



UF2: SISTEMAS ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Tema 2. Implantación de sistemas ERP-CRM en una empresa

Módulo MP10. Sistemas de gestión empresarial

Continuación

RESUMEN SESIÓN ANTERIOR

Conceptos vistos en la sesión anterior

- **Implantación de sistemas ERP-CRM en una empresa**

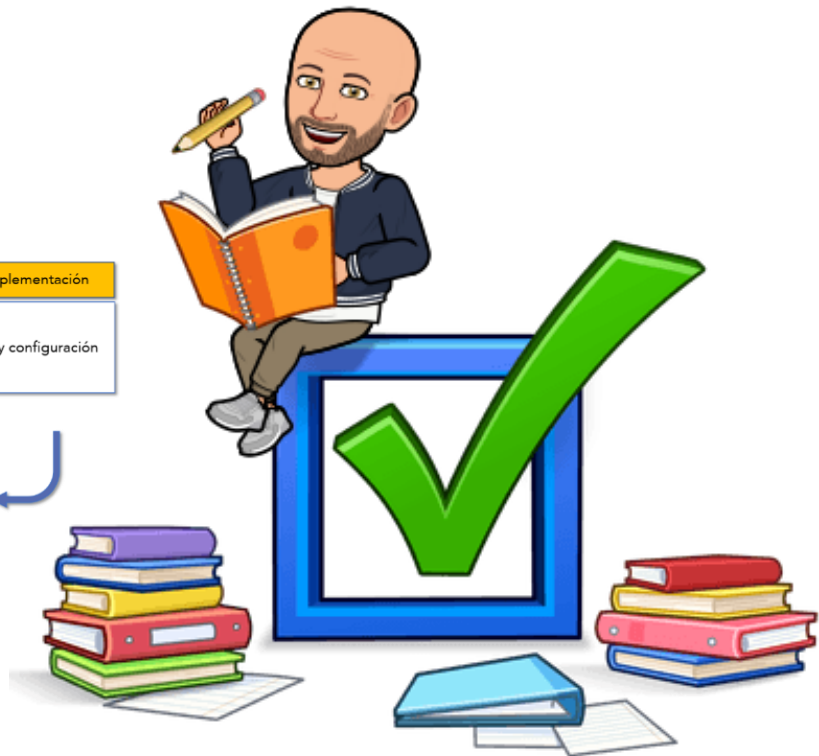
- **Metodologías de implantación**

- Ciclo de vida

- Fases en la Implantación

- INICIACIÓN
 - DESARROLLO
 - IMPLEMENTACIÓN
 - IMPLANTACIÓN
 - PRODUCCIÓN Y SOPORTE

- **Necesidades de la empresa**



ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Selección de los módulos del sistema ERP-CRM

- **Pasos para crear un módulo**

- **Pasos para crear un módulo son:**

- 1. Dentro del directorio adons que se encuentra dentro de la instalación del Server, se crea un nuevo directorio para el módulo: mimodulo.
- 2. A continuación, se añaden los ficheros obligatorios.
 - `_init_.py`
 - `_terp_py`
- 3. Crear el fichero de definición del objeto mimodulo.py vacío.
- 4. Se añade el fichero mimodulo_view.xml, que va a contener los diferentes datos para el sistema.
- 5. Una vez programado el contenido de los ficheros, se debe abrir una sesión con el servidor
 - Habrá que desplazarse hasta Administración/Módulos para actualizar la lista de módulos.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Tablas y vistas a adaptar

