a estar asociada a un criterio de evaluación.

### Actividad 1: Comentarios en el código

<b>Código:</b> UD6.1	<b>CE:</b> RA5.h	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 30 min
----------------------	------------------	-------------------	-------------------	---------------------------

#### DESCRIPCIÓN

Para cada una de las actividades que siguen a ésta, inserta comentarios en el código para poder delimitar cada una de las soluciones que des. Para eso, inserta un comentario previo a cada fragmento de código que crees, especificando el código de cada tarea.

El código de cada tarea se formará con el código de la actividad, más la letra que identifica a la tarea en particular.

Ten en cuenta que pueden ser comentarios en Python o en HTML.

Las respuestas que no sean identificables mediante comentario con el correspondiente código **podrán no ser evaluadas**.

#### ENLACES

- [Comentarios en Python](#)
- [Comentarios en HTML](#)

### Actividad 2: Puesta a punto del proyecto Django

<b>Código:</b> UD6.2	<b>CE:</b> RA5.e	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 2h 30min
----------------------	------------------	-------------------	-------------------	-----------------------------

#### DESCRIPCIÓN

En esta actividad vamos a crear el proyecto para la nueva aplicación web basada en Django. Realiza las siguientes tareas:

- Creación del entorno virtual, llamado avaluapp-env. Instala Django (versión 4.2). **(1.5 puntos)**
- Crea el proyecto, llamado avaluaproy. **(1.5 puntos)**
- Crea un superusuario para el proyecto. **(1 punto)**
- Realiza la recolección de estáticos del proyecto. Comprueba que se hayan creado las carpetas correspondientes. **(1 punto)**
- Realiza los cambios de configuración en settings.py y urls.py especificados en el documento de teoría (apartados 5.1 y 5.3.3). **(1 punto)**
- Crea las siguientes apps y añádelas a INSTALLED\_APPS: **(1.5 puntos)**
  - common
  - core
  - programacion\_didactica
  - programacion\_aula
- Crea la carpeta templates y configúrala en settings.py. Copia a esta carpeta las plantillas que se proporcionan en la tarea de Aules. **(1 punto)**
- Prepara el proyecto para poder ejecutarlo mediante Docker. **(1.5 puntos)**

### Actividad 3: Capa de datos

<b>Código:</b> UD6.3	<b>CE:</b> RA6.b	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 3 h
----------------------	------------------	-------------------	-------------------	------------------------

#### DESCRIPCIÓN

Al final de este documento encontrarás el **Anexo I** con las especificaciones de los modelos iniciales que vamos a definir en este proyecto.

- a) En la Tabla 1 encontrarás la definición de los modelos. Implementa las clases especificadas en los ficheros models.py correspondientes a cada app. **Respetar** el nombre de los modelos, atributos y funciones (si se especifican), no los modifiques. Aplica los comandos makemigrations y migrate. Modelos: **(7 puntos)**
  - core - Modulo **(0.5 puntos)**
  - core - ResAprendizaje **(1 punto)**
  - core - CritEvaluacion **(1 punto)**
  - programacion\_didactica - Unidad **(0.5 puntos)**
  - programacion\_didactica - InstEvaluacion **(0.5 puntos)**
  - programacion\_didactica - PondRA **(1 punto)**
  - programacion\_didactica - PondCriterio **(1 punto)**
  - programacion\_didactica - PondCritUD **(1.5 puntos)**
- b) Para cada modelo, implementa su función `__str__` según la Tabla 2. Esta función es necesaria para que en el administrador de Django se puedan identificar correctamente los objetos de una clave foránea **(2 puntos)**
- c) Implementa las restricciones de clave única especificadas en la Tabla 3. En la clase Meta del modelo correspondiente, tendrás que definir el atributo `unique_together`. La clase Meta de un modelo nos va a servir para cambiar aspectos del comportamiento del modelo. **(1 punto)**

#### ENLACES

- [Modelos en Django](#)
- [Clase Meta](#)

### Actividad 4: Configuración del administrador de Django

<b>Código:</b> UD6.4	<b>CE:</b> RA5.g	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 2h
----------------------	------------------	-------------------	-------------------	-----------------------

#### DESCRIPCIÓN

En esta actividad vamos a configurar el administrador de Django para los nuevos modelos. [Aquí](#) encontrarás una referencia de los atributos de la clase ModelAdmin.

