0Este ejercicio práctico va a tener dos partes:

Una primera parte que será esta tarea del tema 5, en la que nos familiarizaremos con el lenguaje de programación Python y haremos algunos módulos (programas en Python) que podamos utilizar posteriormente. Y un segunda parte, a desarrollar en el examen presencial de mayo, en el que aprovecharemos los módulos creados para crear nuestra propia aplicación o módulo para Odoo A la hora de programar podemos utilizar el paradigma imperativo (estructurado, procedimental, modular) o el paradigma de programación orientada a objetos (POO). Lo lógico es que utilicemos este último, ayudado del primero. Por tanto los programas serán orientados a objetos, utilizando por tanto clases y con control de excepciones.

1. **Crear un módulo para validación de nombres de usuarios**.
   Criterios:
   1. El nombre de usuario debe contener un mínimo de 6 caracteres y un máximo de 12.
   2. El nombre de usuario debe ser alfanumérico.
   3. Nombre de usuario válido, retorna 0 o el mensaje “El nombre de usuario es válido”
   4. Nombre de usuario con menos de 6 caracteres, retorna 1 o el mensaje "El nombre de usuario debe contener al menos 6 caracteres".
   5. Nombre de usuario con más de 12 caracteres, retorna 2 o el mensaje "El nombre de usuario no puede contener más de 12 caracteres".
   6. Nombre de usuario con caracteres distintos a los alfanuméricos, retorna 3 o el mensaje "El nombre de usuario puede contener solo letras y números".

```
def testUsuario(u):
    lmin = 6
    lmax = 12

    if len(u)<6: # Usuario al menos de 6 caracteres de longitud
        print("El nombre se usuario debe contener al menos 6 caracteres")
        Salida = 1
    elif len(u)>12: # Usuario de no más de 12 caracteres de longitud
        print("El nombre de usuario no puede tener más de 12 caractéres")
        Salida = 2
    elif not(u.isalnum()): # Usuario debe ser una cadena alfanumérica
        print("El nombre de usuario sólo puede contener caracteres alfanuméricos")
        Salida = 3
    else: # Nombre se usuario correcto
        print("Nombre de usuario válido")
        nombreUsuario = u
        return 0
```

2. **Crear un módulo para validación de contraseñas**.
   Criterios
   - La contraseña debe contener un mínimo de 8 caracteres.
   - Una contraseña debe contener letras minúsculas, mayúsculas, números y al menos 1 carácter no alfanumérico.
   - La contraseña no puede contener espacios en blanco.
   - Contraseña válida, retorna True
   - Contraseña no válida (segura) según estos criterios, retorna False

```
def testPasswd(p):
    numCar = may = min = dig = noAlfNum = False
    esp = False
    lmin = 7

    if len(p) > lmin:
      numCar = True
    for c in p:
      if c.isspace(): # Comprobamos si hay espacios
        esp = True
      if c.isupper(): # Comprobamos si hay mayúsculas
        may = True
      if c.islower(): # Comprobamos si hay minúsculas
        min = True
      if c.isdigit(): # Comprobamos si hay números
        dig = True
      if not c.isalnum():
        noAlfNum = True # Comprobamos si caracteres especiales

    if (numCar and not esp and may and min and dig and noAlfNum):
      return 0
    else:
      return 1
```

3. **Crear un módulo que solicite un nombre de usuario y contraseña y los valide utilizando los módulos generados en los dos ejercicios anteriores.**

```
import credenciales as func # credenciales contiene las dos funciones anteriores

u = input("Introduzca el nombre de usuario: ")

if (func.testUsuario(u) == 0):
  print('Ya hemos cambiado el usuario')
else:
  print("No hemos podido cambiar el usuario")

contraseña = input("Introduzca la contraseña: ")
if (func.testPasswd(contraseña)== 0):
  print("Hemos cambiado la contraseña")
else:
  print("No hemos cambiado la contraseña")
```

4. **Crear un módulo que construya una tabla con datos proporcionados por el usuario y posteriormente permita la búsqueda en la tabla, mediante clave en los datos introducidos.**
   - Utilizaremos diccionarios.
   - El programa constará de un menú con las siguientes opciones:
     - Solicitar los datos para almacenar (apellidos, nombre, fecha nacimiento, dirección, contraseña. Etc.)
     - Permitir buscar datos correspondientes a un nombre o apellido

introducidos y mostrar en pantalla.
♣ Añadir los datos de la tabla a un fichero CSV
♣ Recuperar datos de un fichero CSV debidamente formateado, para completar la tabla.