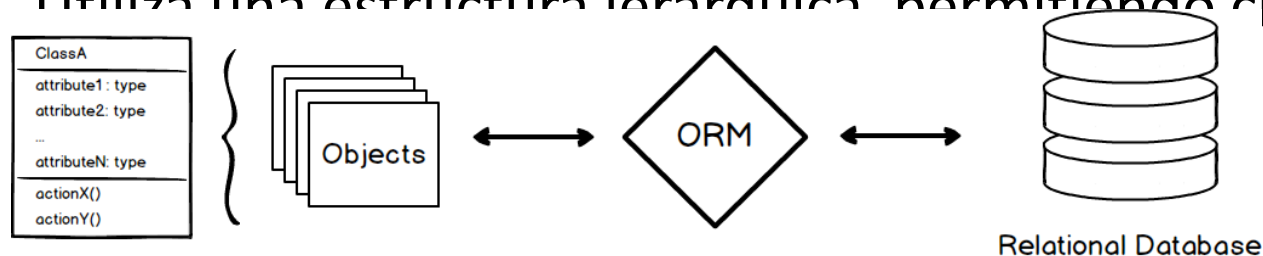
- **Una vez que se detecte el módulo, ya se puede instalar. Tipos de fichero:**
 - **1. El fichero `_init_.py`:** Se utilizará para cargar en el sistema las definiciones de los objetos.
 - **2. El fichero `_terp_.py`:** Define un diccionario anónimo con las propiedades ya definidas, que será utilizado por el sistema para determinar aquellos ficheros XML que se deben seguir tratando.
 - Ficheros como `init_XML` que es una lista de ficheros XML que se van a utilizar cuando se lance el servidor.
 - **3. Fichero `mimodulo.py`:** Mediante este fichero se pueden determinar aquellos objetos nuevos junto con sus propiedades
 - El formato de este fichero es de Python.
 - **4. Fichero `mimodulo_view.xml`:** Este fichero permite determinar los distintos componentes que se deben crear para conseguir que el módulo se integre con el sistema.
 - Un módulo está formado por vistas, gráficos, informes, asistentes, flujos de trabajo, menús y acciones.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Consultas necesarias para obtener información

- **¿Que es el Mapeo Objeto-Relacional?**

- Odoo utiliza la técnica ORM (Mapeo relacional de objetos) para interactuar con la base de datos. ORM ayudará a crear una base de datos de objetos virtuales que se puede usar desde Python. En la técnica ORM, cada modelo está representado por una clase que subclasifica Models.model.
- Models.model es la súper clase principal para los modelos de Odoo de base de datos regulares. Los modelos Odoo se crean heredando de esta clase.
- Utiliza una estructura jerárquica, permitiendo crear un objeto en cada nivel.



ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Tablas y vistas a adaptar

- **Modelo-Vista-Controlador**

- Se usa para realizar la gestión de datos. Cuando se separan los datos de la presentación, se consigue independencia para realizar la implementación de los distintos entornos de acceso al servidor.
- El MVC divide una aplicación en tres componentes:
 - Los datos de la aplicación (modelo).
 - La interfaz del usuario (vista).
 - El controlador, el cual define la forma en que la interfaz reacciona a la entrada del usuario.
- En Odoo, el MVC se implementa de la siguiente forma:
 - El modelo son las tablas de la base de datos. Vienen representadas por una clase en Python.
 - Las vistas son los archivos XML que definen la interfaz de usuario del modulo.
 - El controlador son los objetos creados en Python.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Tablas y vistas a adaptar

- **Motor de un ERP**
 - La estructura que se utiliza para almacenar información es la misma que en bases de datos, tablas.
 - Una herramienta ERP tiene como motor una base de datos, ya que es la mejor manera de organizar, relacionar y almacenar los datos con los que después se va a trabajar.
- **Visualización por pantalla**
 - Cuando llega el momento de diseñar los diferentes modelos de representación, se deben determinar cómo se va a realizar la visualización por pantalla.
 - Por tanto, es posible diferenciar entre dos tipos de vistas principales, más otra de apoyo:
 - Vista formulario, Vista en árbol, Vista de búsqueda, Vista de gráfico