Realiza las siguientes tareas:

- a) La primera vez que definas un atributo de la clase ModelAdmin, pon un comentario en la misma línea que resuma su propósito. **(1 punto)**
- b) Implementa las clases para la app core, según la Tabla 1 del Anexo II:

- ModAdmin (**0.5 puntos**)
  - ResAprendizajeAdmin (**0.5 puntos**)
  - CritEvaluacionAdmin (**1 puntos**)
- c) Implementa las clases para la app programacion\_didactica, según la Tabla 2 del Anexo II:
- UnidadAdmin (**0.5 puntos**)
  - InstEvalAdmin (**0.5 puntos**)
  - PondRAAdmin (**1 puntos**)
  - PondCritAdmin (**1.5 puntos**)
  - PondCritUDAdmin (**1.5 puntos**)
- d) Dale un valor al atributo verbose\_name de las apps para que se muestre en el administrador. Esto se modifica en el fichero apps.py (uno por cada app), añadiendo el atributo verbose\_name a la clase que hereda de AppConfig. [Aquí](#) tienes un ejemplo. (**0.5 puntos**)
- e) Dale un valor a los atributos verbose\_name y verbose\_name\_plural a la clase Meta de cada modelo. Esto se utiliza en el administrador para que Django no autogenera un nombre para cada modelo de forma automática. De esta forma podemos controlar el nombre que va a aparecer asociado a cada modelo. Abajo tienes un enlace sobre la clase Meta, consúltalo si aún no lo has hecho (**0.75 puntos**)
- f) Dale un valor al atributo ordering de la clase Meta de cada modelo. Esto se utiliza para determinar la ordenación al recuperar los registros de un modelo. En la clase ModelAdmin se puede modificar este comportamiento, definiendo también un atributo ordering, según se detalla en el enlace más abajo. Añade ordering a cada modelo, según la Tabla 3 del Anexo II (**0.75 puntos**)

#### ENLACES

- [Clase Meta](#)

#### Actividad 5: Introducción de datos en el administrador

<b>Código:</b> UD6.5	<b>CE:</b> RA6.e	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 1h
----------------------	------------------	-------------------	-------------------	-----------------------

#### DESCRIPCIÓN

Con el fin de crear datos de prueba, vamos a introducir manualmente registros en la aplicación a través del administrador de Django, después de haberlo configurado en la actividad anterior.

Para ello, toma como referencia la Programación Didáctica del presente módulo (disponible en Aules) e inserta los siguientes datos (con todos los campos definidos en el modelo correspondiente, según los datos de la Programación Didáctica):

- a) Módulo de desarrollo en entorno servidor y cliente, según el modelo Modulo (**1 punto**).
- b) Las unidades 6 y 9, según el modelo Unidad (**1.5 puntos**).
- c) Dos instrumentos de evaluación (**1.5 puntos**).
- d) Los dos resultados de aprendizaje asociados a las unidades introducidas (RA5 y RA6) (**1**

**punto).**

- e) Todos los criterios de evaluación asociados a los resultados de aprendizaje anteriores (**2 puntos**).
- f) La ponderación de los dos resultados de aprendizaje respecto a la calificación final (**1.5 puntos**).
- g) La ponderación de cada uno de los criterios de evaluación en esas dos unidades (**1.5 puntos**).

### Actividad 6: ORM de Django

<b>Código:</b> UD6.6	<b>CE:</b> RA6.c	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 3h
----------------------	------------------	-------------------	-------------------	-----------------------

#### DESCRIPCIÓN

En esta actividad vamos a practicar con el ORM de Django.

Define un nuevo fichero `actividadorm.py` dentro de la app `common`, que contenga un método por cada una de las operaciones a realizar.

En el anexo IV tienes una tabla con el nombre que has de dar a cada método, y la descripción de lo que ha de hacer. Guíate en este [enlace](#), o también [este otro](#), para buscar las instrucciones a realizar.

Para practicar las instrucciones, activa la consola del entorno virtual con el siguiente comando:

`python manage.py shell`

(Con este comando, se carga toda la configuración de nuestro proyecto)

**Cada operación puntúa con 0.5 puntos.**

### Actividad 7: Programación de la aplicación

<b>Código:</b> UD6.7	<b>CE:</b> RA5.f	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 3 h
----------------------	------------------	-------------------	-------------------	------------------------

#### DESCRIPCIÓN

En esta actividad vamos a programar una parte de las vistas del módulo `programacion_didactica`, sus URLs, y vamos a crear sus plantillas correspondientes.

Junto al enunciado de la práctica se proporciona los siguientes archivos, de ejemplo:

- `views.py` de la app `core` y `common` (sitúa cada archivo en su carpeta correspondiente).
- `templates` de la app `common` y la app `core`.
- `urls.py` de la versión básica de la aplicación.