Código enviado en un archivo comprimido aparte.

5. Módulo Stock

o Acceder a la correspondiente tabla de Odoo, para mostrar al usuario un listado de productos. El usuario podrá seleccionar uno para obtener información del Código, descripción y stock actual.

o Generar un fichero CSV con el stock de todos los productos. En el fichero deben mostrarse como mínimo código, descripción y stock actual de cada producto. Las líneas del fichero estarán ordenadas por descripción del producto.

Antes de empezar debemos instalar el módulo que contenga las tablas que nos hacen falta para sacar los productos.

Para poder acceder desde Python a una base de datos, tenemos que instalar el módulo necesario para tal acceso. En nuestro caso necesitaremos el adecuado para PostgreSQL.

Hay varias opciones. Yo he elegido “psycopg2”.

Para comprobar si tenemos el módulos instalado, ejecutaremos el comando “help (“psycopg2”) en la línea de comandos de Python. En caso de que exista nos dará toda la información acerca del módulo. Si no existiese, lo instalaremos con el comando, desde la línea de comandos del sistema operativo, “pip install psycopg2”.

```
C:\Users\Julio>pip install psycopg2
Collecting psycopg2
  Downloading psycopg2-2.8.6-cp38-cp38-win_amd64.whl (1.1 MB)
     |████████████████████████████████| 1.1 MB 6.4 MB/s
Installing collected packages: psycopg2
Successfully installed psycopg2-2.8.6
WARNING: You are using pip version 20.2.3; however, version 21.0.1 is available.
You should consider upgrading via the 'c:\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\Julio>
```

Una vez lo tenemos instalado podemos usarlo.

```
Python 3.8 (64-bit)
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> help("psycopg2")
Help on package psycopg2:

NAME
    psycopg2 - A Python driver for PostgreSQL

DESCRIPTION
    psycopg is a PostgreSQL_ database adapter for the Python_ programming
    language. This is version 2, a complete rewrite of the original code to
    provide new-style classes for connection and cursor objects and other sweet
    candies. Like the original, psycopg 2 was written with the aim of being very
    small and fast, and stable as a rock.

    Homepage: https://psycopg.org/

    .. _PostgreSQL: https://www.postgresql.org/
    .. _Python: https://www.python.org/

    :Groups:
      * `Connections creation`: connect
      * `Value objects constructors`: Binary, Date, DateFromTicks, Time,
        TimeFromTicks, Timestamp, TimestampFromTicks

PACKAGE CONTENTS
    _ipaddress
    _json
    _lru_cache
    _psycopg
    _range
    compat
    errorcodes
    errors
    extensions
    extras
    pool
    sql
    tz

SUBMODULES
    _ext

CLASSES
    builtins.Exception(builtins.BaseException)
        Error
            DatabaseError
                DataError
                IntegrityError
```

Para la presentación de la lista de productos por pantalla he utilizado el módulo "tabulate".

El código está en el archivo adjunto.

6. Módulo Clientes/proveedores

o A partir de un fichero CSV con datos de clientes, añadirlos a la tabla correspondiente de la base de datos de Odoo.

