

Sistemas de gestión empresarial

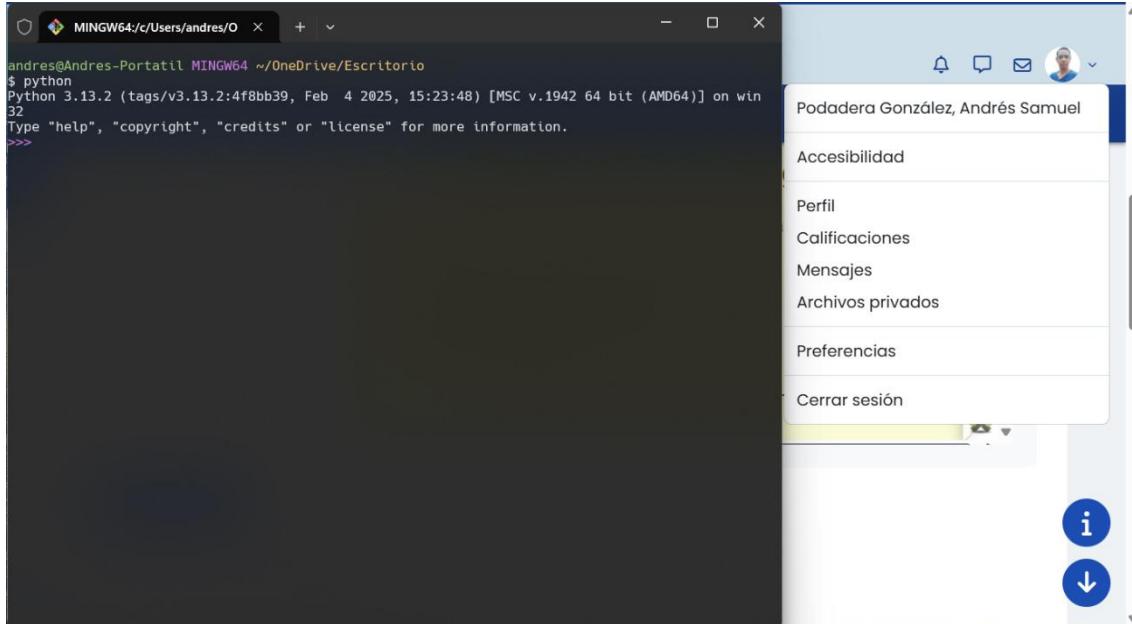
Tarea Unidad 4

Contenido

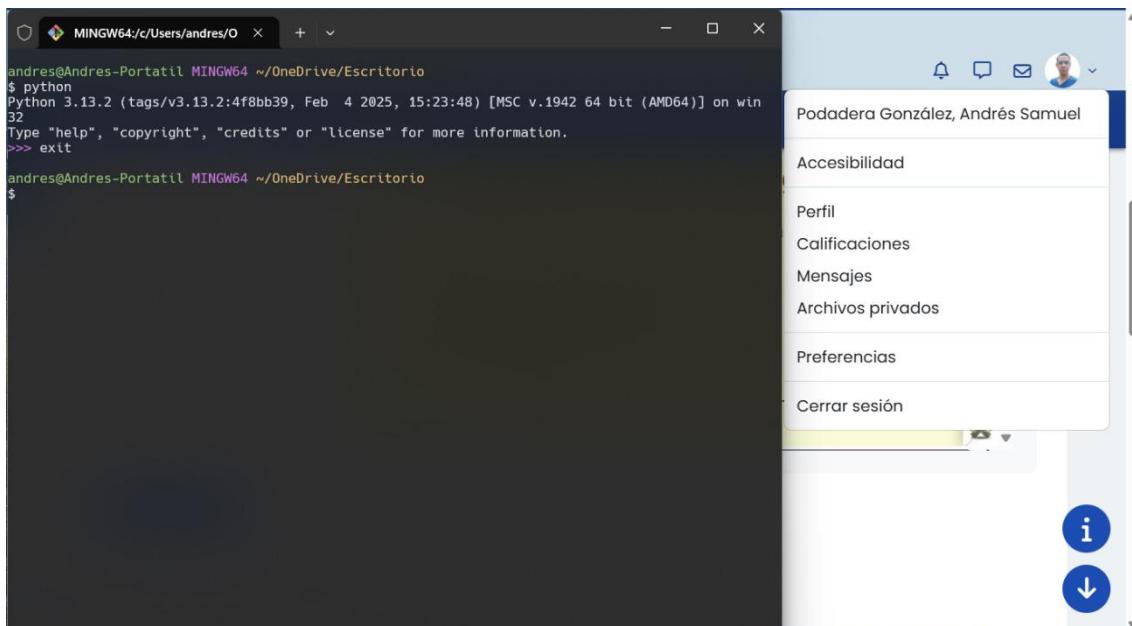
Parte I - Uso del intérprete de comandos	2
1. Ejecuta el intérprete de python desde la terminal de comandos del sistema operativo:	2
2. Identifica la versión de PostgreSQL instalada en tu máquina.....	2
3. Ejecutar programas simples en Python.	4
Parte II - Programación en Python	5
Parte III - Creación de un módulo en Odoo	13
Actividad 01: Datos del módulo	15
Actividad 02: Modelo	17
Actividad 03: Menú	19
ACTIVIDAD 4: Vistas	20
ACTIVIDAD 5: Datos precargados	23

Parte I - Uso del intérprete de comandos

1. Ejecuta el intérprete de python desde la terminal de comandos del sistema operativo:



Para salir del intérprete de python podemos utilizar la orden **exit()**.

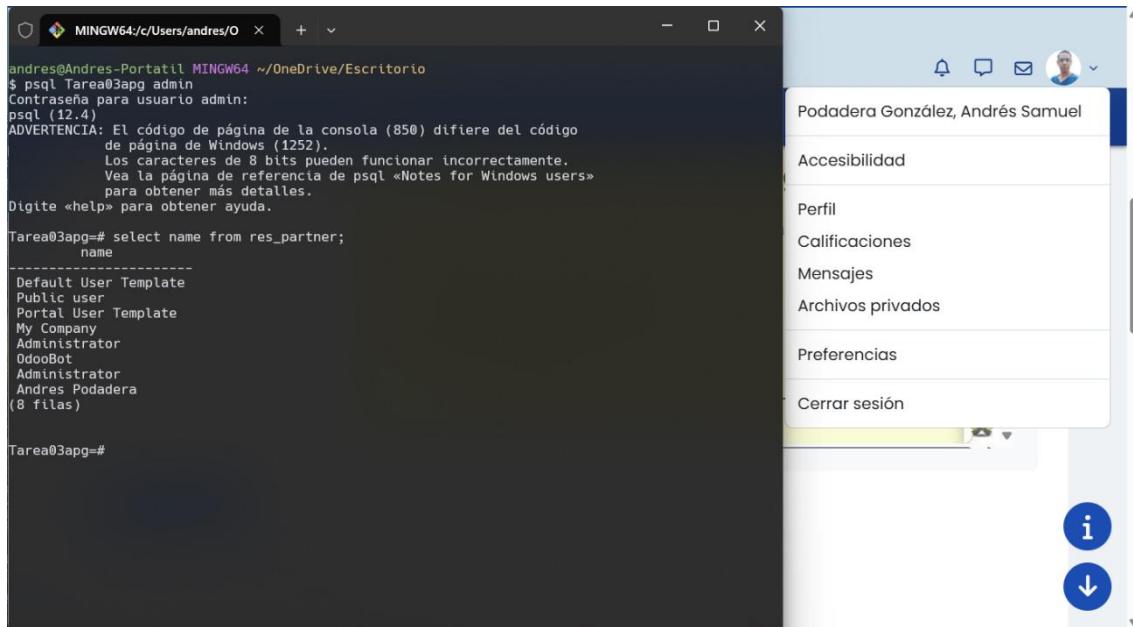


2. Identifica la versión de PostgreSQL instalada en tu máquina. Para ello, podemos ejecutar el intérprete de PostgreSQL desde la terminal de comandos del sistema operativo. Utiliza la orden:

psql nombre_basedatos nombre_usuario

Es posible que te pida la contraseña del usuario de la base de datos, en ese caso debes introducir la contraseña que proporcionaste al instalar Odoo.