Se trata de realizar lo siguiente:

- a) Programa las vistas de la app `programacion_didactica`, según la [Tabla 1](#) del anexo III. Por el momento asígnales el nombre de la plantilla correspondiente. Las plantillas serán configuradas en la actividad 8. (**5 puntos**)
- b) Crea las URLs para la app `programacion_didactica`, según la [Tabla 2](#) del anexo III. (**5 puntos**)

**Actividad 8: Publicación de los datos**

<b>Código:</b> UD6.8	<b>CE:</b> RA6.d	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 3 h
----------------------	------------------	-------------------	-------------------	------------------------

**DESCRIPCIÓN**

En esta actividad vamos a configurar una parte de las plantillas del módulo programacion\_didactica.

- a) Crea las plantillas de la app programacion\_didactica, según la Tabla 3 del anexo III. (7 puntos)
- b) Implementa la paginación en las vistas y templates de listas, de la app core. Bázate en este [enlace](#). (3 puntos)

## ANEXO I

**Tabla 1:** definición de modelos

app	Nombre	Campos
core	Modulo	- codigo: Charfield, longitud máxima de 10, único. - nombre: Charfield, longitud máxima de 256, único
core	ResAprendizaje	- modulo: ForeignKey a Modulo (on_delete=models.PROTECT). - codigo: Charfield, longitud máxima de 10, <b>único</b> . - descripcion: Textfield
core	CritEvaluacion	- resultado_aprendizaje: ForeignKey a ResAprendizaje (on_delete=models.PROTECT). - codigo: Charfield, longitud máxima de 4. - descripcion: Textfield - minimo: BooleanField
programacion_ didactica	Unidad	- codigo: Charfield, longitud máxima de 4 caracteres, único. - nombre: Charfield, longitud máxima de 256, único.
programacion_ didactica	InstEvaluacion	- codigo: Charfield, longitud máxima de 4, único. - nombre: Charfield, longitud máxima de 256, único. - descripcion: Textfield
programacion_ didactica	PondRA	- resultado_aprendizaje: ForeignKey a ResAprendizaje (on_delete=models.PROTECT). - porcentaje: DecimalField, 2 decimales, dígitos máximos 5, 0 por defecto.
programacion_ didactica	PondCriterio	- criterio_evaluacion: ForeignKey a CritEvaluacion (on_delete=models.PROTECT). - porcentaje: DecimalField, 2 decimales, dígitos máximos 5, 0 por defecto.
programacion_ didactica	PondCritUD	- criterio_evaluacion: ForeignKey a CritEvaluacion (on_delete=models.PROTECT). - unidad: ForeignKey a Unidad (on_delete=models.PROTECT). - porcentaje: DecimalField, 2 decimales, dígitos máximos 5, 0 por defecto.

**Tabla 2:** Representación textual de modelos (función `__str__`):



Modelo	Lo que ha de devolver __str__
Modulo	nombre
ResAprendizaje	codigo + '.' + 100 primeros caracteres de la descripcion
CritEvaluacion	codigo del resultado de aprendizaje + '.' + codigo del criterio + ' - ' + 100 primeros caracteres de la descripcion del criterio
Unidad	codigo + ' - ' + nombre
InstEvaluacion	codigo + ' - ' + nombre
PondRA	representación textual del resultado de aprendizaje + ' - ' + porcentaje NOTA: Para conseguir la representación textual de otro objeto, utiliza la función str de la forma: str(self.resultado_aprendizaje). Para el porcentaje, pásalo a str también para que no devuelva error de tipo: str(self.porcentaje)
PondCriterio	representación textual del criterio de evaluación + ' - ' + porcentaje
PondCritUD	representación textual de la unidad + ': ' + representación textual del criterio de evaluación + ' - (' + porcentaje + ')'

**Tabla 3:** restricciones de combinación única

Modelo	Campos
ResAprendizaje	- modulo - codigo
CritEvaluacion	- resultado_aprendizaje - codigo
PondCritUD	- criterio_evaluacion - unidad

## ANEXO II

Tabla 1: Clases de core basadas en ModelAdmin:

Clase	Atributos
ModuloAdmin	<ul style="list-style-type: none"> <li>• list_display: id, codigo, nombre</li> <li>• list_display_links: codigo, nombre</li> <li>• search_fields: codigo, nombre</li> <li>• readonly_fields: id</li> </ul>
ResAprendizajeAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: codigo, modulo, descripcion</li> <li>• list_filter: modulo</li> <li>• search_fields: codigo, descripcion</li> <li>• preserve_filters: verdadero</li> </ul>
CritEvaluacionAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: resultado_aprendizaje, codigo, descripcion, minimo</li> <li>• list_filter: resultado_aprendizaje, minimo</li> <li>• search_fields: codigo, resultado_aprendizaje, descripcion</li> <li>• preserve_filters: verdadero</li> </ul>

