

PROGRAMACIÓN AVANZADA CON PYTHON

(CEFIRE CTEM)



Bases de datos con Python
Ejercicios voluntarios

Esta obra está sujeta a la licencia Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visitad

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.



Autora: María Paz Segura Valero (mpazprofe@gmail.com)

CONTENIDO

1. Introducción.....	2
2. Ejercicios voluntarios.....	2
2.1. Antes de empezar.....	2
2.2. Ejercicio 1: Crear la tabla LISTA_HERRAMIENTAS.....	3
2.3. Ejercicio 2: Poblar la tabla LISTA_HERRAMIENTAS.....	4
2.4. Ejercicio 3: Consultar la base de datos.....	5
2.4.1. Algunos consejos.....	6
2.5. Ejercicio 4: Jugadores.....	7
2.5.1. Algunos consejos.....	10

1. Introducción

En este documento puedes encontrar una serie de **ejercicios voluntarios** para que pongas en práctica lo aprendido durante esta unidad.

En el aula virtual dispondrás de las soluciones pero te recomiendo que intentes solucionarlos por ti mismo/a porque, aunque no sean obligatorios para superar el curso, sí pueden ayudarte a enfrentarte al *ejercicio obligatorio* del final.

Puedes utilizar el foro del curso para consultar tus dudas.

2. Ejercicios voluntarios

Para resolver los ejercicios de este boletín vamos a trabajar con la siguiente base de datos



La base de datos se compone de tres tablas (JUGADORES, HERRAMIENTAS y LISTA_HERRAMIENTAS) que sirven para almacenar los jugadores de un juego, las herramientas que pueden utilizar y la vida de cada jugador.

Como se puede ver, un jugador puede tener más de una herramienta en su poder y una herramienta puede pertenecer a más de un jugador. Por ejemplo, podría haber dos jugadores con un martillo y solo un jugador con una poción mágica.

2.1. Antes de empezar

Aunque la base de datos es la misma para todos los ejercicios del boletín, el estado en el que necesitamos que se encuentre es diferente en cada uno de ellos.


En la siguiente tabla puedes encontrar la explicación de los distintos ficheros de bases de datos que puedes descargar del aula virtual.

Fichero	Descripción
ejers_voluntarios_ejer1.db	La base de datos en el estado correcto para resolver el ejercicio 1. Contiene las tablas pobladas JUGADORES, HERRAMIENTAS.
ejers_voluntarios_ejer2.db	La base de datos en el estado correcto para resolver el ejercicio 2. Contiene la definición de las tres tablas pero LISTA_HERRAMIENTAS

	no tiene registros.
ejers_voluntarios_ejer3.db	La base de datos en el estado correcto para resolver el ejercicio 3. Contiene las tablas pobladas JUGADORES, HERRAMIENTAS y LISTA_HERRAMIENTAS.
ejers_voluntarios_ejer4.db	La base de datos en el estado correcto para resolver el ejercicio 4. Contiene las tablas pobladas JUGADORES, HERRAMIENTAS y LISTA_HERRAMIENTAS.

Ahora, deberías preparar las bases de datos con las que vas a ir trabajando en los ejercicios. Puedes hacerlo de otra forma, pero te dejo aquí mi sugerencia:

- Crea una carpeta nueva en tu ordenador para los ejercicios voluntarios y otra llamada **bd** dentro de la primera.

 ejercicios voluntarios

 bd

- Descarga del aula virtual los ficheros de bases de datos y ubícalos en la carpeta **bd**.
- Copia también el fichero **sqlite3.exe** en la carpeta **bd**. Así tu base de datos y el terminal estarán en la misma carpeta y no tendrás que ir introduciendo rutas para abrir los ficheros desde el terminal.
- Recuerda cargar la base de datos correspondiente (comando `.open`) cuando vayas a trabajar con el terminal **sqlite3.exe**.

2.2. Ejercicio 1: Crear la tabla LISTA_HERRAMIENTAS

Diseña una sentencia SQL que cree la tabla LISTA_HERRAMIENTAS utilizando el fichero de bases de datos **ejers_voluntarios_ejer1.db**. No es necesario que escribas un programa de Python. Simplemente crea la sentencia y ejecútala para ver que funciona.

Deberás tener en cuenta los siguientes puntos:

- Los dos campos son de tipo `TEXT`.
- La clave primaria de la tabla son los dos campos que contiene.
- El campo `jugador` referencia al campo `nombre` de la tabla `JUGADORES`.
- El campo `herramienta` referencia al campo `nombre` de la tabla `HERRAMIENTAS`.

