

PROGRAMACIÓN AVANZADA CON PYTHON

(CEFIRE CTEM)



Bases de datos con Python Ejercicio obligatorio

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. Enunciado.....	2
2.1.1. Algunos consejos.....	5

1. Introducción

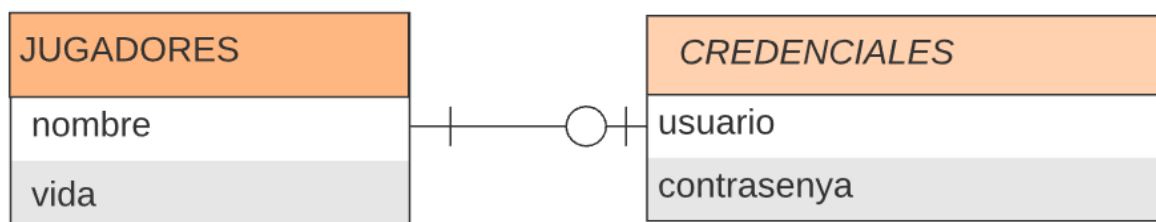
En este documento puedes encontrar el **ejercicio obligatorio** de esta unidad. Es imprescindible entregarlo en tiempo y forma para superar esta parte del curso.

Tendrás la oportunidad de realizar la entrega de varias versiones del ejercicio hasta que consigas superarlo y la profesora te indicará en cada corrección las mejoras necesarias.

En cualquier momento puedes lanzar preguntas al **foro** del curso o realizar una **entrega parcial** del ejercicio acompañada de una **lista de dudas** para que la profesora pueda orientarte en su resolución.

2. Enunciado

Vamos a escribir un programa que gestionará las credenciales de los jugadores de la base de datos **ejer_obligatorio.db** cuyo diagrama E-R es el siguiente:



Como se puede observar, un jugador puede no tener asociada una credencial pero si la tiene, solo tendrá una. Y una credencial pertenecerá obligatoriamente a un jugador.

El programa mostrará el siguiente menú:

```

=====
          CREDENCIALES JUGADORES
=====
1 - Imprimir jugadores
2 - Validar credencial
3 - Cambiar credencial
4 - Insertar nuevo jugador
5 - Borrar jugador
0 - SALIR
-----
Dame la opción:
    
```

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

Opción	Función ejecutada
1	imprimir_jugadores() Esta función leerá todos los registros de la tabla JUGADORES y mostrará la información por pantalla.

	<p>Ejemplo:</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>validar_credencial()</p> <p>Esta función comprueba si un par usuario-contraseña existe en la tabla CREDENCIALES. Para ello, pedirá los datos por pantalla.</p> <p>Ejemplo 1: La credencial existe.</p> <pre>VALIDAR CREDENCIAL Dame jugador: ana Dame contraseña: 111 [ana/111] sí es una credencial válida</pre> <p>Ejemplo 2: La credencial no existe.</p> <pre>VALIDAR CREDENCIAL Dame jugador: ana Dame contraseña: hola [ana/hola] no es una credencial válida</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>cambiar_credencial()</p> <p>Esta función cambiará la contraseña de un usuario. Para ello, pedirá por pantalla el usuario, la contraseña vieja y una contraseña nueva. Si todos los datos son correctos, actualizará la tabla CREDENCIALES.</p> <p>Ejemplo 1: Se cambia la credencial correctamente</p> <pre>CAMBIAR CREDENCIAL Dame jugador: ana Dame contraseña vieja: 111 Dame contraseña nueva: 999 La contraseña del jugador <ana> se ha modificado correctamente</pre> <p>La función deberá comprobar que el jugador existe en la base de datos y, si no existe, mostrará un mensaje de error. Para ello se creará una función nueva llamada existe_jugador() que se explica más adelante.</p> <p>Ejemplo 2: El jugador no existe en la base de datos.</p> <pre>CAMBIAR CREDENCIAL Dame jugador: kk ERROR: El jugador no existe</pre>

	<p>La función deberá comprobar que la credencial introducida por teclado (usuario, contraseña) jugador existe en la base de datos y, si no existe, mostrará un mensaje de error. Para ello se creará un función nueva llamada existe_credencial() que se explica más adelante.</p> <p><u>Ejemplo 3:</u> El usuario existe pero la credencial introducida es incorrecta.</p> <pre>CAMBIAR CREDENCIAL Dame jugador: ana Dame contraseña vieja: hola [ana/hola] no es una credencial válida</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>insertar_jugador()</p> <p>Esta función pedirá por teclado los datos de un nuevo jugador y, si no existe, lo dará de alta en la base de datos. Para ello, deberá actualizar los datos de la tabla JUGADORES y la tabla CREDENCIALES.</p> <p><u>Ejemplo 1:</u> Todos los datos son correctos y se da de alta el nuevo jugador.</p> <pre>INSERTAR JUGADOR Dame nombre: perico Dame vida: 95 Dame contraseña: 666 El jugador <perico> se ha insertado correctamente</pre> <p><u>Ejemplo 2:</u> El usuario ya existía en la base de datos.</p> <pre>INSERTAR JUGADOR Dame nombre: perico 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>
5	<p>borrar_jugador()</p> <p>Esta función pedirá un nombre de jugador por teclado y borrará todos los registros relacionados de las tablas JUGADORES y CREDENCIALES.</p> <p><u>Ejemplo 1:</u> El jugador existe y se borra correctamente</p> <pre>BORRAR JUGADOR Dame el nombre del jugador: perico Se ha eliminado el jugador <perico> correctamente</pre> <p><u>Ejemplo 2:</u> El jugador no existe</p> <pre>BORRAR JUGADOR Dame el nombre del 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</p>

	y el cursor pero no devolverá ningún parámetro de salida.
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_credencial()	<p>Esta función comprobará si existe la credencial en la tabla CREDENCIALES dado un nombre de usuario y contraseña 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 y contraseña 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>

2.1.1. 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_ejercicio_obligatorio (ESQUEMA).py** del aula virtual. En este caso, deberás sustituir las instrucciones *pass* por las instrucciones adecuadas para que funcione el programa según el enunciado.