# ACTIVIDADES UD9 - SERVIDOR

Mediante las tareas que se detallan en este documento vamos a conseguir la parte de los resultados de aprendizaje cubiertos por la UD9, atendiendo a sus correspondientes criterios de evaluación:

| 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. | e) Se han identificado y aplicado los parámetros relativos a la configuración de la aplicación Web. |
| RA6. Desarrolla aplicaciones web de acceso a almacenes de datos, aplicando medidas para mantener la seguridad y la integridad de la información. | b) Se han creado aplicaciones que establezcan conexiones con bases de datos.<br>f) Se han creado aplicaciones web que permitan la actualización y la eliminación de información disponible en una base de datos.<br>g) Se han probado y documentado las aplicaciones web. |

Consideraciones adicionales:

- 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 se leído no podrá ser evaluado.

| Actividad 1: Comentarios en el código | | | | |
|---|---|---|---|---|
| **Código**: UD9.1 | **CE**: RA6.g | IE: I3, I6 | **Puntos**: 10 | **Estimación**: 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: PostgreSQL | | | | |
|---|---|---|---|---|
| **Código**: UD9.2 | **CE**: RA5.e | IE: I3, I6 | **Puntos**: 10 | **Estimación**: 1 h |

**DESCRIPCIÓN**

En primer lugar, crea una copia de tu aplicación para esta nueva unidad 9.

A continuación, realiza los siguientes pasos (detallados en el enlace más abajo):

a) Actualiza el docker-compose.yml de tu proyecto para incluir el nuevo servicio "db", según los parámetros del ejemplo del enlace. (**2 puntos**)

b) Modifica settings.py para que el motor de BBDD de la aplicación sea PostgreSQL (básate en el mismo enlace). (**2 puntos**)

c) Realiza las migraciones de las aplicaciones conforme se especifica en el tutorial. (**6 puntos**)

**ENLACES**

- Django, Docker, and PostgreSQL Tutorial

| Actividad 3: Migración de datos | | | | |
|---|---|---|---|---|
| **Código**: UD9.3 | **CE**: RA6.f (**70%**) | IE: I3, I6 | **Puntos**: 10 | **Estimación**: 3 h |

**DESCRIPCIÓN**

Una de las consecuencias de haber sustituido SQLite por PostgreSQL, es que hemos perdido los datos que habíamos creado hasta ahora (realmente los seguimos conservando en el fichero db.sqlite3).

Pero hemos de pensar que, cuando vayamos a desplegar nuestra aplicación a un entorno de producción, tendremos el mismo problema, ya que SQLite está pensada para un entorno de

desarrollo.

Es por ello que en esta actividad vamos a crear ficheros de migración para los datos iniciales. Revisamos el enlace adjunto y lo explicamos en clase.

Con todo esto, vamos a crear tres ficheros de migración, para las siguientes tres apps:

a) core_migracion_inicial.py: contendrá la creación de los datos iniciales de los siguientes modelos: (**4 puntos**)

- Modulo: solo crearemos un módulo, el de Desarrollo en entorno servidor
- ResultadoAprendizaje: crearemos todos los RAs del módulo.
- CriterioEvaluacion: todos los CEs de todos los RAs del módulo.

b) prog_didactica_migracion_inicial.py: en este caso, trasladaremos a los siguientes modelos la información correspondiente, especificada en la programación didáctica del módulo (publicada en Aules, en el aula de Tutoría): (**5 puntos**)

- InstrumentoEvaluacion
- PonderacionRA
- PonderacionCriterio
- PonderacionCriterioUD

c) users_migracion_inicial.py (**1 punto**): crearemos un usuario administrador para el modelo MyUser.

**ENLACES**

- Migraciones de datos

| **Actividad 4: Modelos** | | | | |
|---|---|---|---|---|
| **Código**: UD9.4 | **CE**: RA6b | IE: I3, I6 | **Puntos**: 10 | **Estimación**: 2 h |

**DESCRIPCIÓN**

En esta actividad vamos a crear modelos abstractos, y nuevos modelos para varias de las apps ya creadas.