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Tablas y vistas a adaptar

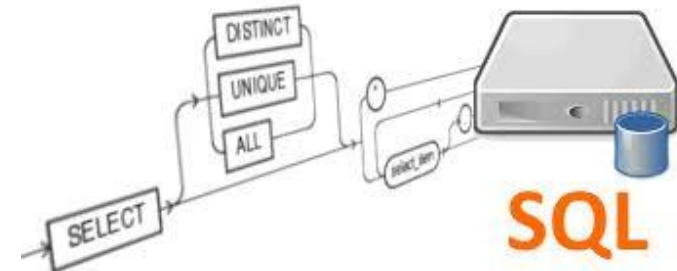
- **En la definición de vista.**
 - **Algunos de los elementos que se necesitan para crear la estructura adecuada visualizarla son:**
 - **<newline />**
 - Permite crear un salto de línea en la visualización, que obliga al siguiente control a pasar a la siguiente fila.
 - **<label string="Texto" />**
 - Permite añadir una etiqueta con su texto correspondiente.
 - **<field onchange="funcion">**
 - Nombre y parámetros de la función que hay que llamar cuando cambia el valor del campo.
 - **<field select="1">**
 - Cuando se desee que el campo actual sea un campo que se utilice en búsquedas, debe tener un índice.
 - **<page string="">**
 - Permite organizar en pestañas el contenido de un notebook.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Consultas necesarias para obtener información

- **Sistema de consultas**

- Este sistema ERP, al estar basado en motores de Bases de datos, presenta la misma dinámica de trabajo que un sistema gestor de Bases de Datos.
- Una de las ventajas que presenta este sistema es que, al estar la información relacionada entre sí, es posible consultar datos referidos a tablas relacionadas.
- Se sacará más jugo a la información almacenada, hasta tal punto de conocer algunos puntos importantes y ponerlos en práctica en el futuro.



ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Creación de formularios personalizados

- **La interfaz gráfica**

- Está destinada a visualizar los datos de una manera clara y atractiva.
- El usuario final pueda ver la información de una forma más atractiva y dinámica.
- Se cuenta con un asistente que ayudará a diseñar el formulario y todos los campos que necesite dicho formulario, e incluso a insertar un subformulario dentro de otro.



a crear formularios estará diseñada para que, de una realizando los formularios

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Creación de informes personalizados

- **Los informes**

- Son una serie de nuevas vistas que se han creado para almacenar la información correspondiente a la base de datos, ofreciendo la posibilidad de una gestión más cómoda y práctica.
- Los informes no incorporan estructuras novedosas, utilizan las existentes, aumentando su eficiencia.
- Pasos que se deben seguir para crear una nueva vista:
 - 1. Configurar el sistema (opción de Informes avanzados)
 - 2. Crear una nueva vista (informe).
 - 3. Fijar aquellos campos que se van a utilizar en la pestaña Campos a mostrar
 - 4. Se pueden establecer filtros sobre alguno de los campos seleccionados.
 - 5. Finalizar el diseño del informe, seleccionar la opción de Guardar.
 - 6. Lanzar la vista, pestaña Configuración general a través del botón Abrir informe.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Creación de informes personalizados

- **Creación de informes impresos**

- Se tienen una serie de mecanismos que se pueden utilizar para crear informes imprimibles, haciendo uso de la información almacenada en la base de datos.
- El motor por defecto que usaba OpenERP es JasperReports, este motor permite imprimir informes a partir de ficheros RML. Un fichero RML es un fichero XML que define unos datos y una estructura para que el motor de OpenReport pueda transformarlo en un informe HTML o PDF
- Entre las opciones más conocidas para la creación de informes están
 - JasperReports
 - Biblioteca de creación de informes. Entrega contenido enriquecido al monitor, a la impresora o a ficheros PDF
 - Aeroo
 - Programación directa de RML
 - OpenOffice.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Creación de cuadros de mando personalizados

- **Ventajas de una herramienta de código libre**
 - Está generalizada a cualquier tipo de empresa, ya que cubre las necesidades empresariales más generales.
 - Se pueden desarrollar a partir de ella, una interfaz y una base de datos más acorde a las necesidades específicas de cada corporación
 - Poder tener un software realizado a nuestra medida con muy poco desarrollo
 - La creación de cuadros de mando personalizados poseen un gran auge a la hora de implantarla en el ámbito empresarial
 - La puesta en marcha es más rápida y sin coste, ya que el periodo de implantación en la empresa no tendría coste alguno, ya que está adaptado 100% al protocolo de actuación laboral.