Tabla 2: Clases de programacion\_didactica basadas en ModelAdmin:

Clase	Atributos
UnidadAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: codigo, nombre</li> <li>• search_fields: codigo, nombre</li> <li>• ordering: id, en orden inverso</li> </ul>
InstEvalAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: codigo, nombre</li> <li>• search_fields: codigo, nombre</li> </ul>
PondRAAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: resultado_aprendizaje, porcentaje</li> <li>• list_filter: resultado_aprendizaje</li> <li>• search_fields: <ul style="list-style-type: none"> <li>◦ Código del resultado</li> <li>◦ Descripción del resultado</li> </ul> </li> <li>• preserve_filters: verdadero</li> </ul>
PondCritAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: criterio_evaluacion, porcentaje</li> <li>• list_filter: resultado de aprendizaje del criterio</li> <li>• search_fields:</li> </ul>

	<ul style="list-style-type: none"> <li>◦ Código del criterio</li> <li>◦ Descripción del criterio</li> <li>◦ Código del resultado</li> <li>◦ Descripción del resultado</li> </ul> <ul style="list-style-type: none"> <li>• preserve_filters: verdadero</li> </ul>
PondCritUDAdmin	<ul style="list-style-type: none"> <li>• list_display y list_display_links: unidad, criterio_evaluacion, porcentaje</li> <li>• list_filter: resultado_aprendizaje, unidad</li> <li>• search_fields: <ul style="list-style-type: none"> <li>◦ Código del criterio</li> <li>◦ Descripción del criterio</li> <li>◦ Código del resultado</li> <li>◦ Descripción del resultado</li> <li>◦ Nombre de la unidad</li> </ul> </li> <li>• preserve_filters: verdadero</li> </ul>

**Tabla 3:** Ordenación por defecto de los modelos

Modelo	Ordenación
Modulo	codigo
ResAprendizaje	codigo
CritEvaluacion	Primero por el código de su resultado de aprendizaje, y después por el código del criterio de evaluación.
Unidad	id
InstEvaluacion	codigo
PondRA	Código de su resultado de aprendizaje
PondCriterio	Código del criterio de evaluación
PondCritUD	Primero por el nombre de la unidad, después por el código del criterio de evaluación.

## ANEXO III

Tabla 1: vistas de la app programacion\_didactica

Vista	Características
UnidadListView	- Modelo: Unidad - Template: programacion_didactica/unidad_list.html
UnidadDetailView	- Modelo: Unidad - Template: programacion_didactica/unidad_detail.html
InstEvListView	- Modelo: InstEvaluacion - Template: programacion_didactica/ie_list.html
InstEvDetailView	- Modelo: InstEvaluacion - Template: programacion_didactica/ie_detail.html
PondRAListView	- Modelo: PondRA - Template: programacion_didactica/pond_ra_list.html
PondRADetailView	- Modelo: PondRA - Template: programacion_didactica/pond_ra_detail.html
PondCritListView	- Modelo: PondCriterio - Template: programacion_didactica/pond_ce_list.html
PondCritDetailView	- Modelo: PondCriterio - Template: programacion_didactica/pond_ce_detail.html
PondCritUDListView	- Modelo: PondCritUD - Template: programacion_didactica/pond_ce_ud_list.html
PondCritUDDetailView	- Modelo: PondCritUD - Template: programacion_didactica/pond_ce_ud_detail.html

Tabla 2: URLs de la app programacion\_didactica

Nombre URL	Características
unidad_list	- Estructura: unidad_list/ - Vista: UnidadListView
unidad_detail	- Estructura: unidad_detail/<int:pk>/ - Vista: UnidadDetailView
ie_list	- Estructura: ie_list/

	- Vista: InstEvListView
ie_detail	- Estructura: ie_detail/<int:pk>/ - Vista: InstEvDetailView
pond_ra_list	- Estructura: pond_ra_list/ - Vista: PondRAListView
pond_ra_detail	- Estructura: pond_ra_detail/<int:pk>/ - Vista: PondRADetailView
pond_ce_list	- Estructura: pond_ce_list/ - Vista: PondCritListView
pond_ce_detail	- Estructura: pond_ce_detail/<int:pk>/ - Vista: PondCritDetailView
pond_ce_ud_list	- Estructura: pond_ce_ud_list/ - Vista: PondCritUDListView
pond_ce_ud_detail	- Estructura: pond_ce_ud_detail/<int:pk>/ - Vista: PondCritUDDetailView