Cuando hayas acabado, ejecuta el comando `.schema LISTA_HERRAMIENTAS` y sube una captura de pantalla al aula virtual.

En el aula virtual dispones de la solución del ejercicio en el fichero **ud2_ejer1_crear_tabla.sql**.

2.3. Ejercicio 2: Poblar la tabla LISTA_HERRAMIENTAS

Crea un programa llamado **ud2_ejer2_poblar_tabla.py** que se encargará de insertar distintos registros en la tabla LISTA_HERRAMIENTAS.

Recuerda que deberás crear una conexión a la base de datos **ejers_voluntarios_ejer2.db** y utilizar un cursor para ejecutar las instrucciones.

Ten en cuenta los siguientes puntos:

- Crea el siguiente registro utilizando la función **execute()** sin parámetros.

```
jugador  herramienta
-----  -
ana      alas
```

- Crea el siguiente registro utilizando la función **execute()** pasando como parámetro el valor de los dos campos.

```
jugador  herramienta
-----  -
ana      botines
```

- Crea los siguientes registros utilizando la función **executemany()** y pasando la lista de tuplas correspondiente.

```
jugador  herramienta
-----  -
ana      rastrillo
pep      alas
pep      varita
mary     varita
mary     esencia
mary     martillo
quico    botines
quico    rastrilo
sandy    varita
```

- Recuerda ejecutar la función **commit()** para que los cambios se hagan permanentes.

En el aula virtual dispones de la solución del ejercicio en el fichero **ud2_ejer2_poblar_tablas.py**.

2.4. Ejercicio 3: Consultar la base de datos

Vamos a crear un programa que permita consultar la información almacenada en la base de datos **ejers_voluntarios_ejer3.db**. Al principio del programa habrá que crear la conexión contra la base de datos y el cursor que se va a utilizar para manejar las instrucciones SQL.

El programa mostrará el siguiente menú:

```
=====
                        CONSULTAR BASE DE DATOS
=====
1 - Imprimir jugadores
2 - Imprimir herramientas-jugador
3 - Consultar jugador
4 - Consultar herramienta-jugador
0 - SALIR
-----
Dame la opción:
```

Cada opción seleccionada ejecutará una nueva función que responderá a la tarea.

Opción	Función ejecutada
1	<p>imprimir_jugadores()</p> <p>Esta función leerá todos los registros de la tabla JUGADORES y mostrará la información por pantalla.</p> <p><u>Ejemplo:</u></p> <pre>IMPRIMIR JUGADORES Nombre: ana - Vida: 100 Nombre: pep - Vida: 120 Nombre: mary - Vida: 80 Nombre: quico - Vida: 100 Nombre: sandy - Vida: 80</pre> <p>A la función se le pasará como parámetro de entrada el cursor a la base de datos y no devolverá ningún parámetro de salida.</p>
2	<p>imprimir_herramientas_jugador()</p> <p>Esta función pedirá el nombre de un jugador y mostrará por pantalla todas las herramientas que tiene asociadas. Para ello, consultará la tabla LISTA_HERRAMIENTAS.</p> <p><u>Ejemplo:</u></p> <pre>HERRAMIENTAS JUGADOR Dame el nombre del jugador: ana Herramientas: alas - botines - rastrillo -</pre> <p>A la función se le pasará como parámetro de entrada el cursor a la base de datos y no devolverá ningún parámetro de salida.</p>

3	<p>consultar_jugador()</p> <p>Esta función pedirá por pantalla el nombre del jugador y mostrará sus datos. Para ello, se consultará la tabla JUGADORES.</p> <p><u>Ejemplo:</u></p> <pre>CONSULTAR JUGADOR Dame el nombre del jugador: ana Nombre: ana - Vida: 100</pre> <p>A la función se le pasará como parámetro de entrada el cursor a la base de datos y no devolverá ningún parámetro de salida.</p>
4	<p>consultar_herramienta_jugador()</p> <p>Esta función pedirá por pantalla el nombre de un jugador y de una herramienta y comprobará si dicho jugador tiene asociada dicha herramienta. Para ello, se consultará la tabla LISTA_HERRAMIENTAS.</p> <p><u>Ejemplo 1:</u> El jugador sí tiene asociada la herramienta.</p> <pre>CONSULTAR HERRAMIENTA-JUGADOR Dame el nombre del jugador: ana Dame herramienta: botines El jugador <ana> sí tiene la herramienta <botines></pre> <p><u>Ejemplo 2:</u> El jugador no tiene asociada la herramienta.</p> <pre>CONSULTAR HERRAMIENTA-JUGADOR Dame el nombre del jugador: ana Dame herramienta: varita El jugador <ana> no tiene la herramienta <varita></pre> <p>A la función se le pasará como parámetro de entrada el cursor a la base de datos y no devolverá ningún parámetro de salida.</p>
0	El programa acabará.