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025



A screenshot of a Windows desktop environment. On the left is a terminal window titled 'MINGW64:/c/Users/andres/O' showing a PostgreSQL session. The user 'andres' is connected as 'Tarea03apg' and is running the command 'select name from res_partner;'. The output shows a list of names from the 'res_partner' table. On the right is a user profile sidebar for 'Podadera González, Andrés Samuel', listing options like Accesibilidad, Perfil, Calificaciones, Mensajes, Archivos privados, Preferencias, and Cerrar sesión.

```
andres@Andres-Portatil MINGW64 ~/OneDrive/Escritorio
$ psql Tarea03apg admin
Contraseña para usuario admin:
psql (12.4)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

Tarea03apg=# select name from res_partner;
 name
-----
 Default User Template
 Public user
 Portal User Template
 My Company
 Administrator
 OdooBot
 Administrator
 Andres Podadera
(8 filas)

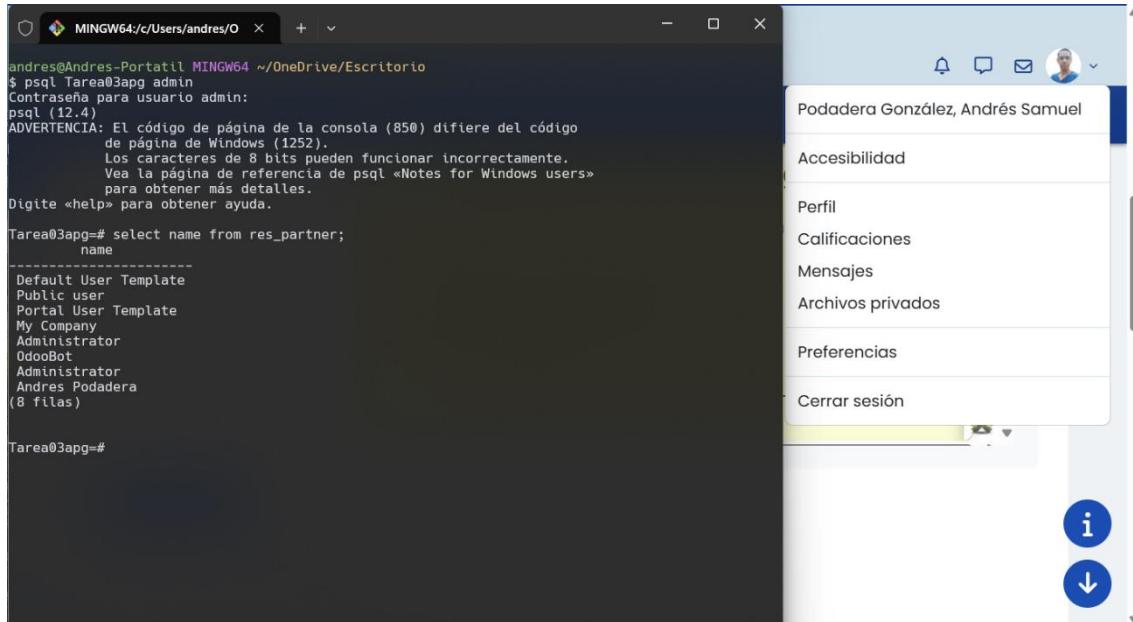
Tarea03apg=#

```

Ejecuta la sentencia:

```
select name from res_partner;
```

Con esta sentencia estamos mostrando el campo **name** de la tabla **res_partner**:



A screenshot of a Windows desktop environment, identical to the one above, showing a terminal window with the same PostgreSQL session and user profile sidebar.

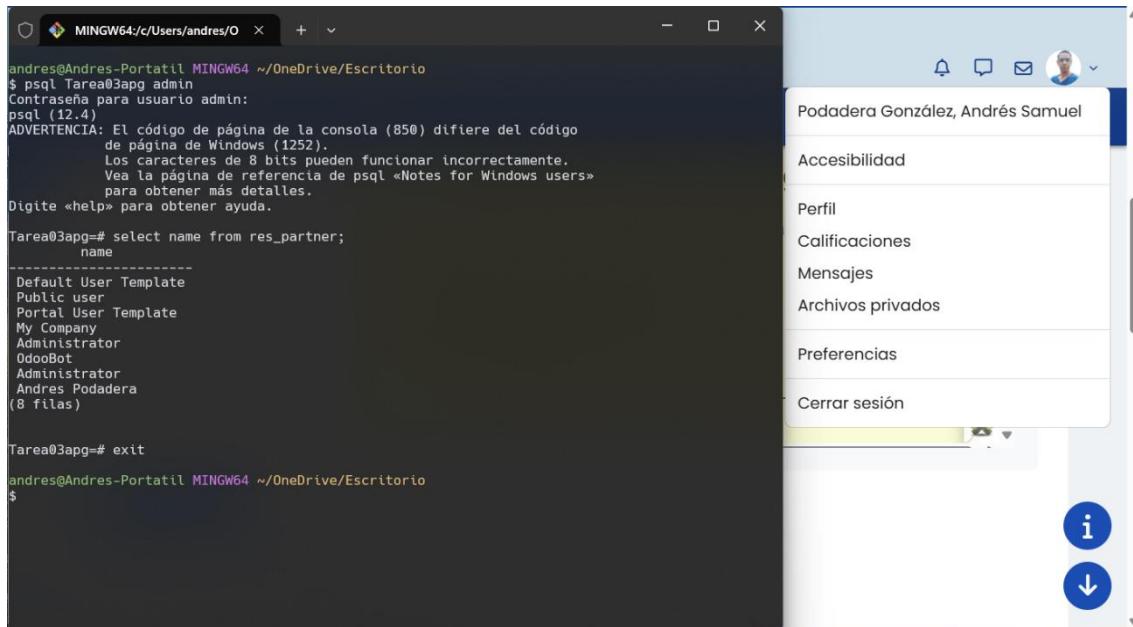
```
andres@Andres-Portatil MINGW64 ~/OneDrive/Escritorio
$ psql Tarea03apg admin
Contraseña para usuario admin:
psql (12.4)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

Tarea03apg=# select name from res_partner;
 name
-----
 Default User Template
 Public user
 Portal User Template
 My Company
 Administrator
 OdooBot
 Administrator
 Andres Podadera
(8 filas)

Tarea03apg=#

```

Para salir del intérprete de PostgreSQL podemos utilizar **exit**, **\q** o **Ctrl+C**.



A screenshot of a Windows desktop environment. On the left is a terminal window titled 'MINGW64:/c/Users/andres/O' showing a PostgreSQL session. The session starts with connecting to the 'admin' user, followed by a query to select names from 'res_partner'. The results show several entries including 'Default User Template', 'Public user', 'Portal User Template', 'My Company', 'Administrator', 'OdooBot', and 'Andrés Podadera'. The session ends with an 'exit' command. On the right is a vertical user profile menu for 'Podadera González, Andrés Samuel', listing options like Accesibilidad, Perfil, Calificaciones, Mensajes, Archivos privados, Preferencias, and Cerrar sesión. Below the menu are two blue circular icons with white symbols: an 'i' and a downward arrow.

```
andres@Andres-Portatil MINGW64 ~/OneDrive/Escritorio
$ psql Tarea03apg admin
Contraseña para usuario admin:
pqsl (12.4)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

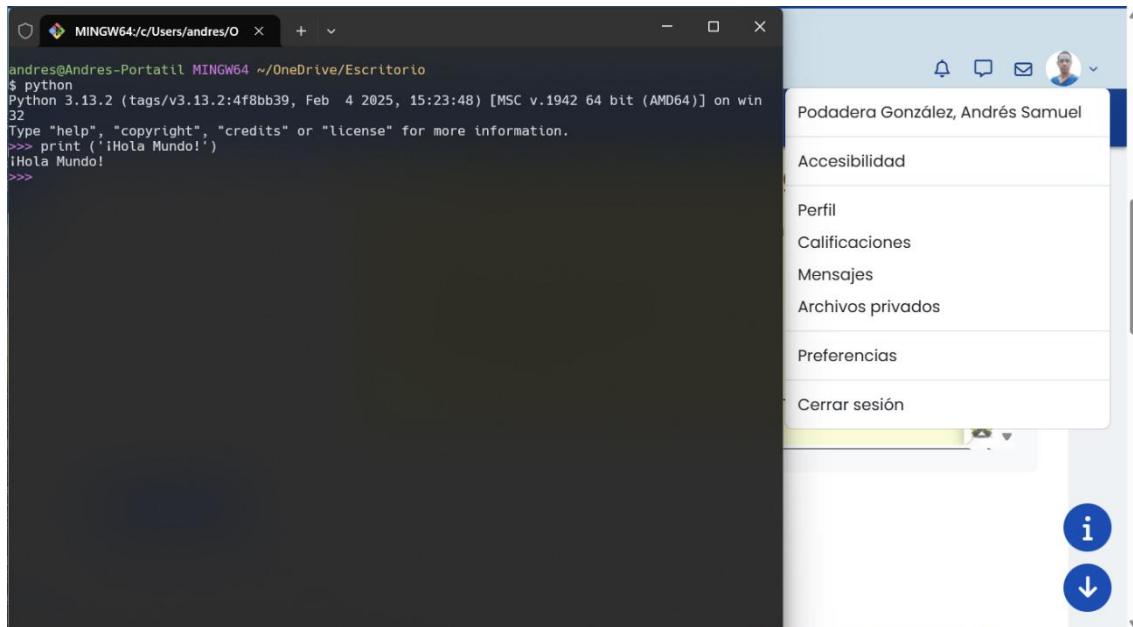
Tarea03apg=# select name from res_partner;
   name
-----
 Default User Template
 Public user
 Portal User Template
 My Company
 Administrator
 OdooBot
 Administrator
 Andrés Podadera
(8 filas)

Tarea03apg=# exit
andres@Andres-Portatil MINGW64 ~/OneDrive/Escritorio
$
```

3. Ejecutar programas simples en Python.

Vamos a ejecutar una sentencia simple que muestre un mensaje de ¡Hola mundo!. Para ejecutar este programa en Python podemos utilizar:

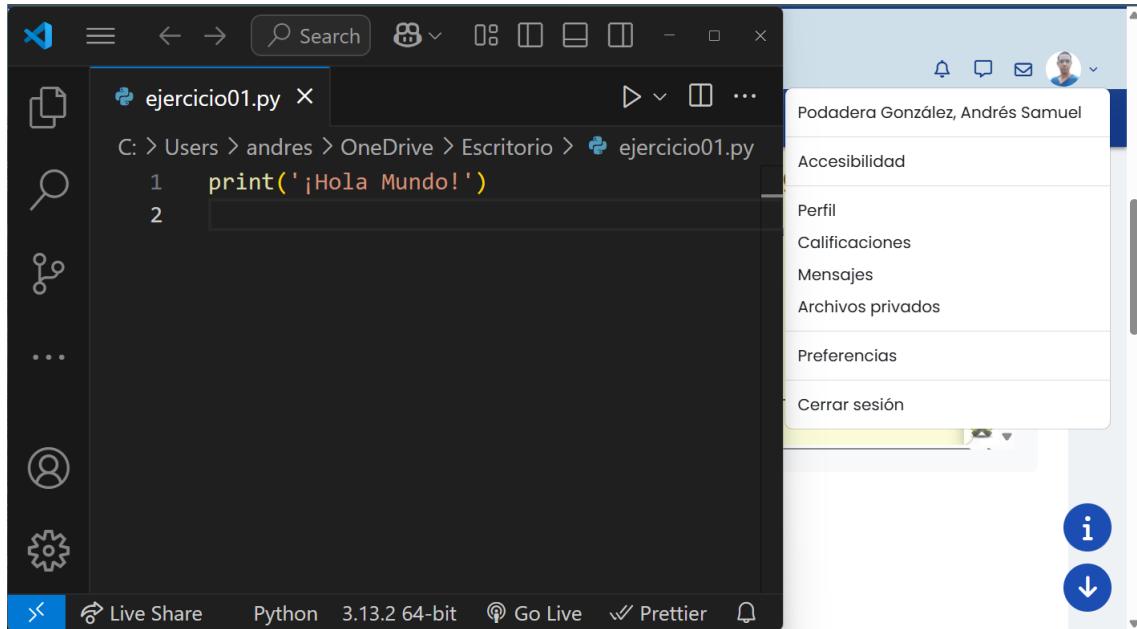
- El intérprete de comandos



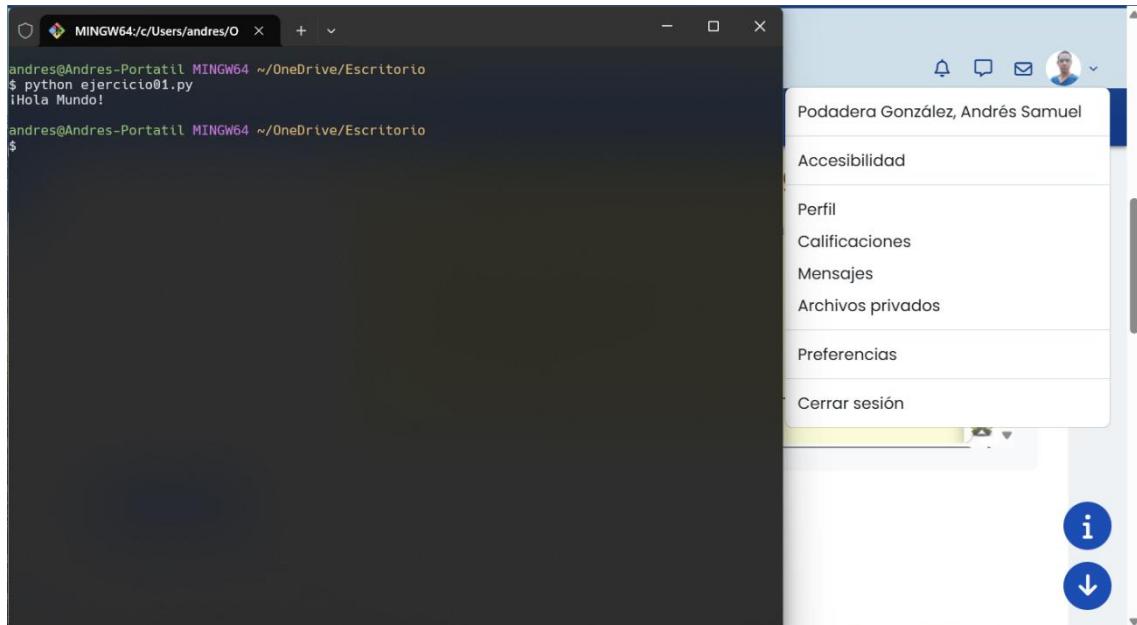
A screenshot of a Windows desktop environment, similar to the previous one. On the left is a terminal window titled 'MINGW64:/c/Users/andres/O' showing a Python session. The session starts with running the Python interpreter ('\$ python'), followed by a 'help' command which lists options like 'copyright', 'credits', and 'license'. Then, a print statement is executed: '>>> print ("¡Hola Mundo!")'. The output shows the text '¡Hola Mundo!'. On the right is a vertical user profile menu for 'Podadera González, Andrés Samuel', listing options like Accesibilidad, Perfil, Calificaciones, Mensajes, Archivos privados, Preferencias, and Cerrar sesión. Below the menu are two blue circular icons with white symbols: an 'i' and a downward arrow.

```
andres@Andres-Portatil MINGW64 ~/OneDrive/Escritorio
$ python
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print ('¡Hola Mundo!')
¡Hola Mundo!
```

- Almacenar el programa en un archivo que podemos crear como por ejemplo el bloc de notas y guardar con extensión .py:



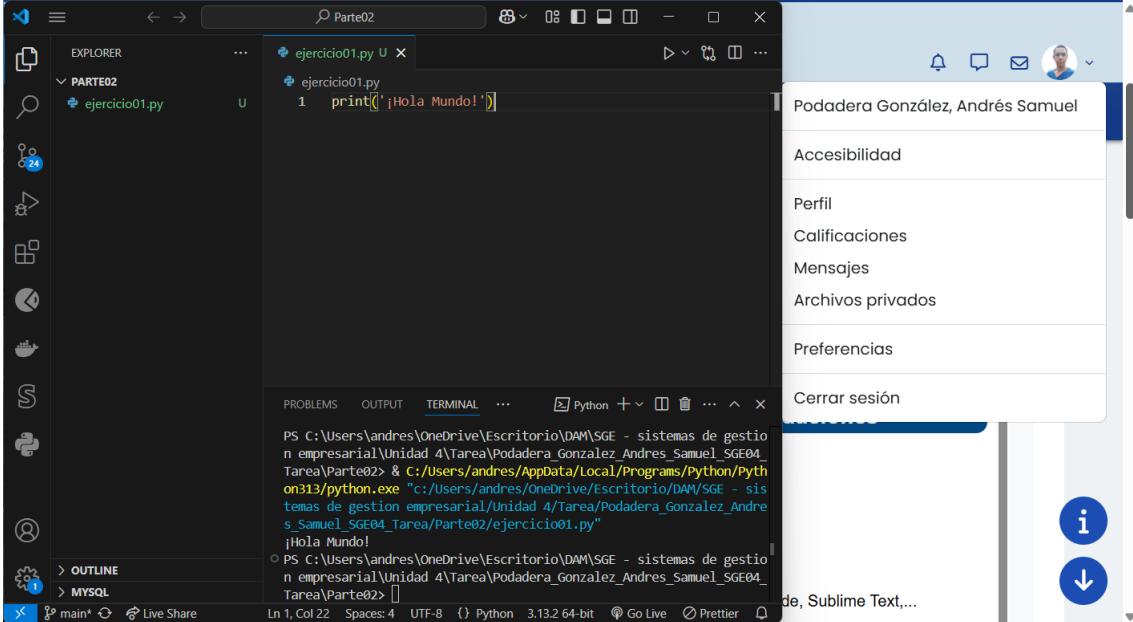
Luego, podemos ejecutar este archivo con la siguiente orden desde la terminal de comandos del sistema operativo:



Parte II - Programación en Python

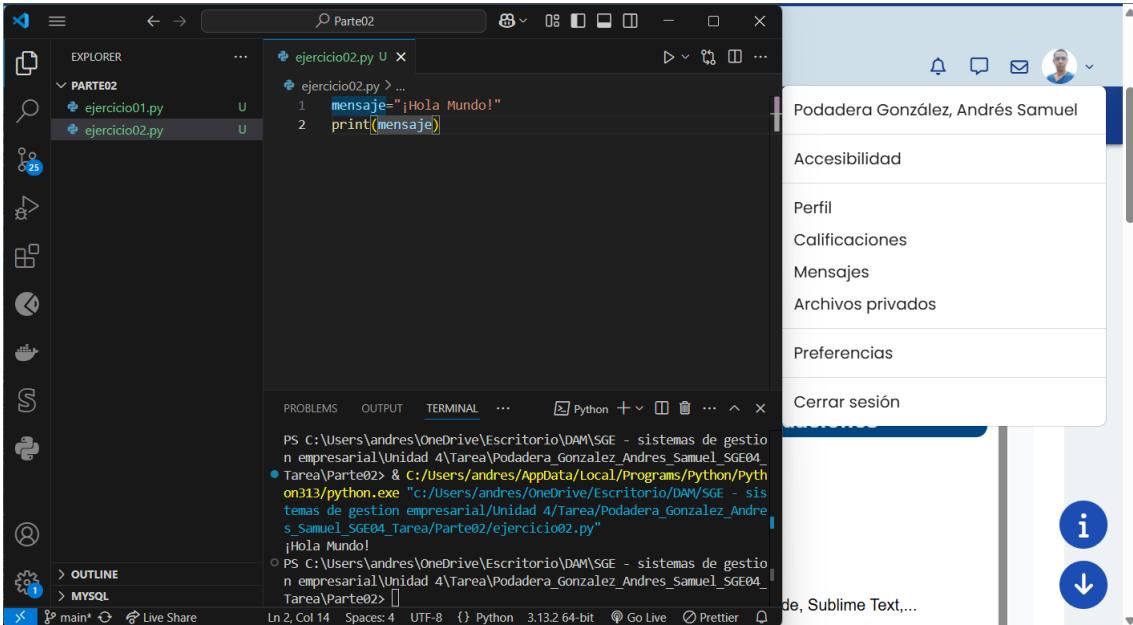
Ejecuta el código correspondiente a los siguientes ejercicios. Debes crear un archivo .py por cada ejercicio.

