

SGE UT1 SISTEMAS ERP

Empresa

Entidad económica pública o privada que produce bienes y servicios. Esta integrada por recursos humanos, financieros, materiales y técnico-administrativo. Sus funciones principales son de dirección, finanzas, producción y recursos humanos.

Persona física y jurídica

- **Física** individuo humano responde con su patrimonio.
- **Jurídica** entidad (sociedad, corporación o fundación) responde con bienes propios.

Clientes y proveedores

- **Cliente** adquiere bienes y servicios a cambio de dinero.
- **Proveedor** suministra bienes y servicios a empresas o personas.

Empleados

Persona que presta servicios a cambio de un salario regulado por un contrato.

IVA

El IVA se aplica en todas las fases productivas, el IVA soportado en compras y el repercutido en ventas se compensan, de modo que el consumidor final es el que lo paga.

Facturas

Documento mercantil de compraventa. Los tipos de factura son: completa, simplificada (también llamada ticket), proforma (sirven como una oferta previa y no tienen valor contable).

Procesos productivos

- **Producción/fabricación** proceso por el cual se elabora un producto.
- **Línea de producción** operaciones secuenciales y recursos necesarios para organizar la fabricación de un producto.
- **Orden de producción** concreción de la fabricación de un producto con la información de cantidad, fecha y línea usada.

Logística e inventario

- **Logística** procesos de coordinación, gestión y transporte de bienes desde el lugar de distribución hasta el cliente final.

- **Almacenamiento** proceso que guarda la información de materiales y stock.
- **Inventario/stock** existencias para venta o producción.
- **Distribución** proceso por el cual la empresa hace llegar bienes y servicios al consumidor final.

Sistema de información (SI)

Conjunto de elementos que apoyan actividades de la empresa. Incluye hardware, software, personas y procedimientos.

Proceso de negocio

Conjunto de actividades que transforman entradas en salidas con valor.

Gestión empresarial

Actividad que busca mejorar productividad y competitividad. Integra funciones como marketing, finanzas, operaciones, calidad, recursos humanos, I+D y dirección.

Aplicación de SGE

Los SGE deben trabajar de manera unificada y sinérgica, para ello deben ser rápidos, tener calidad de datos, estar disponibles en tiempo real, tener información global y selectiva.

Estrategia empresarial

- **Misión** actividad principal de la empresa, a qué público objetivo va dirigido la empresa y que ventaja ofrece ante los competidores.
- **Visión** define la situación futura de la empresa, los objetivos a futuro deben ser realistas y motivadores.
- **Valores** qué principios éticos y cultura tiene la empresa.

Funciones empresariales

- **Gestión de personal** se encarga de coordinar la parte de recursos humanos.
- **Financiera** se encarga de definir cómo se usará el dinero, el rendimiento y la contabilidad de la empresa.
- **Marketing** se encarga de la investigación de mercados, el diseño y promoción de productos.
- **Operaciones compraventa** se encarga del proceso desde el pedido hasta el pago.
- **Innovación** se encarga del proceso de transformación de ideas en productos o servicios nuevos o métodos de organización.

Evolución de la informática de gestión

- **Años 60** sistemas contables y administrativos no integrados, los sistemas trabajan sin relación entre ellas.
- **Años 70 MRP** automatización de pedidos de materiales, reducción de stocks en almacén, mejora eficiencia.
- **Años 80 MRP II** se tiene en cuenta las necesidades de planificación de materiales y recursos de fabricación se inicia la integración de todos los sistemas
- **Años 90 ERP** integración total de procesos en un único sistema.
- **ERP II** evolución con Bussiness Intelligence ,IA y CRM automatizado.

ERP (Enterprise Resource Planning)

Software que coordina información en todas las áreas, sus objetivos son optimizar los procesos, obtener información fiable, compartir datos y eliminar redundancias. Sus características son: programas con acceso a una base de datos, sus componentes interactúan entre sí y los datos son consistentes y completos.

La consultoría en materia de ERP puede ser de dos tipos:

- **Consultoría de negocios** adapta los procesos de empresa al ERP.
- **Consultoría técnica** programación del sistema, obtención de informes y el estudio de recursos tecnológicos.

CRM (Customer Relationship Managment)

Sistema de gestión de información referida a los clientes de la empresa. Facilita la automatización de gestión de datos de clientes, gestión y seguimiento de ventas, gestión y seguimiento de campañas, etc.

Los sistemas ERP suelen tener incluido un módulo CRM.

BI(Bussiness Intelligence)

Sistema que proporciona herramientas para facilitar el análisis y visualización de datos y se encarga de analizar el pasado, le presente y el futuro de la empresa, realizando predicciones y facilitando la toma de decisiones.

Sus herramientas son: dashboards, OLAP y data mining.

ERP-CRM-BI integrados

Complementos naturales de los sistemas ERP, hoy en día la mayoría de ERP incluyen CRM y BI. Los sistemas ERP incorporan sistemas de plugins modulares que permiten ampliarlos.

Ventajas de implantar ERP

- Instalación única.
- Personalización.
- Sistema integrado.
- Funcionalidades modulares.

6 módulos más importantes en SAP

Módulo SAP	Función principal
FI-Financial Accounting	Gestiona operaciones financieras de la empresa. Permite conocer la posición económica en tiempo real y centralizar los datos contables.
SD-Sales and Distribution	Controla ventas, distribución e inventario. Gestiona transacciones: consultas, ofertas, propuestas, precios, pedidos.
CO-Controlling	Soporta planificación, reporte y monitorización de operaciones. Optimiza informes financieros y análisis de costes.
PP-Production Planning	Planifica y gestiona la producción. Coordina datos maestros, planificación global de ventas, recursos, materiales y costes.
MM-Materials Managment	Supervisa aprovisionamiento y gestión de materiales. Incluye proveedores, compras, stocks y facturas.
HCM-Human Capital Management	Gestiona procesos de recursos humanos, nominas, contrataciones, promociones, compensaciones y datos de empleados.