Cuando se acabe de ejecutar una acción, el programa volverá a mostrar el menú al usuario, excepto si se ha seleccionado la *opción 0*.

2.4.1. Algunos consejos

Puedes crear el programa desde cero o basarte en el esquema que tienes en el fichero **ud2_ejer3_consultar_bd (ESQUEMA).py** del aula virtual. En este caso, deberás cambiar las instrucciones *pass* que encuentres en el código.

En el aula virtual dispones de la solución del ejercicio en el fichero **ud2_ejer3_consultar_bd.py**.

2.5. Ejercicio 4: Jugadores

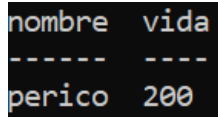
Vamos a crear un programa que permita gestionar alguna información de la base de datos **ejers_voluntarios_ejer4.db**. Al principio del programa habrá que crear la conexión contra la base de datos y el cursor que se va a utilizar para manejar las instrucciones SQL.

El programa mostrará el siguiente menú:

```
=====
                        GESTIONANDO JUGADORES
=====
1 - Imprimir jugadores
2 - Modificar vida de jugador
3 - Insertar nuevo jugador
4 - Imprimir herramientas-jugador
5 - Quitar herramienta-jugador
0 - SALIR
-----
Dame la opción:
```

Cada opción seleccionada ejecutará una nueva función que responderá a la tarea.

Opción	Función ejecutada				
1	<p>imprimir_jugadores()</p> <p>Esta función leerá todos los registros de la tabla JUGADORES y mostrará la información por pantalla.</p> <p><u>Ejemplo:</u></p> <pre>IMPRIMIR JUGADORES Nombre: ana - Vida: 100 Nombre: pep - Vida: 120 Nombre: mary - Vida: 80 Nombre: quico - Vida: 100 Nombre: sandy - Vida: 80</pre> <p>A la función se le pasará como parámetro de entrada el cursor a la base de datos y no devolverá ningún parámetro de salida.</p>				
2	<p>modificar_vida()</p> <p>Esta función pedirá por pantalla el nombre del jugador y la vida (número entero) y actualizará el campo VIDA de la tabla JUGADORES para aquel registro cuyo campo NOMBRE coincida con el introducido por pantalla.</p> <p><u>Ejemplo 1:</u> El jugador existe en la base de datos.</p> <pre>MODIFICAR VIDA Dame jugador: ana Dame vida: 200 La vida del jugador <ana> se ha modificado correctamente</pre> <div data-bbox="1149 1765 1388 1881"> <table> <thead> <tr> <th>nombre</th> <th>vida</th> </tr> </thead> <tbody> <tr> <td>ana</td> <td>200</td> </tr> </tbody> </table> </div> <p>La función deberá comprobar que el jugador existe en la base de datos antes de</p>	nombre	vida	ana	200
nombre	vida				
ana	200				

	<p>intentar modificarlo y, si no es así, mostrar un mensaje de error. Para ello se creará un función nueva llamada existe_jugador() que se explica más adelante.</p> <p><u>Ejemplo 2:</u> El jugador no existe en la base de datos.</p> <pre>MODIFICAR VIDA Dame jugador: kk ERROR: El jugador no existe</pre> <p>A la función se le pasará como parámetro de entrada la conexión a la base de datos y el cursor pero no devolverá ningún parámetro de salida.</p>
3	<p>insertar_jugador()</p> <p>Esta función leerá los elementos de la lista del programa y, cada elemento, lo almacenará en una línea distinta del fichero. Deberás añadir el carácter de fin de línea ("\\n") antes de escribir cada línea en el fichero.</p> <p><u>Ejemplo 1:</u> Añadimos un nuevo jugador a la base de datos.</p> <pre>INSERTAR JUGADOR Dame nombre: perico Dame vida: 200 El jugador <perico> se ha insertado correctamente</pre>  <p>La función deberá comprobar que el jugador no existe en la base de datos antes de intentar modificarlo y, si no es así, mostrar un mensaje de error. Para ello se creará un función nueva llamada existe_jugador() que se explica más adelante.</p> <p><u>Ejemplo 2:</u> El jugador ya existe en la base de datos.</p> <pre>INSERTAR JUGADOR Dame nombre: ana ERROR: El jugador ya existe</pre> <p>A la función se le pasará como parámetro de entrada la conexión a la base de datos y el cursor pero no devolverá ningún parámetro de salida.</p>
4	<p>imprimir_herramientas_jugador()</p> <p>Esta función imprimirá la lista de herramientas que posee el jugador leído por pantalla. Para ello, consultará la tabla LISTA_HERRAMIENTAS de la base de datos.</p> <p><u>Ejemplo 1:</u> Buscamos las herramientas de un jugador existente.</p> <pre>HERRAMIENTAS JUGADOR Dame el nombre del jugador: ana Herramientas: alas - botines - rastrillo -</pre> <p>La función deberá comprobar que el jugador existe en la base de datos antes de intentar modificarlo y, si no es así, mostrar un mensaje de error. Para ello se creará un función nueva llamada existe_jugador() que se explica más adelante.</p> <p><u>Ejemplo 2:</u> El jugador no existe en la base de datos.</p>