1. Escribir un programa que muestre por pantalla la cadena **¡Hola Mundo!**



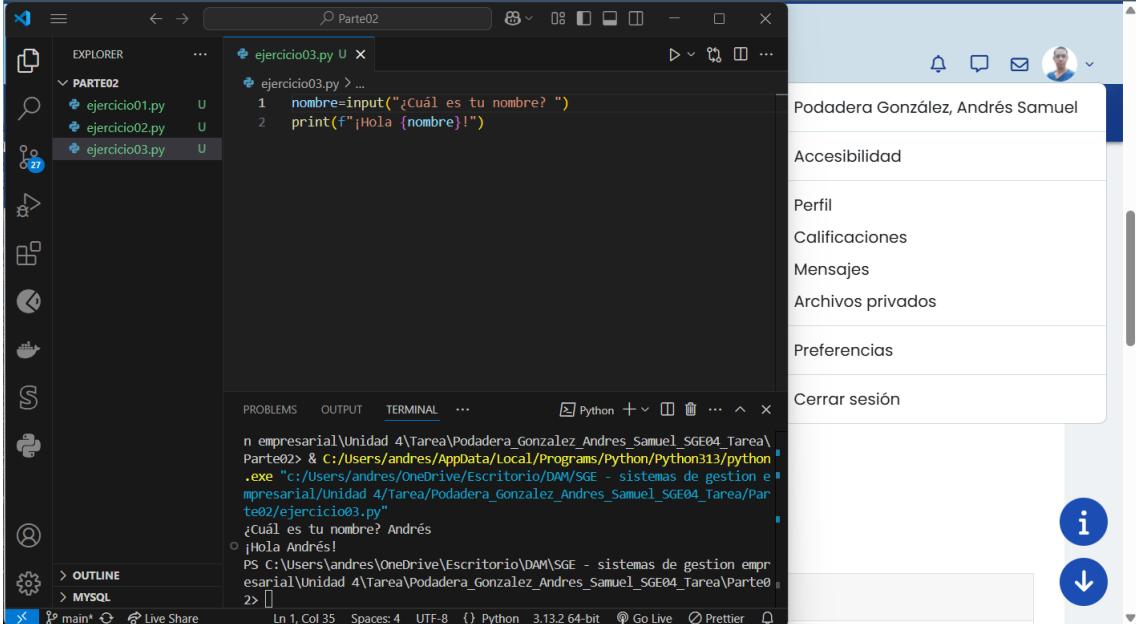
The screenshot shows the VS Code interface with the terminal tab active. The terminal output shows the command PS C:\... being run, followed by the execution of C:/Users/andres/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/andres/OneDrive/Escritorio/DAM/SGE - sistemas de gestio...". The output of the program is '¡Hola Mundo!', indicating successful execution.

2. Escribir un programa que almacene la cadena ¡Hola Mundo! en una variable y luego muestre por pantalla el contenido de la variable.



The screenshot shows the VS Code interface with the terminal tab active. The terminal output shows the command PS C:\... being run, followed by the execution of C:/Users/andres/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/andres/OneDrive/Escritorio/DAM/SGE - sistemas de gestio...". The output of the program is '¡Hola Mundo!', indicating successful execution.

3. Escribir un programa que pregunte el nombre del usuario en la consola y después de que el usuario lo introduzca muestre por pantalla la cadena ¡Hola <nombre>!, donde <nombre> es el nombre que el usuario haya introducido.



The screenshot shows the VS Code interface with the title bar "Parte02". The Explorer sidebar on the left shows files "ejercicio01.py", "ejercicio02.py", and "ejercicio03.py" under the "PARTE02" folder. The main editor window displays the following Python code:

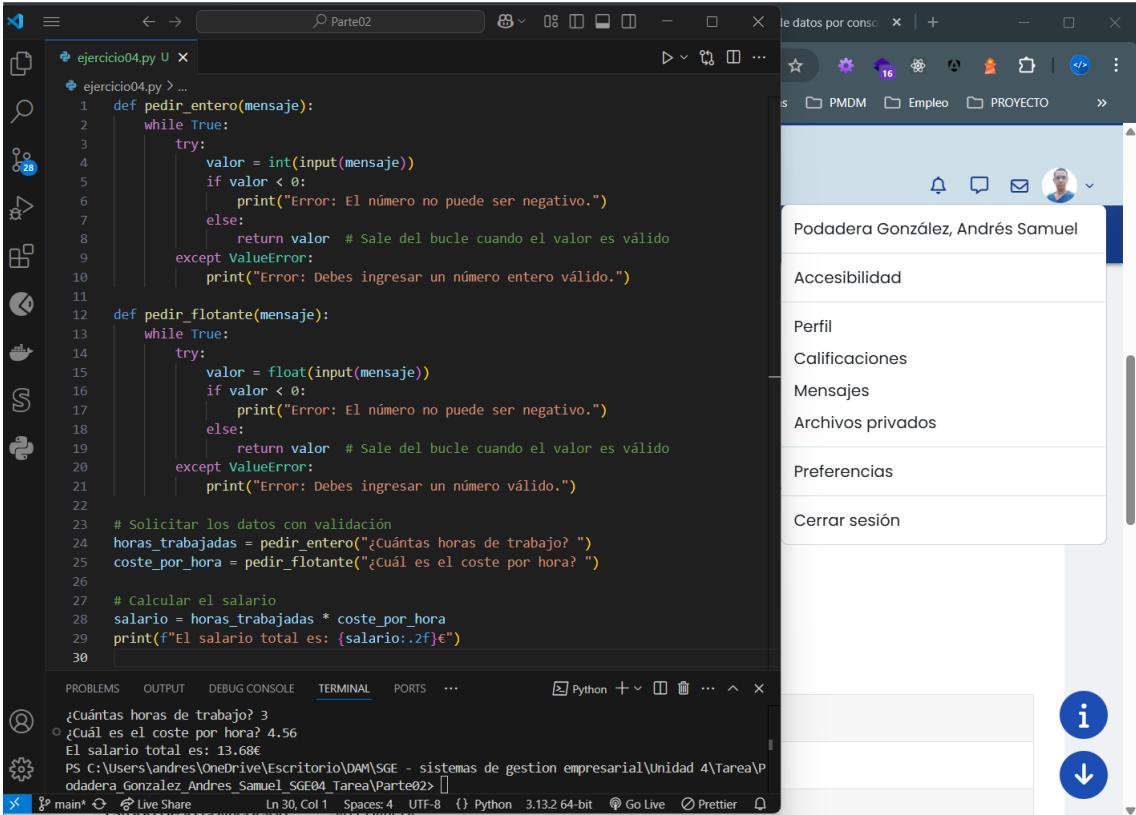
```
nombre=input("¿Cuál es tu nombre? ")
print(f"¡Hola {nombre}!")
```

The terminal below shows the output of running the script:

```
n empresarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02> & C:/Users/andres/AppData/Local/Programs/Python/Python313/python.exe "C:/Users/andres/OneDrive/Escritorio/DAM/SGE - sistemas de gestion empresarial/Unidad 4/Tarea/Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea/Parte02/ejercicio03.py"
¿Cuál es tu nombre? Andrés
¡Hola Andrés!
```

The status bar at the bottom indicates "Ln 1, Col 35" and "Spaces: 4" in "UTF-8" encoding.

4. Escribir un programa que pregunte al usuario por el número de horas trabajadas y el coste por hora. Después debe mostrar por pantalla la paga que le corresponde.