Implementa los modelos especificados en el Anexo I, tabla 1.

| **Actividad 5: Disparadores** | | | | |
|---|---|---|---|---|
| **Código**: UD9.5 | **CE**: RA6.f (**30%**) | IE: I3, I6 | **Puntos**: 10 | **Estimación**: 2 h |

**DESCRIPCIÓN**

Habrás notado que el atributo "calificacion" del modelo CalificacionUDCE es opcional. Esto se debe a que vamos a pre-generar los objetos de este modelo. ¿Cuándo? cada vez que se cree un nuevo objeto Alumno. ¿Por qué? porque si no, tendríamos que insertar uno a uno cada uno de estos registros en la BBDD, lo cual puede ser muy tedioso.

Vamos a pregenerar estos registros sin calificación, y después les pondremos la calificación.

Para ello vamos a utilizar dos herramientas:

- Los objetos creados en el modelo CriterioEvalUD, que relaciona cada unidad con sus correspondientes criterios de evaluación, y que son a lo que hemos de dar una calificación. Una vez hayas creado este modelo, crea los registros para las seis primeras unidades del curso (bien mediante migraciones, mediante el administrador, consola, etc.).
- Las señales de Django (ver enlaces). Cada vez que se cree un nuevo objeto Alumno, se disparará una señal que insertará en CalificacionUDCE tantos registros como asociaciones UD - Crit. Eval (según la información de CriterioEvalUD), para ese alumno creado.

Elige correctamente el tipo de señal que necesitas utilizar.

**ENLACES**

- Introducción a las señales
- Signals

# ANEXO I

**Tabla 1**: definición de modelos

| app | Nombre | Campos |
|---|---|---|
| common | Persona (abstracto)<br>(**1 punto**) | - nombre: Charfield, longitud máxima de 128. Obligatorio.<br>- apellidos: Charfield, longitud máxima de 256. Obligatorio. |
| common | Localizacion (abstracto)<br>(**1 punto**) | - direccion: Charfield, longitud máxima de 256. Obligatorio.<br>- codigo_postal: Charfield, longitud máxima de 5. Obligatorio.<br>- ciudad: Charfield, longitud máxima de 256. Obligatorio. |
| programacion_aula | Alumno<br>(**1 punto**) | - Hereda de Persona y Localizacion |
| programacion_aula | CriterioEvalUD<br>(**1 punto**) | Almacenará qué criterios de evaluación están asociados a cada unidad.<br>- unidad: clave foránea a programacion_didactica.Unidad. Protegido. Obligatorio.<br>- criterio_evaluacion: clave foránea a prog_didactica.CritEvaluacion. Protegido. Obligatorio. |
| programacion_aula | CalificacionUDCE<br>(**1.5 puntos**) | - alumno: clave foránea a programacion_aula.Alumno. Protegido. Obligatorio.<br>- unidad: clave foránea a programacion_didactica .Unidad. Protegido. Obligatorio.<br>- crit_evaluacion: clave foránea a programacion_didactica.CritEvaluacion. Protegido. Obligatorio.<br>- calificacion: DecimalField, 5 dígitos máximo, 2 lugares decimales. **Opcional**. |
| programacion_aula | CalificacionCE<br>(**1.5 puntos**) | - alumno: clave foránea a programacion_aula.Alumno. Protegido. Obligatorio.<br>- crit_evaluacion: clave foránea a programacion_didactica.CritEvaluacion. Protegido. Obligatorio.<br>- calificacion: DecimalField, 5 dígitos máximo, 2 lugares decimales. Opcional. |
| programacion_ | CalificacionRA | - alumno: clave foránea a programacion_aula.Alumno. |

| | | |
|---|---|---|
| aula | (**1.5 puntos**) | Protegido. Obligatorio.<br>- res_aprendizaje: clave foránea a programacion_didactica.ResAprendizaje. Protegido. Obligatorio.<br>- calificacion: DecimalField, 5 dígitos máximo, 2 lugares decimales. Opcional. |
| programacion_aula | CalificacionTotal<br>(**1.5 puntos**) | - alumno: clave foránea a programacion_aula.Alumno. Protegido. Obligatorio.<br>- modulo: clave foránea a core.Modulo. Protegido. Obligatorio.<br>- calificacion: DecimalField, 5 dígitos máximo, 2 lugares decimales. Opcional. |