**Tabla 3:** templates de la app programacion\_didactica

Template	Información a mostrar
unidad_list.html	- Título: Lista de unidades - Texto: representación textual del objeto
unidad_detail.html	- Título: Detalle de la uniad - Campos: codigo, nombre
ie_list.html	- Título: Lista de instrumentos de evaluación - Texto: representación textual del objeto
ie_detail.html	- Título: Detalle del instrumento de evaluación - Campos: codigo, nombre, descripcion
pond_ra_list.html	- Título: Lista de ponderaciones de RA - Texto: representación textual del objeto
pond_ra_detail.html	- Título: Detalle de ponderación de RA - Campos: resultado_aprendizaje, porcentaje
pond_ce_list.html	- Título: Lista de ponderaciones de CE - Texto: representación textual del objeto

pond_ce_detail.html	- Título: Detalle de ponderación de CE - Campos: criterio_evaluacion, porcentaje
pond_ce_ud_list.html	- Título: Lista de ponderaciones de CE por UD - Texto: representación textual del objeto
pond_ce_ud_detail.html	- Título: Detalle de ponderación de CE por UD - Campos: criterio_evaluacion, unidad, porcentaje

## ANEXO IV

#	Nombre del método	Descripción
1	get_modulos_all	Devuelve un queryset con todos los módulos.
2	get_ie_3	Devuelve un queryset con los 3 primeros instrumentos de evaluación.
3	get_crit_min	Devuelve un queryset con los criterios de evaluación mínimos.
4	get_ra_not_0612	Devuelve un queryset con todos los resultados de aprendizaje, excluyendo aquellos cuyo código empiece por "0612".
5	get_ie_l1_defensa	Devuelve un queryset de los instrumentos de evaluación cuyo código sea l1 y que en el nombre contenga "defensa".
6	get_mod_id_lt_2_cod_des	Devuelve un queryset de los módulos cuyo id sea menor a 2, ordénalos por código, descendente.
7	get_mod_id_gt_2	Devuelve un queryset de los módulos cuyo id sea mayor a 2.
8	get_mod_des_nom_des	Devuelve un queryset con los módulos que contienen la palabra "desarrollo", ordenados por nombre ascendente.
9	get_ra_not_RA1	Devuelve un queryset con todos los resultados de aprendizaje, excluyendo aquellos cuyo código acabe en "1".
10	get_ra_cli_RA3	Devuelve un <b>objeto</b> (no un queryset) de la clase ResAprendizaje cuyo módulo contenga "cliente" y cuyo código (el del resultado de aprendizaje) acabe en "3".
11	get_ce_RA06_not_cli	Devuelve un queryset con todos los criterios de evaluación cuyo resultado de aprendizaje contenga "06" en el código. Se han de excluir los del módulo que contenga en el nombre "cliente".
12	get_pon_ra_gt_5	Devuelve un queryset de las ponderaciones de los resultados de aprendizaje con un porcentaje mayor o igual a 5.
13	get_ce_5_10_RA1	Devuelve un queryset de las ponderaciones de los criterios de evaluación con un porcentaje entre 5 y 10, cuyo resultado de aprendizaje tenga un código que acabe en "1" y cuyo código de módulo sea "0613".
14	create_ie12	Crea un instrumento de evaluación con código l12, nombre "Nuevo instrumento 12", y la descripción igual al nombre.
15	create_ud15_pon	Crea una nueva ponderación para: - Nueva unidad UD15, dale datos de tu elección. - Un criterio de evaluación de tu elección. - Ponderación 20.

16	update_last_ud	Modifica el nombre de la última unidad para concatenar el texto "(última)".
17	update_ce_RA2_x2	Modifica las ponderaciones de los criterios de evaluación asociados al resultado de aprendizaje cuyo código acabe en "2" y cuyo código de módulo sea "0613", para que su ponderación se multiplique por 2.
18	update_pon_ce_RA1_div 2	Modifica las ponderaciones de los criterios de evaluación asociados al resultado de aprendizaje con código que acabe en "1" y cuyo código de módulo sea "0613", que tengan una ponderación par, y divídelas por 2.
19	delete_pon_ce_lt_5	Eliminar las ponderaciones de los criterios de evaluación menores a 5.
20	delete_pon_ce_UD15	Elimina las ponderaciones de los criterios de evaluación por unidad, cuya unidad contenga "UD3" en el nombre. Crea esta unidad si es necesario para hacer las pruebas.