UF2: SISTEMAS ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Tema 3. Desarrollo de componentes

Módulo MP10. Sistemas de gestión empresarial

¿QUÉ VAMOS A VER?



ÍNDICE TEMA 3

- **Desarrollo de componentes**
 - **Lenguaje proporcionado por los ERP-CRM.**
 - **Características del lenguaje**
 - **Entornos de desarrollo**
 - **Herramientas de desarrollo**
 - **Llamadas a funciones**
 - **Depurar, manejar errores, etc**

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Desarrollo de componentes



- **El lenguaje Python (versión 3.7.)**
 - Se escoge este lenguaje porque es el que se usa para crear módulos en Odoo.
 - Es imprescindible el conocimiento de este lenguaje de programación, así como poder programar diferentes módulos para el sistema en cuestión. Si se quieren realizar nuevos módulos
- El tema que veremos se divide en dos partes:
 - 1) El lenguaje Python. Aunque no se entrará en detalles muy específicos.
 - 2) La creación de módulos propios que se pueden integrar en Odoo.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Desarrollo de componentes

- **Paso previo al desarrollo y programación de módulos**
 - Analizar el módulo para definir el diseño de los datos:
 - Su almacenamiento
 - Las distintas relaciones
 - Los posibles procedimientos
 - etc
 - Para este análisis, se van a utilizar las técnicas y estándares necesarios para asegurarse de qué es lo que se debe hacer y cómo hacerlo.



ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Lenguaje de programación Python**

- Este lenguaje de programación surge alrededor de los años 90, con la intención de poder programar en los distintos servidores web de forma limpia y multiplataforma.
- Este lenguaje, en principio, pretendía ser interpretado y orientado a objetos, añadiendo los paradigmas de la programación tradicional.

- **¿Qué significa lenguaje interpretado?**

- Un lenguaje de programación interpretado es un lenguaje en el que el código fuente se ejecuta directamente, instrucción a instrucción, sin ser compilado a diferencia de otros tipos de lenguajes.
- Es decir, el código no pasa por un proceso de compilación, sino que tenemos un programa llamado intérprete que lee la instrucción en tiempo real, y la ejecuta.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Las características de este lenguaje, entre otras son**
 - Es software libre y esto implica que puedas estudiarlo y crear derivados.
 - Es multiplataforma, lo cual te permite programar y ejecutar Python en Windows, Linux o MacOS.
 - Multiparadigma, ya que Python adopta los paradigmas de programación orientado a objetos (POO), programación imperativa y programación funcional.
 - Lenguaje interpretado, lo que nos permite programar scripts y ejecutar código Python, sin necesidad de compilarlo.
 - Sintaxis simple, cuando alguien conoce varios lenguajes de programación se agradece que la sintaxis y en general el formato del código sea sencillo y legible.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **¿Por dónde empezar?**

- Si se quiere comenzar a programar hay varias herramientas que podemos utilizar:
 - IDLE: es el intérprete oficial que se puede descargar de la página python.org.
 - Atom: Es un editor ligero desarrollado por Github y perfectamente integrado con este servicio, de modo que lo convierte en el más recomendable para los que quieren tener su código en estos repositorios.
 - Pycharm: Es una herramienta que no es un simple editor de texto, si no un entorno de desarrollo integrado (IDE), pensado específicamente para Python y desarrollado por JetBrains.
- En el lenguaje Python no existen mecanismos para definir bloques explícitos, sino que se van creando con la implementación del propio código.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Tipos de datos básicos de Python**

- En Python podemos encontrar distintos tipos de datos. Los tipos de datos básicos de Python son los booleanos, los numéricos (enteros, punto flotante y complejos) y las cadenas de caracteres.
- Python también define otros tipos de datos, entre los que se encuentran:
 - Secuencias: Los tipos list, tuple y range
 - Mapas: El tipo dict
 - Conjuntos: El tipo set
 - Iteradores
 - Clases
 - Instancias
 - Excepciones