The screenshot shows the VS Code interface with the title bar "Parte02". The Explorer sidebar on the left shows files "ejercicio01.py", "ejercicio02.py", and "ejercicio04.py" under the "PARTE02" folder. The main editor window displays the following Python code:

```
def pedir_entero(mensaje):
    while True:
        try:
            valor = int(input(mensaje))
            if valor < 0:
                print("Error: El número no puede ser negativo.")
            else:
                return valor # Sale del bucle cuando el valor es válido
        except ValueError:
            print("Error: Debes ingresar un número entero válido.")

def pedir_flotante(mensaje):
    while True:
        try:
            valor = float(input(mensaje))
            if valor < 0:
                print("Error: El número no puede ser negativo.")
            else:
                return valor # Sale del bucle cuando el valor es válido
        except ValueError:
            print("Error: Debes ingresar un número válido.")

# Solicitar los datos con validación
horas_trabajadas = pedir_entero("¿cuántas horas de trabajo? ")
coste_por_hora = pedir_flotante("¿Cuál es el coste por hora? ")

# Calcular el salario
salario = horas_trabajadas * coste_por_hora
print(f"El salario total es: {salario:.2f}")
```

The terminal below shows the output of running the script:

```
¿cuántas horas de trabajo? 3
¿Cuál es el coste por hora? 4.56
El salario total es: 13.68
```

The status bar at the bottom indicates "Ln 30, Col 1" and "Spaces: 4" in "UTF-8" encoding.

5. Escribir un programa que pregunte el nombre del usuario en la consola y un número entero e imprima por pantalla en líneas distintas el nombre del usuario tantas veces como el número introducido.

The screenshot shows the VS Code interface with the following details:

- File:** ejercicio05.py
- Code:**

```
ejercicio05.py > ...
def pedir_entero(mensaje):
    while True:
        try:
            valor = int(input(mensaje))
            if valor < 0:
                print("Error: El número no puede ser negativo.")
            else:
                return valor # Sale del bucle cuando el valor es válido
        except ValueError:
            print("Error: Debes ingresar un número entero válido.")

veces_a_imprimir=pedir_entero("¿Cuántas veces deseas imprimir tu nombre? ")
nombre=input("¿Cuál es tu nombre? ")
for i in range(veces_a_imprimir):
    print(f"[{nombre}]")
```
- Terminal Output:**

```
¿Cuántas veces deseas imprimir tu nombre? 3
¿Cuál es tu nombre? Andrés
Andrés
Andrés
Andrés
PS C:\Users\andres\OneDrive\Escritorio\DMAM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Parte02>
```
- Status Bar:** PS C:\Users\andres\OneDrive\Escritorio\DMAM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Parte02 main* Live Share Ln 16, Col 21 Spaces: 4 UTF-8 Python 3.13.2 64-bit Go Live Prettier

6. Escribir un programa que pregunte el nombre completo del usuario en la consola y después muestre por pantalla el nombre completo del usuario tres veces, una con todas las letras minúsculas, otra con todas las letras mayúsculas y otra solo con la primera letra del nombre y de los apellidos en mayúscula. El usuario puede introducir su nombre combinando mayúsculas y minúsculas como quiera.

The screenshot shows the VS Code interface with the following details:

- File:** ejercicio06.py
- Code:**

```
ejercicio06.py > ...
nombre_completo = input("¿Cuál es tu nombre completo? ")
print("Nombre en minúsculas:", nombre_completo.lower())
print("Nombre en mayúsculas:", nombre_completo.upper())
print("Nombre con primera letra en mayúscula:", nombre_completo.title())
```
- Terminal Output:**

```
¿Cuál es tu nombre completo? Andrés samuel Podadera GONZÁLEZ
Nombre en minúsculas: andrés samuel podadera gonzález
Nombre en mayúsculas: ANDRÉS SAMUEL PODADERA GONZÁLEZ
Nombre con primera letra en mayúscula: Andrés Samuel Podadera González
```
- Status Bar:** PS C:\Users\andres\OneDrive\Escritorio\DMAM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Parte02 main* Live Share Ln 5, Col 73 Spaces: 4 UTF-8 Python 3.13.2 64-bit Go Live Prettier

7. Escribir un programa que pregunte al usuario su edad y muestre por pantalla si es mayor de edad o no.

The screenshot shows the VS Code interface with the title bar "Parte02". The Explorer sidebar shows files in the "PARTE02" folder: ejercicio01.py, ejercicio02.py, ejercicio03.py, ejercicio04.py, ejercicio05.py, ejercicio06.py, ejercicio07.py, and ejercicio08.py. The "ejercicio07.py" file is open in the editor, containing Python code to ask for age and print years lived. The terminal shows the execution of the program and its output: "¿Cuál es tu edad? 24" and "Eres mayor de edad". The status bar at the bottom indicates the file is Python 3.13.2 64-bit.

```
def pedir_entero(mensaje):
    while True:
        try:
            valor = int(input(mensaje))
            if valor < 0:
                print("Error: El número no puede ser negativo.")
            else:
                return valor # Sale del bucle cuando el valor es válido
        except ValueError:
            print("Error: Debes ingresar un número entero válido")

edad=pedir_entero("¿Cuál es tu edad? ")
if edad<18:
    print("Eres menor de edad")
else:
    print("Eres mayor de edad")
```

8. Escribir un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

The screenshot shows the VS Code interface with the title bar "Parte02". The Explorer sidebar shows files in the "PARTE02" folder: ejercicio01.py, ejercicio02.py, ejercicio03.py, ejercicio04.py, ejercicio05.py, ejercicio06.py, ejercicio07.py, and ejercicio08.py. The "ejercicio08.py" file is open in the editor, containing Python code to ask for age and print all years lived from 1 to the input. The terminal shows the execution of the program and its output: "¿Cuál es tu edad? 15" followed by a list of years from 1 to 15. The status bar at the bottom indicates the file is Python 3.13.2 64-bit.

```
def pedir_entero(mensaje):
    while True:
        try:
            valor = int(input(mensaje))
            if valor < 0:
                print("Error: El número no puede ser negativo.")
            else:
                return valor # Sale del bucle cuando el valor es válido
        except ValueError:
            print("Error: Debes ingresar un número entero válido")

edad=pedir_entero("¿Cuál es tu edad? ")
for i in range(1,edad+1):
    print(f"Has cumplido {i} años")
```

9. Listas y tuplas. Escribir un programa que almacene las asignaturas de un curso (por ejemplo, Matemáticas, Física, Química, Historia y Lengua) en una lista y la muestre por pantalla.

The screenshot shows the VS Code interface. On the left is the code editor with file 'ejercicio09.py' containing:

```
ejercicio09.py > ...
1 asignaturas=["Matemáticas", "Física", "Química", "Historia", "Lengua"]
2 for asignatura in asignaturas:
3     print(asignatura)
```

The terminal on the right shows the output of running the script:

```
arial/Unidad 4/Tarea/Padadera_Gonzalez_Andres_Samuel_SGE04_Tarea/Parte02/ejercicio09.py"
Matemáticas
Física
Química
Historia
Lengua
PS C:\Users\andres\OneDrive\Escritorio\DM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Parte02>
```

10. Diccionarios. Escribir un programa que guarde en una variable el diccionario `{'Euro':'€', 'Dollar':'$', 'Yen':'¥'}`, pregunte al usuario por una divisa y muestre su símbolo o un mensaje de aviso si la divisa no está en el diccionario.

The screenshot shows the VS Code interface. On the left is the code editor with file 'ejercicio10.py' containing:

```
ejercicio10.py > ...
1 divisas={'Euro':'€', 'Dollar':'$', 'Yen':'¥'}
2
3 divisa = input("Introduce una divisa (Euro, Dollar, Yen): ").capitalize()
4
5 if divisa in divisas:
6     print(f"El símbolo de la {divisa} es {divisas[divisa]}")
7 else:
8     print(f"La divisa {divisa} no está disponible.")
```

The terminal on the right shows the output of running the script twice:

```
odadadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02> & C:/Users/andres/AppData/Local/Programs/Python/3.13/python313/python.exe "c:/Users/andres/OneDrive/Escritorio/DM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea/Padadera_Gonzalez_Andres_Samuel_SGE04_Tarea/Parte02/ejercicio10.py"
Introduce una divisa (Euro, Dollar, Yen): euro
El símbolo del Euro es €
PS C:\Users\andres\OneDrive\Escritorio\DM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Parte02> & C:/Users/andres/AppData/Local/Programs/Python/3.13/python313/python.exe "c:/Users/andres/OneDrive/Escritorio/DM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea/Padadera_Gonzalez_Andres_Samuel_SGE04_Tarea/Parte02/ejercicio10.py"
Introduce una divisa (Euro, Dollar, Yen): prueba
La divisa Prueba no está disponible.
PS C:\Users\andres\OneDrive\Escritorio\DM\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Parte02>
```

11. Diccionarios. Escribir un programa que pregunte al usuario su nombre, edad, dirección y teléfono y lo guarde en un diccionario. Después debe mostrar por pantalla el mensaje `<nombre> tiene <edad> años, vive en <dirección> y su número de teléfono es <teléfono>`.