	<p>HERRAMIENTAS JUGADOR Dame el nombre del jugador: <code>kk</code> ERROR: El jugador no existe</p> <p>A la función se le pasará como parámetro de entrada el cursor pero no devolverá ningún parámetro de salida.</p>
5	<p>quitar_herramienta() Esta función pedirá un nombre de jugador y un nombre de herramienta por pantalla y eliminará dicha herramienta de la lista de herramientas de dicho jugador. Para ello, utilizará la tabla LISTA_HERRAMIENTAS de la base de datos.</p> <p><u>Ejemplo 1:</u> Quitamos una herramienta de un jugador existente. <pre> QUITAR HERRAMIENTA Dame el nombre del jugador: pep Dame herramienta: alas Se ha eliminado la herramienta <alas> del jugador <pep> correctamente </pre></p> <p>La función deberá comprobar que el jugador existe en la base de datos antes de intentar modificarlo y, si no es así, mostrar un mensaje de error. Para ello se creará un función nueva llamada existe_jugador() que se explica más adelante.</p> <p><u>Ejemplo 2:</u> El jugador no existe en la base de datos. <pre> QUITAR HERRAMIENTA Dame el nombre del jugador: kk ERROR: El jugador no existe </pre></p> <p>La función deberá comprobar que el jugador tiene asignada dicha herramienta y, si no es así, mostrará un mensaje de error. Para ello se creará un función nueva llamada existe_herramienta_jugador() que se explica más adelante.</p> <p><u>Ejemplo 3:</u> El jugador no tiene asociada la herramienta. <pre> QUITAR HERRAMIENTA Dame el nombre del jugador: pep Dame herramienta: capa ERROR: El jugador no tiene asignada la herramienta <capa> </pre></p> <p>A la función se le pasará como parámetro de entrada la conexión a la base de datos y el cursor pero no devolverá ningún parámetro de salida.</p>
0	El programa acabará.

Cuando se acabe de ejecutar una acción, el programa volverá a mostrar el menú al usuario, excepto si se ha seleccionado la *opción 0*.

En esta tabla se describen las funciones adicionales que debes programar:

Función	Descripción
existe_jugador()	<p>Esta función comprobará si el jugador existe en la tabla JUGADORES pero no mostrará nada por pantalla.</p> <p>A la función se le pasará como parámetro de entrada el cursor de la base de datos y el nombre del jugador y devolverá como parámetro de salida un valor lógico: True si el jugador existe en la tabla JUGADORES y False en otro caso.</p>
existe_herramienta_jugador()	<p>Esta función comprobará si en la tabla LISTA_HERRAMIENTAS hay algún registro donde se asocie la herramienta al jugador pero no mostrará nada por pantalla.</p> <p>A la función se le pasará como parámetro de entrada el cursor de la base de datos y el nombre del jugador y devolverá como parámetro de salida un valor lógico: True si el jugador existe en la tabla JUGADORES y False en otro caso.</p>

2.5.1. Algunos consejos

En algunos casos hará falta manejar excepciones y, en otros, solo será necesario imprimir un aviso al usuario.

.....

Puedes crear el programa desde cero o basarte en el esquema que tienes en el fichero **ud2_ejer4_jugadores (ESQUEMA).py** del aula virtual. En este caso, deberás cambiar las instrucciones *pass* que encuentres en el código.

En el aula virtual dispones de la solución del ejercicio en el fichero **ud2_ejer4_jugadores.py**.