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Impresión por pantalla**

- Para imprimir por pantalla se utiliza una variable denominada cadena de formato.
- La función que se utiliza para mostrar por pantalla es la función **print**, a la que se le pasa una cadena que determinará la forma en la que se desean imprimir las variables. Se especifica su posición, tipo y formato, con unos caracteres de formato y las distintas variables a utilizar.
- La función print, dentro de la impresión, tiene una serie de características:
 - Si se separan los valores por comas (,) se imprime, de forma automática, un espacio entre ellos.
 - Cuando se utiliza el operador de concatenación (+), será necesario añadir los espacios en el lugar necesario.
 - Habrá que convertir a cadena, mediante la función str(), aquellas variables que no sean de este tipo.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Tipos numéricos**

- Python define tres tipos de datos numéricos básicos:

- Enteros

- Lo más importante es saber cuál es el valor máximo según el tipo que se utilice y la plataforma en la que se ejecute.

- Números de coma flotante (simularía el conjunto de los números reales)

- Se usa para la representación de valores decimales, teniendo en cuenta la menor precisión que garantiza Python

para hacer comentarios.

```
1. >>> diez = 10
2. >>> diez_binario = 0b1010
3. >>> diez_octal = 0o12
4. >>> diez_hex = 0xa
5. >>> print(diez)
6. 10
7. >>> print(diez_binario)
8. 10
9. >>> print(diez_octal)
10. 10
11. >>> print(diez_hex)
12. 10
```

```
1. >>> real = 1.1 + 2.2 # real es un float
2. >>> print(real)
3. 3.3000000000000003 # Representación aproximada de 3.3
4. >>> print(f'{real:.2f}')
5. 3.30 # real mostrando únicamente 2 cifras decimales
```

```
#Tipo cadena de caracteres
hola = 'Hola "Universo"'
hola_2 = 'Hola \'Universo\''
hola_3 = "Hola 'Universo'"
print(hola)
print(hola_2)
print(hola_3)
```

```
Hola "Universo"
Hola 'Universo'
Hola 'Universo'
```

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Operadores numéricos básicos**

- Entre los operadores básicos se pueden encontrar:

- suma (+)
 - resta (-)
 - multiplicación (*)
 - división (/)
 - división entera (//)
 - exponenciación (**)
 - módulo (%).

Todos los elementos para Python son objetos, los tipos numéricos también.

Los números, además, cuentan con un conjunto de funciones que ofrecen el acceso a diferentes funciones matemáticas avanzadas, como pueden ser, entre otras:

- La raíz cuadrada (sqrt).
 - Logaritmos (log10).
 - ...

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Tipo booleano**

- En Python la clase que representa los valores booleanos es bool. (True or False).
Trabaja con expresiones condicionales. Define una serie de operando, que pueden ser:
 - Comparación y → (and)
 - Comparación o → (or)
 - Negación de una expresión → (not)
 - Igualdad → (==)
 - Desigualdad → (!=)
 - Mayor → (>)
 - Mayor igual → (>=)
 - Menor → (<)
 - Menor igual → (<=)

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Tipo cadena de caracteres**

- Son las secuencias o cadenas de caracteres. Conocido como string aunque su clase verdadera es str. Formalmente, un string es una secuencia inmutable de caracteres en formato Unicode.
- Para crear un string, simplemente tienes que encerrar entre comillas simples " o comillas dobles " o comillas triples """ de caracteres.

```
1. >>> hola = 'Hola "Pythonista"'
2. >>> hola_2 = 'Hola \'Pythonista\''
3. >>> hola_3 = "Hola 'Pythonista'"
4.
5. >>> print(hola)
6. Hola "Pythonista"
7.
8. >>> print(hola_2)
9. Hola 'Pythonista'
10.
11. >>> print(hola_3)
12. Hola 'Pythonista'
```

```
1. >>> caracter_a = 'a'
2. >>> print(caracter_a)
3. a
```