The screenshot shows the VS Code interface. On the left, the Explorer sidebar shows a folder named 'PARTE02' containing files: ejercicio01.py, ejercicio02.py, ejercicio03.py, ejercicio04.py, ejercicio05.py, ejercicio06.py, ejercicio07.py, ejercicio08.py, ejercicio09.py, ejercicio10.py, ejercicio11.py, and ejercicio12.py. The 'ejercicio11.py' file is open in the main editor area. The code reads user input for name, age, address, and phone number, then prints a formatted string. The terminal below shows the execution of the script and its output. A sidebar on the right displays the user's profile information.

```
nombre=input("¿Cuál es tu nombre? ")
edad=input("¿Cuál es tu edad? ")
direccion=input("¿Cuál es tu dirección? ")
telefono=input("¿Cuál es tu número de teléfono? ")

datos_usuario={'nombre':nombre,
               'edad':edad,
               'direccion':direccion,
               'telefono':telefono}

print(f"[datos_usuario['nombre']] tiene {datos_usuario['edad']} años, vive en {datos_usuario['direccion']} y su número de teléfono es {datos_usuario['telefono']}")
```

```
odadera Gonzalez Andres_SGE04_Tarea\Parte02 & C:/Users/andres/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/andres/OneDrive/Escritorio/DAM/SGE - sistemas de gestion empresarial/Unidad 4/Tarea/Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02/ejercicio11.py"
¿Cuál es tu nombre? Andrés Podadera
¿Cuál es tu edad? 37
¿Cuál es tu dirección? Calle Monda 7 Málaga
¿Cuál es tu número de teléfono? 12345678
Andrés Podadera tiene 37 años, vive en Calle Monda 7 Málaga y su número de teléfono es 12345678
```

12. Escribir una función que muestre por pantalla el saludo **¡Hola amiga!** cada vez que se la invoque.

The screenshot shows the VS Code interface. The Explorer sidebar shows the same 'PARTE02' folder structure as the previous screenshot. The 'ejercicio12.py' file is open in the main editor area. The code defines a function 'saludar' that prints a greeting. The terminal below shows the execution of the script and its output. A sidebar on the right displays the user's profile information.

```
def saludar(mensaje='¡Hola amiga!'):
    print(mensaje)

saludar()
saludar('¡Hola amiga!')
```

```
esarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02 > & C:/Users/andres/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/andres/OneDrive/Escritorio/DAM/SGE - sistemas de gestion empresarial/Unidad 4/Tarea/Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02/ejercicio12.py"
¡Hola amiga!
¡Hola amiga!
```

13. Escribir una función a la que se le pase una cadena **<nombre>** y muestre por pantalla el saludo **¡hola <nombre>!**

```
ejercicio13.py
1 def saludar(nombre):
2     print(f"¡Hola {nombre}!")
3 
4 saludar("Flor")
```

Terminal output:

```
PS C:\Users\andres\OneDrive\Escritorio\DAW\SGE - sistemas de gestion empresarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02> ejercicio13.py
¡Hola Flor!
```

14. Escribir una función que reciba un número entero positivo y devuelva su factorial.

```
ejercicio14.py
1 def factorial(n):
2     if n == 0 or n == 1:
3         return 1
4     return n * factorial(n - 1) # Llamada recursiva
5 
6 def pedir_entero_positivo(mensaje):
7     while True:
8         try:
9             num = int(input(mensaje))
10            if num < 0:
11                print("Error: Debes ingresar un número entero positivo")
12            else:
13                return num
14        except ValueError:
15            print("Error: Entrada no válida. Debes ingresar un número entero positivo")
16 
17 # Solicitan el número válido
18 num = pedir_entero_positivo("Introduce un número entero positivo: ")
19 print(f"El factorial de {num} es {factorial(num)}")
```

Terminal output:

```
Introduce un número entero positivo: 8,9
Error: Entrada no válida. Debes ingresar un número entero.
Introduce un número entero positivo: 9
El factorial de 9 es 362880
PS C:\Users\andres\OneDrive\Escritorio\DAW\SGE - sistemas de gestión empresarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02>
```

15. Depuración. Corregir los errores sintácticos del siguiente programa:

```
Ejercicio15.py
1 pwd= input('Introduce la contraseña: ')
2 if pwd in ['curso2324']:
3     print('Inicio de sesión realizado')
4 else:
5     print('Error en inicio de sesión')
```

The screenshot shows a Jupyter Notebook interface with the following details:

- EXPLORER:** Shows a tree view of files under "PARTE02", including "ejercicio01.py" through "ejercicio15.py".
- CELL:** Displays a code cell with Python code:

```
ejercicio15.py > ...
1 pwd= input('Introduce la contraseña: ') # sustituimos " por '
2 if pwd in ['curso2324']: # cambiamos ) por ]
3     print('Inicio de sesión realizado')
4 else: # añadimos los :
5     print('Error en inicio de sesión')
```
- TERMINAL:** Shows the output of the code execution:

```
PS C:\Users\andres\OneDrive\Escritorio\dam\SGE - sistemas de gestion empresarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02\ejercicio15.py"
Introduce la contraseña: prueba
Error en inicio de sesión
PS C:\Users\andres\OneDrive\Escritorio\dam\SGE - sistemas de gestion empresarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02\ejercicio15.py"
2> & C:/Users/andres/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/andres/OneDrive/Escritorio/dam/SGE - sistemas de gestion empresarial/Unidad 4/Tarea/Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea/Parte02/ejercicio15.py"
○ Introduce la contraseña: curso2324
Inicio de sesión realizado
PS C:\Users\andres\OneDrive\Escritorio\dam\SGE - sistemas de gestion empresarial\Unidad 4\Tarea\Podadera_Gonzalez_Andres_Samuel_SGE04_Tarea\Parte02\ejercicio15.py"
2> []
```
- SIDE BAR:** Includes a user profile section and a sidebar menu with options: Accesibilidad, Perfil, Calificaciones, Mensajes, Archivos privados, Preferencias, and Cerrar sesión.

Parte III - Creación de un módulo en Odoo

Antes de comenzar.

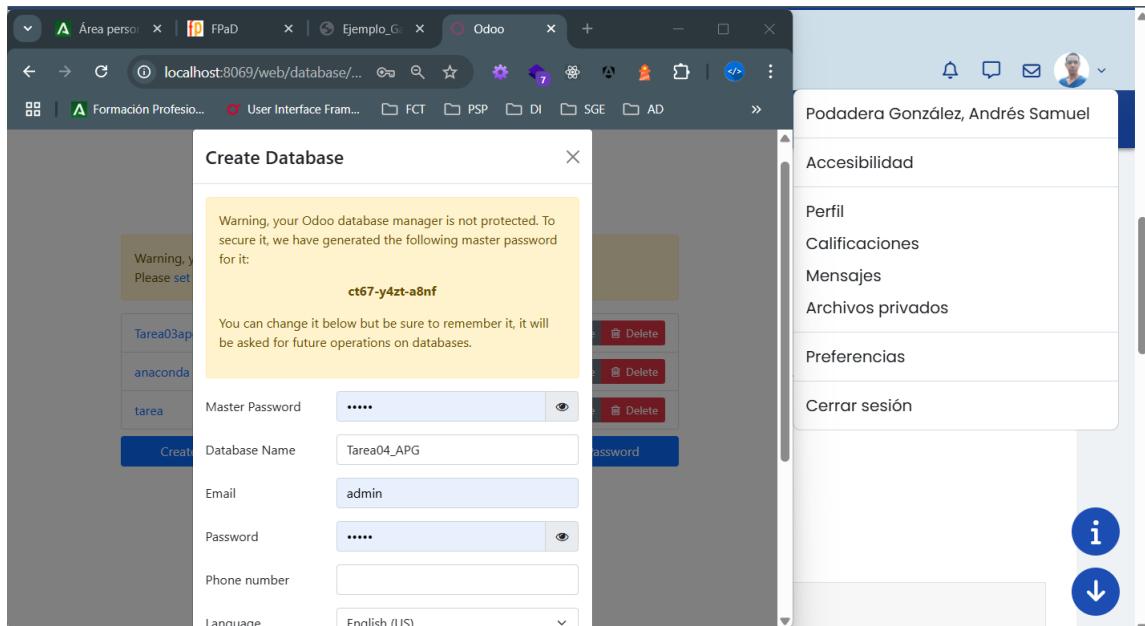
- Esta tarea la puedes realizar de tres formas:

Sobre tu versión de Odoo instalada en modo local con "localhost:8069" en tu navegador.

Sobre tu versión de Odoo instalada en la máquina virtual de Ubuntu modo cliente/servidor con "IPServidor:8069" en tu navegador.