- ♣El fichero CSV contendrá como información los siguientes datos
- ♣Nombre del cliente
- ♣Persona de contacto
- ♣Dirección
- ♣Teléfono
- ♣e-mail
- ♣Página web
- ♣Población
- ♣Código Postal
- ♣Tipo: (P: proveedor, C: cliente, T: proveedor y cliente)
  - →En el caso de P, se pondrá a TRUE el correspondiente campo de proveedor
  - →Igual con C, pero para cliente
  - →Si es T, se pondrán a TRUE tanto el proveedor como el cliente.

o Comprobar previamente si el cliente existe, en este caso se modificará el registro correspondiente con los datos del fichero.

o Para crear el login se tomará la primera letra del nombre y el primer apellido, todo en minúsculas

o Teniendo en cuenta que los nombres de usuario del fichero estarán formados por **Nombre Apellido1 Apellido2**, el campo **password**se actualizará con el dato generado según la siguiente regla:

♣Se mezclarán aleatoriamente el **Nombre, Apellido1 y Apellido2**

♣Se tomará la primera letra del primer elemento, seguido de los segundos de la hora actual, seguido de las letras 2 y 3 del segundo elemento (la tercera letra se convertirá a mayúscula), seguido los minutos de la hora actual, seguido de la letra 4 del tercer elemento, finalizando con uno de estos tres símbolos: $, %, &, escogidos al azar.

♣Ejemplo son las 18:55:43 y el usuario se llama Manolo Martín Pajares. Tras mezclar aleatoriamente obtenemos Martín Pajares Manolo. La password sería: **M43aJ55a&**

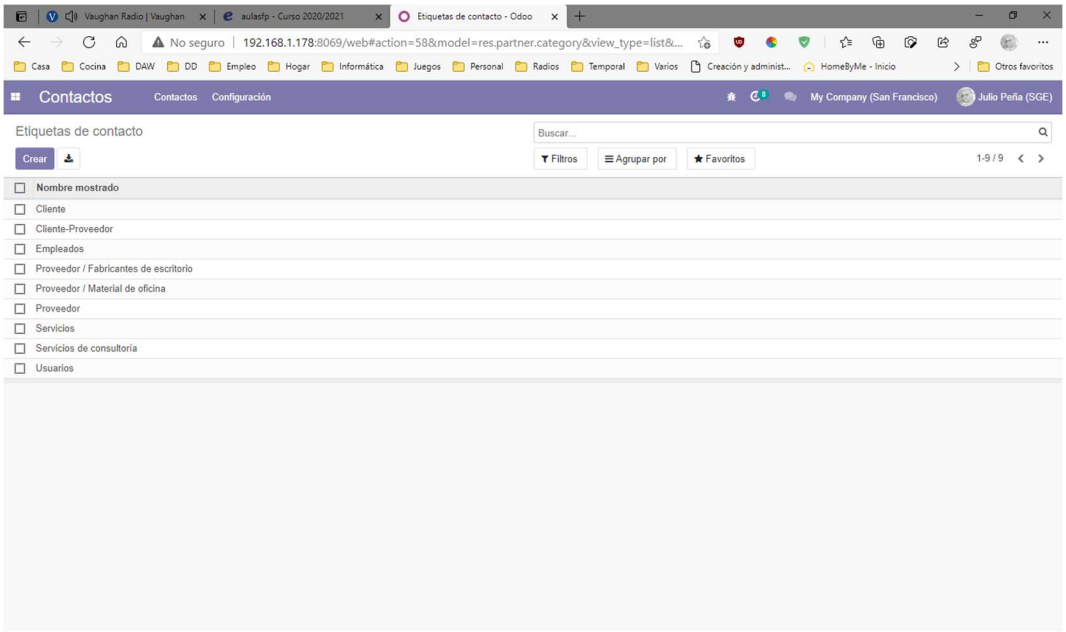
o Se informará si se ha procedido o no a las modificaciones. En caso de que se hayan realizado modificaciones se indicará cuantos registros se han añadido y cuantos se han modificado.

o Módulo Modificar cliente/Proveedor

o Al igual que el módulo de stock, se proporcionará un listado de clientes/proveedores

o Se permitirá la modificación del registro que elija el usuario

Antes de poder responder a esta pregunta, necesitamos crear las etiquetas de Odoo "cliente" y "cliente-proveedor". Esto lo hacemos dentro del módulo contactos y entrando en "configuración". Una vez allí añadimos las etiquetas.



Ahora podremos asignar dichas etiquetas en el módulo “contactos”.
Para el ejercicio, en el fichero csv, he creado 4 usuarios que comienzan con “Pedi”.