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Tipo cadena de caracteres**

- Mediante la barra invertida (\) puede introducir una serie de valores especiales, como:
 - \n → Para representar el carácter de una nueva línea.
 - \t → Para tabular.
 - \r → Representa el retroceso de carro.
- Permiten incorporar una serie de funciones bastante útiles, como pueden ser:
 - capitalize() → Devuelve una cadena con la primera letra mayúscula.
 - center(num) → Permite centrar una cadena según el número de caracteres que se le pasen por parámetro, haciendo uso de los caracteres en blanco.
 - ljust() y rjust() → Para justificar tanto a izquierda como a derecha.
 - count(subcadena) → Devuelve el número de veces que aparece la cadena pasada por parámetro.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Tipo cadena de caracteres**

- **Mas funciones útiles con cadenas**

- `find (subcadena)` → Devuelve la primera posición en la que aparece la cadena pasada por parámetro.
 - `upper()` → Para convertir a mayúsculas.
 - `strip ()` → Para eliminar los espacios en blanco.
 - `split (carácter)` → Permite dividir una cadena en distintas partes según el separador que se pase por parámetro.
 - `splitlines()` → Permite dividir en distintas líneas.
 - `len (cadena)` → Devuelve la longitud de un determinado objeto.
 - `join (cadena)` → Permite unificar dos cadenas.
 - `format (valores)` → Permite formatear la cadena de entrada en función de los valores pasados por parámetro.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Otros tipos**

- Todos ellos son tipos compuestos para agrupar juntos valores
 - **Las listas:**
 - Son secuencias mutables de valores.
 - **Las tuplas:**
 - Son secuencias inmutables de valores.
 - **Los conjuntos:**
 - Se utilizan para representar conjuntos únicos de elementos, es decir, en un conjunto no pueden existir dos objetos iguales.
 - **Los diccionarios:**
 - Tipos especiales de contenedores en los que se puede acceder a sus elementos a partir de una clave única.

```
1. >>> lista = [1, 2, 3, 8, 9]
2. >>> tupla = (1, 4, 8, 0, 5)
3. >>> conjunto = set([1, 3, 1, 4])
4. >>> diccionario = {'a': 1, 'b': 3, 'z': 8}
5. >>> print(lista)
6. [1, 2, 3, 8, 9]
7. >>> print(tupla)
8. (1, 4, 8, 0, 5)
9. >>> print(conjunto)
10. {1, 3, 4}
11. >>> print(diccionario)
12. {'a': 1, 'b': 3, 'z': 8}
```

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Características y sintaxis del lenguaje.



- **Las Variables**

- Las variables en **Python**, no necesitan ser definidas para su utilización .
- Cuando se tenga que hacer uso de una nueva variable, solo habrá que **asignar el valor deseado** y, a partir de ese momento, ya estará accesible.
- Se pueden diferenciar dos tipos de variables, dependiendo del ámbito en el que vayan a estar visibles:
 - **Globales:** tienen un determinado valor a lo largo de todo el programa, para las distintas funciones y clases definidas por el usuario.
 - **Locales:** solo tienen sentido dentro del bloque en el que se inicialicen, como, por ejemplo, dentro de una determinada función.

ERP-CRM. EXPLOTACIÓN Y ADECUACIÓN

Entrada/salida de datos - Variables - Tipos de datos

- #Programa que solicite al usuario que ingrese su nombre. El nombre se debe almacenar en una variable llamada nombre. Mostrar en pantalla el texto. “El nombre es...(nombre)”
 - nombre=input("Tu nombre:")
 - print("El nombre es: ", nombre)
- #Programa que solicite al usuario ingresar la cantidad de kilómetros recorridos por una motocicleta y la litros de combustible que consumió durante ese recorrido. Mostrar el consumo por kilómetro.
 - kilometros=float(input("Kilómetros recorridos:"))
 - litros=float(input("Litros de combustible gastados:"))
 - print("El consumo por kilómetro es de", kilometros/litros)



¿DUDAS?