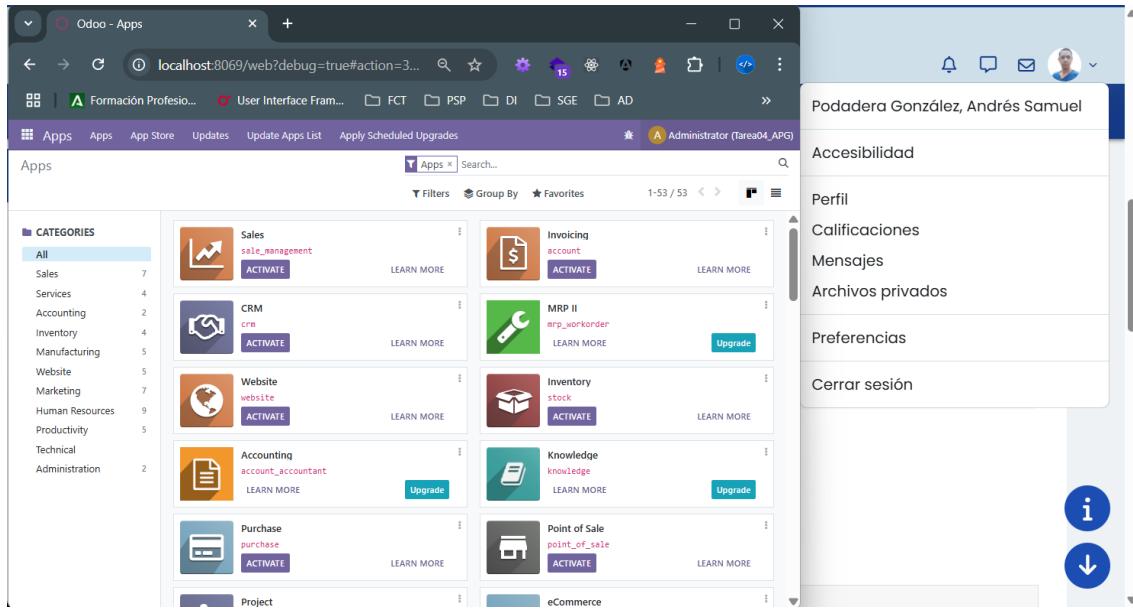
Sobre tu contenedor de Odoo creado con Docker con "localhost:8069" en tu navegador.

- Creación de la base de datos: desde la pantalla de login de Odoo, crea una nueva base de datos denominada **Tarea04_TusIniciales**, recuerda que no debes cargar los datos de prueba/demostración.



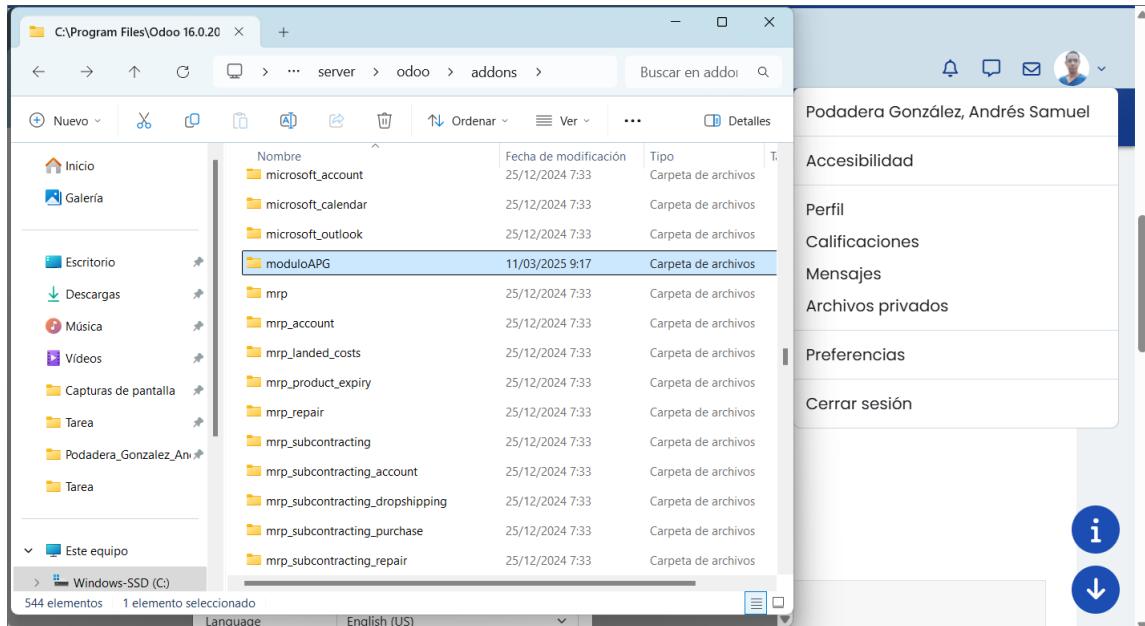
Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

- Realiza la tarea con el modo desarrollador activo.

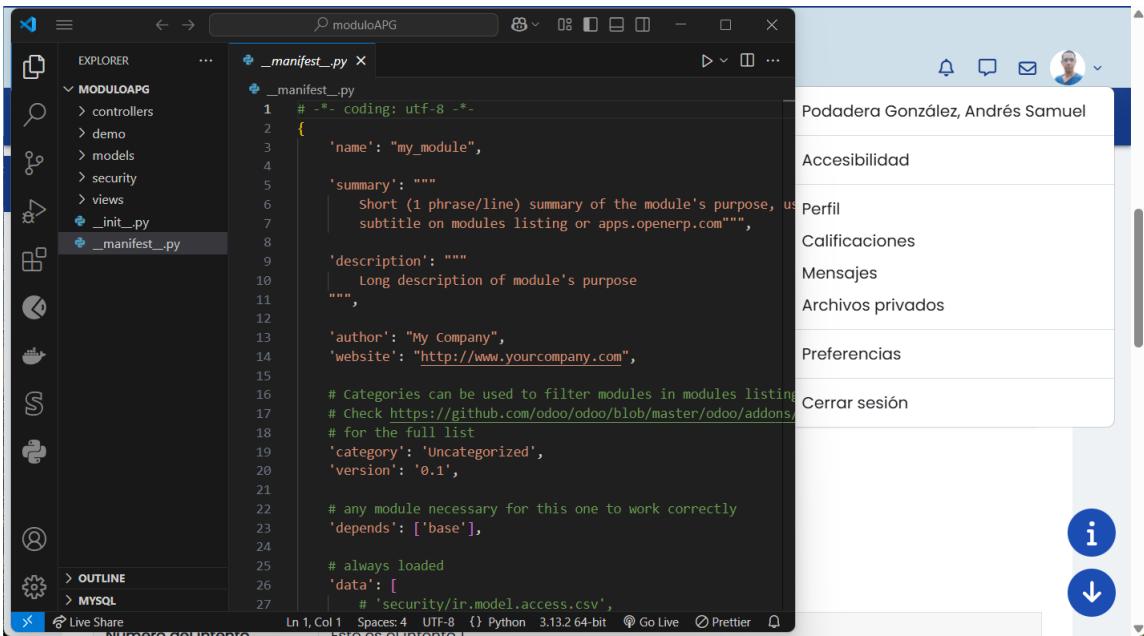


- Todos los apartados solicitados sobre la creación de un nuevo módulo en Odoo se debe hacer en código (Python, XML...) no será válido si se hace con la interfaz de Odoo.
Utilizaremos la interfaz de Odoo solo para la comprobación de lo realizado en código y cuando así se especifique en el enunciado.

- Siguiendo este [ejemplo](#) documentando la creación de un módulo para la gestión de un garaje, debes realizar un supuesto similar que cumpla los siguientes requerimientos que se indican en los siguientes apartados. (Descarga de aquí el [archivo .zip](#) con la estructura de un módulo para Odoo y sitúalo dentro de la carpeta **addons**)



Y empezamos a programar el módulo:



```
# -*- coding: utf-8 -*-
{
    'name': "my_module",
    'summary': """
        Short (1 phrase/line) summary of the module's purpose, used
        subtitle on modules listing or apps.openerp.com""",
    'description': """
        Long description of module's purpose
    """,
    'author': "My Company",
    'website': "http://www.yourcompany.com",
    'category': 'Uncategorized',
    'version': '0.1',
    'depends': [
        'base'
    ],
    'data': [
        # 'security/ir.model.access.csv',
    ]
}
```

Actividad 01: Datos del módulo

Modifica los archivos del módulo para responder a los siguientes requerimientos:

- Nombre de la carpeta: moduloXXX, donde XXX son la primera letra del nombre, apellido1 y apellido2 del alumno.
- Nombre del módulo (dentro de Manifest) : moduloXXX.
- Sumario o subtítulo del módulo: Módulo realizado por XXXXX, SGE 24-25, donde XXXXX es el nombre completo alumno.
- Categoría: Extra Tools
- Indicación de los demás datos del módulo: Versión, Licencia, Website, Autor.
- Otros datos del módulo: Deberá tener un ícono que lo represente.
- En la pestaña **Información** después de los datos del módulo, deberá aparecer lo siguiente: Mi primer módulo en Odoo 16, Features: Fácil, Sencillo, Intuitivo. Utiliza un formato similar al siguiente ejemplo:

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

The screenshot shows the Odoo application store interface. At the top, there are tabs for 'Aplicaciones', 'Actualizaciones', and a search bar. Below the header, the URL 'Aplicaciones / my_module' is visible, along with a status indicator '1 / 1' and a gear icon labeled 'Acción'. The main content area displays the module details for 'my_module' by 'My Company'. A large yellow hand-drawn circle highlights the title 'Mi primer modulo en Odoo16' and the 'Features' section. The 'Features' section lists three items: 'Facil', 'Sencillo', and 'Intuitivo', each preceded by a small orange icon.

Descripción larga de un módulo

Incluye una captura donde se muestre esta información del módulo nuevo desde Odoo.

