

Acceso a Datos

Prueba

UD 03: Gestión de la Persistencia de los Datos con ORM

UD 04: Gestión de la persistencia con base de datos orientada a objetos

Tabla de contenidos

Instrucciones generales:	1
Criterios de evaluación:	2
Actividad única:	4
Rúbrica:	5
Traza de ejemplo:	5

Instrucciones generales:

- El alumnado deberá descargarse este enunciado de la Moodle durante los primeros 10 minutos de clase.
- Cada alumno **es responsable** de tener en su sistema un servidor de base de datos Mysql, virtualenv, un intérprete de Python y todas las librerías pip necesarias para ejecutar las actividades de las UD3y4. Si un alumno no tiene listo esto antes de la prueba, no podrá descargar nada durante la prueba, dado que durante el transcurso de la misma está prohibido conectarse a Internet.
- Esta prueba consta de 1 actividad. El resultado de la actividad se subirá a Moodle. Debe ser un archivo con un programa en Python. El nombre del archivo debe ser "<nombre>_<apellido1>_<apellido2>_prueba3y4.py". Todo en minúsculas y sin tildes. Por ejemplo "amanda_salas_casas_prueba3y4.py"
- Después de los 10 primeros minutos de clase, se dispone de 1h30min de tiempo para realizar la prueba. Dentro de este tiempo:
 - El ordenador del alumnado permanecerá desconectado de Internet. Tanto por cable como de forma inalámbrica.
 - Se deberá permanecer en absoluto silencio.
 - No se podrá utilizar ningún dispositivo electrónico
 - No se podrá consultar nada en Internet.
 - Si algún/a alumn@ tiene una pregunta para el profesor, levantará la mano y esperará a ser atendido/a.
 - Cuando el/la estudiante haya terminado su prueba, levantará la mano, pedirá permiso para conectar su ordenador a Internet y subirá a Moodle su prueba, siempre bajo la atenta mirada del profesor.

- o Cuando el/la estudiante haya realizado su entrega seguirán vigentes todas las normas descritas en estas instrucciones (no usar dispositivos electrónicos, mantener silencio, etc....)
- A partir de la hora y 30 minutos, los alumnos que no hayan entregado la prueba dejarán de trabajar en la misma y pondrán su teclado encima del ordenador. A partir de este momento el alumno que desee entregar su prueba, lo hará copiando su **fichero py** en un pen drive que entregará el profesor. Este mecanismo se usará durante el tiempo restante hasta que termine la clase. El orden elegido al azar para entregar la prueba en pendrive será éste:

1. González Justo, Inés	11. Pedrera López, Ángela Kaijing
2. Rodríguez Cardenal, Carmen	12. Mejías De Matos, Francisco
3. Pineda Toro, Antonio	13. Rodríguez Gómez, Jesús Daniel
4. Alonso Fernández, Felipe	14. Alcaide Pérez, Mario
5. Bárcena López, Guillermo	15. López Navarro-Casas, Olalla
6. Solano Delgado, Isaac	16. González Rodríguez, Fernando
7. Rodríguez Segura, Pablo	17. Iglesias Fernández, José María
8. Aguilar Castellanos, Raquel	18. Ramos Navarro, Andrés
9. Bernáldez Carrasco, Manuel	19. Romero Iglesias, Manuel
10. Rodríguez Peña, Pablo	
- El/la estudiante ha de tener en cuenta que el profesor ejecutará su script en su propio ordenador.
- El/la estudiante tendrá en cuenta que la base de datos a usar se denominará 1dam. El usuario para acceder a la base de datos se llama “usuario” y su clave será “usuario”. Si el alumno no tiene preparado esto antes de realizar la prueba será **responsabilidad suya** el perjuicio que le pueda acarrear durante la ejecución de la misma. Deberá llevar a cabo esta tarea durante los días anteriores a la prueba y no el mismo día en que tenga lugar.
- El/la estudiante deberá descargarse la documentación que considere oportuna para la realización de la prueba durante los días anteriores a la misma y no el mismo día de la prueba. Si el alumno no tiene preparado esto antes de realizar la prueba será **responsabilidad suya** el perjuicio que le pueda acarrear durante la ejecución de la misma.

Criterios de evaluación:

- a) Se ha instalado la herramienta ORM.
 - b) Se ha configurado la herramienta ORM.
 - c) Se han definido configuraciones de mapeo.
 - d) Se han aplicado mecanismos de persistencia a los objetos.
 - e) Se han desarrollado aplicaciones que modifican y recuperan objetos persistentes.
 - f) Se han desarrollado aplicaciones que realizan consultas usando el lenguaje SQL.
 - g) Se han gestionado las transacciones.
-
- a) Se han identificado las ventajas e inconvenientes de las bases de datos que almacenan objetos.