Python

Características principales

- **Lenguaje interpretado**, ejecuta instrucción a instrucción, sin compilación previa.
- **Sintaxis sencilla**, cercana al pseudocódigo, fácil de leer y aprender.
- **Potente** permite realizar muchas acciones con pocas líneas.
- **Tipado dinámico** no requiere declarar tipos de datos.
- **Curva de aprendizaje** suave ideal para principiantes y útil para expertos.

Elementos básicos

- **Comentarios** # para una línea y tres comillas para multilínea.

```
# Esto es un comentario de una línea

"""
Esto es un comentario
multilínea en Python
"""
```

- **Tipos de datos** enteros, float, booleanos, strings, None.
- **Operadores** aritméticos (+, -, *, /, //), comparaciones (==, !=, <, >), lógicos (and, or, not).

```
# Enteros y operaciones básicas
a = 10
b = 3
print(a + b)    # 13
print(a / b)    # 3.333... (float)
print(a // b)   # 3 (división entera)

# Booleanos
print(True and False) # False
print(not True)        # False

# Comparaciones
print(5 > 2)    # True
print(5 == 2)   # False
```

Variables y colecciones

- **Variables** se crean al asignar valor, convención *snake_case*.
- **Listas** mutables, soportan *slicing*, *append*, *pop*, *extend*.
- **Tuplas** inmutables, permiten desempaqueado.
- **Diccionarios** clave/valor, métodos *get*, *setdefault*, *keys*, *values*.
- **Conjuntos**(set) operaciones de unión, intersección, diferencia.

```

# Variables
nombre = "Raquel"
edad = 22

# Listas
lista = [1, 2, 3]
lista.append(4)
print(lista)          # [1, 2, 3, 4]
print(lista[-1])      # 4 (último elemento)

# Tuplas
tupla = (10, 20, 30)
print(tupla[0])       # 10

# Diccionesarios
dicc = {"uno": 1, "dos": 2}
print(dicc["uno"])     # 1
print(dicc.get("tres", 0)) # 0 (valor por defecto)

# Conjuntos
conjunto = {1, 2, 3}
conjunto.add(4)
print(conjunto)        # {1, 2, 3, 4}

```

Control de flujo

- If/ elif/ else identacion obligatoria.
- Bucles
 - For
 - While
- Excepciones *try/except, raise*.
- Iterables/iteradores *iter(), __next__(), StopIteration*.

```
# If / elif / else
x = 5
if x > 10:
    print("Mayor que 10")
elif x == 10:
    print("Igual a 10")
else:
    print("Menor que 10")

# For
for animal in ["perro", "gato", "ratón"]:
    print(animal, "es un mamífero")

# While
i = 0
while i < 3:
    print(i)
    i += 1
```

Funciones

Se definen con *def*.

- **Parámetros posicionales**, palabras clave, **args*, ***kwargs*.
- **Funciones anónimas** *lambda*.
- **Funciones de orden superior** *map*, *filter*, *reduce*.
- **Comprensiones** listas *[x for x in ...]*, diccionarios *{k:v}*, conjuntos *{c}*.

```

# Definición básica
def suma(a, b):
    return a + b

print(suma(3, 4)) # 7

# Función con *args y **kwargs
def mostrar(*args, **kwargs):
    print("Args:", args)
    print("Kwargs:", kwargs)

mostrar(1, 2, nombre="Raquel", curso="DAM")

# Lambda
doble = lambda x: x * 2
print(doble(5)) # 10

```

Clases

Se definen con `class Nombre(object)`.

- **Atributos de clase** compartidos por todas las instancias.
- **Constructor** `__init__`.
- **Métodos** de instancia(`self`), de clase(`@classmethod`), estáticos(`@staticmethod`).


```

class Humano:
    especie = "H. sapiens" # atributo de clase

    def __init__(self, nombre):
        self.nombre = nombre

    def decir(self, msg):
        return f"{self.nombre}: {msg}"

    @classmethod
    def get_especie(cls):
        return cls.especie

    @staticmethod
    def roncar():
        return "*roncar*"

persona = Humano("Raquel")
print(persona.decir("Hola"))           # Raquel: Hola
print(Humano.get_especie())           # H. sapiens
print(Humano.roncar())                 # *roncar*

```

Módulos

- Importación *import*, *from ... import...*, *import...as...* .
- Herramienta *pip* para instalar paquetes.
- *Dir()* para listar funciones y atributos de un módulo.

```

import math
print(math.sqrt(16))    # 4.0

from math import ceil, floor
print(ceil(3.7))        # 4
print(floor(3.7))       # 3

```

Avanzado

- **Generadores** *yield*, producen valores bajo demanda.
- **Decoradores** funciones que envuelven otras, añaden funcionalidad (*@wraps*)

```
# Generador
def contar_hasta(n):
    for i in range(n):
        yield i

for numero in contar_hasta(5):
    print(numero) # 0,1,2,3,4

# Decorador
def decorador(func):
    def wrapper():
        print("Antes de ejecutar")
        func()
        print("Después de ejecutar")
    return wrapper

@decorador
def saludar():
    print("Hola mundo")

saludar()
```