- b) Se han establecido y cerrado conexiones.
 - c) Se ha gestionado la persistencia de objetos simples.
 - d) Se hA gestionado la persistencia de objetos estructurados.
 - e) Se han desarrollado aplicaciones que realizan consultas.
 - f) Se han modificado los objetos almacenados.
 - g) Se han gestionado las transacciones.
 - h) Se han probado y documentado las aplicaciones desarrolladas.
-

Actividad única

La biblioteca municipal ha decidido modernizar su sistema de gestión de inventarios y quiere integrar tanto un sistema relacional para consultas rápidas como un sistema orientado a objetos para gestionar objetos de gran complejidad. Como parte de su equipo de desarrollo, tu tarea es crear un programa en Python que:

1. Conecte con MySQL utilizando Peewee (ORM) para crear una tabla `Libros` y gestione los datos relacionados con los libros que ya están catalogados.
2. Gestione una base de datos ZODB para almacenar objetos `Prestamo` que contengan información sobre los préstamos de cada libro a los lectores.
3. Asegure la integridad de datos mediante transacciones y controle las excepciones adecuadamente.

Pasos específicos a realizar:

1. Tabla de libros en MySQL:

- Crea una tabla llamada `Libros` con los siguientes campos:
 - `id` (entero, clave primaria, autoincremental)
 - `titulo` (varchar, máximo 100 caracteres)
 - `autor` (varchar, máximo 100 caracteres)
 - `anio_publicacion` (entero)
 - `genero` (varchar, máximo 50 caracteres)
- Inserta los 5 libros siguientes en la tabla. Asegúrate de que al menos uno de ellos sea de "Novela":

```
libros = [  
    {'titulo': 'Cien años de soledad', 'autor': 'Gabriel García Márquez', 'anio_publicacion':  
1967, 'genero': 'Novela'},  
    {'titulo': 'Don Quijote de la Mancha', 'autor': 'Miguel de Cervantes', 'anio_publicacion':  
1605, 'genero': 'Novela'},  
    {'titulo': 'El Principito', 'autor': 'Antoine de Saint-Exupéry', 'anio_publicacion': 1943,  
'genero': 'Infantil'},  
    {'titulo': 'Crónica de una muerte anunciada', 'autor': 'Gabriel García Márquez',  
'anio_publicacion': 1981, 'genero': 'Novela'},  
    {'titulo': '1984', 'autor': 'George Orwell', 'anio_publicacion': 1949, 'genero': 'Distopía'}  
]
```

2. Base de datos ZODB para los préstamos:

- Define una clase `Prestamo` que contenga los siguientes atributos:
 - `libro_id` (entero que coincida con un id de libro en MySQL)
 - `nombre_usuario` (nombre de la persona que tomó el préstamo)
 - `fecha_prestamo` (fecha en formato `yyyy-mm-dd`)
 - `fecha_devolucion` (fecha en formato `yyyy-mm-dd`)
- Crea y almacena en ZODB tres objetos de tipo `Prestamo` con estos datos:

```
(1, 'Juan Perez', '2023-10-01', '2023-11-01'),  
(2, 'Ana Lopez', '2023-09-15', '2023-10-15'),  
(4, 'Maria Gomez', '2023-09-20', '2023-10-20')
```

3. Integración y consultas cruzadas: Desarrolla una función que permita buscar los préstamos por género del libro. Esta función debe:

- Consultar el género en MySQL.
- Retornar todos los préstamos en ZODB de los libros que pertenecen a ese género.

- Si el género no existe o si el préstamo no está disponible en ZODB, debe mostrar un mensaje apropiado.
- Su prototipo será:

```
def buscarprestamos_por_genero(genero):
```

4. Transacciones y manejo de excepciones: En la sección de inserción de préstamos en ZODB, implementa transacciones y gestión de excepciones. Si ocurre un error al crear un préstamo, deshaz la operación y muestra un mensaje de error.

Rúbrica

Tarea	Descripción	Puntos
Conexión a MySQL y creación de la tabla Libros	Configura correctamente la conexión a la base de datos MySQL y crea la tabla Libros en Peewee.	1
Inserción de registros en la tabla Libros	Inserta correctamente los registros en la tabla Libros en la base de datos MySQL.	2
Configuración de ZODB y creación de la clase Prestamo	Configura correctamente la base de datos ZODB y crea la clase Prestamo con los atributos requeridos.	2
Transacciones y manejo de excepciones en ZODB	Implementa transacciones y controla adecuadamente las excepciones durante la inserción de los préstamos en ZODB.	2,5
Consulta cruzada por género	Implementa la función que realiza la consulta cruzada entre MySQL y ZODB, filtrando préstamos por género del libro.	2
Mensajes de estado y salida clara	Proporciona mensajes claros de estado en cada paso de la ejecución, informando sobre las acciones realizadas.	0,5

Traza de ejemplo

A modo de guía y sin que sea obligatorio que tus mensajes sean como los que se muestran a continuación, se muestra una traza de ejemplo de la ejecución del programa:

Libros insertados correctamente en la base de datos.

Préstamos almacenados correctamente en ZODB.

Préstamos de libros del género 'Novela':

Libro: Cien años de soledad, Usuario: Juan Perez, Fecha Préstamo: 2023-10-01, Fecha Devolución: 2023-11-01

Libro: Don Quijote de la Mancha, Usuario: Ana Lopez, Fecha Préstamo: 2023-09-15, Fecha Devolución: 2023-10-15

Libro: Crónica de una muerte anunciada, Usuario: Maria Gomez, Fecha Préstamo: 2023-09-20, Fecha Devolución: 2023-10-20