Nuestro texto de descripción larga contiene:

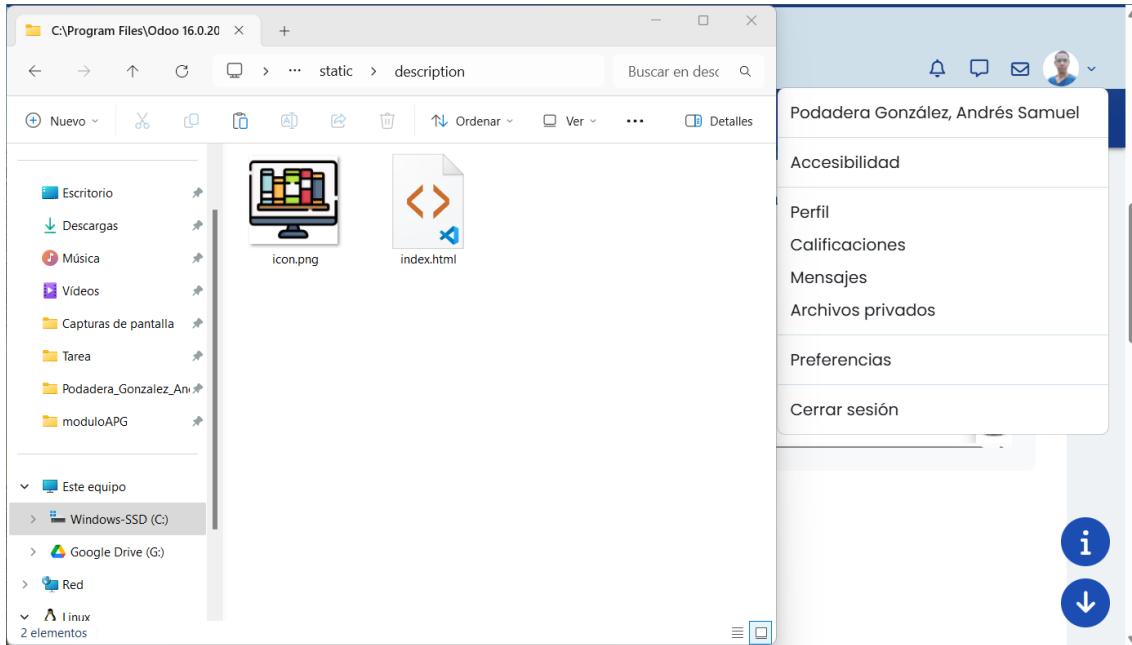
The screenshot shows a code editor with several tabs: '_manifest_.py', 'views.xml', and 'index.html'. The 'index.html' tab is active, displaying the following HTML code:

```
static > description > index.html > html > body > header > h2
1   <!DOCTYPE html>
2   <html lang="es">
3     <body>
4       <header>
5         <h1 style="text-align: center; color: purple">
6           Mi primer modulo en Odoo16
7         </h1>
8         <h2 style="text-align: center; color: seagreen">★ Features:</h2>
9         <p style="text-align: left; color: green">👉 Facil</p>
10        <p style="text-align: left; color: green">👉 Sencillo</p>
11        <p style="text-align: left; color: green">👉 Intuitivo</p>
12       </header>
13     </body>
14   </html>
```

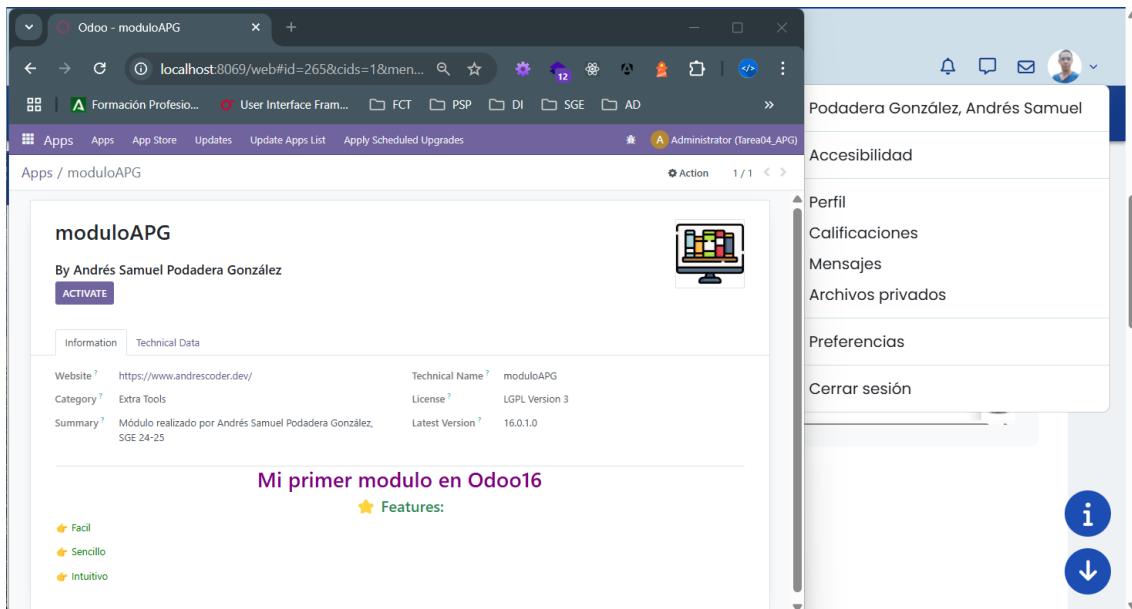
The right side of the screen shows a user profile sidebar with options like 'Podadera González, Andrés Samuel', 'Accesibilidad', 'Perfil', 'Calificaciones', 'Mensajes', 'Archivos privados', 'Preferencias', and 'Cerrar sesión'.

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

Tenemos que colocar tanto en icono como el index.html de descripción larga del módulo en la carpeta moduloAPG/static/description:



Reiniciamos el servidor de Odoo y actualizamos la lista de módulos para obtener:



Actividad 02: Modelo

- Clase **Biblioteca** con campos **nombre** (Char) y **capacidad** (Integer).
- Clase **Libro** con campos **nombre** (Char), **publicado** (date), **precio**(Float), **genero**, a seleccionar entre Ficción, Novela, Poesía.
- Clase **Autor** con campos **nombre** (char), **nacionalidad** (char)

Añade las siguientes relaciones entre las clases:

- **One2many:** Una biblioteca puede tener muchos libros

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

- **Many2one:** Un libro pertenece a una sola biblioteca
- **Many2many:** Un libro puede tener muchos autores, un autor puede escribir muchos libros.

Incluye una captura de pantalla del menú **Ajustes/Técnico/Estructura de la base de datos/Modelos**, donde se muestre el modelo creado con los diferentes campos.

The screenshot shows the Odoo Models menu. On the left, there's a sidebar with 'Settings', 'Users & Companies', 'Translations', and 'Technical'. The main area is titled 'Models' and shows a table with three rows:

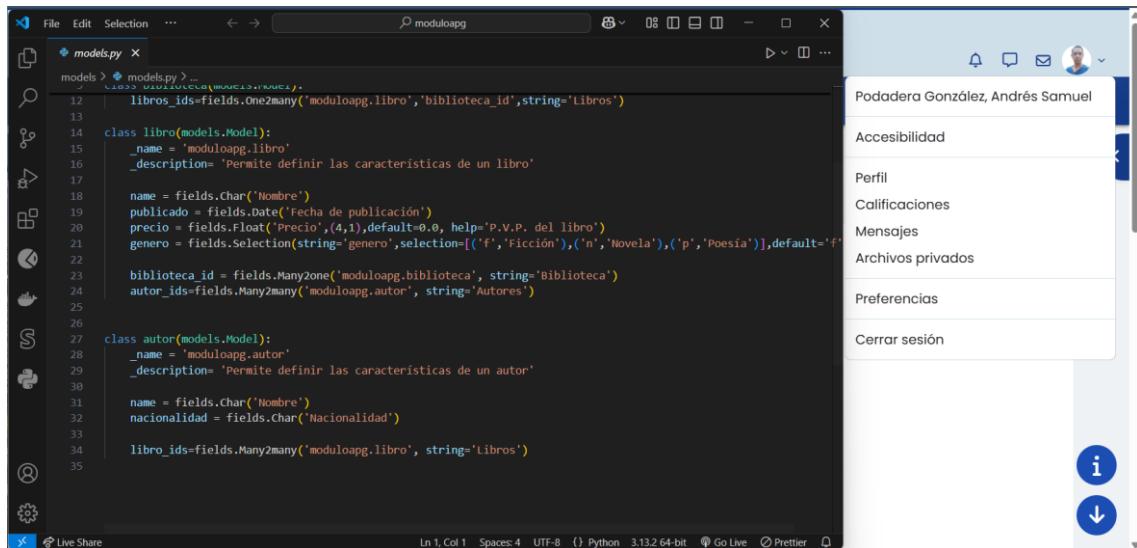
Model	Model Description	Type	Transient Model
moduloapg.autor	Permite definir las características de un autor	Base Object	<input type="checkbox"/>
moduloapg.biblioteca	Permite definir las características de una biblioteca	Base Object	<input type="checkbox"/>
moduloapg.libro	Permite definir las características de un libro	Base Object	<input type="checkbox"/>

Biblioteca:

```
models.py
models > models.py > ...
1 # -*- coding: utf-8 -*-
2
3 from odoo import models, fields, api # type: ignore
4
5 class biblioteca(models.Model):
6     _name = 'moduloapg.biblioteca'
7     _description= 'Permite definir las características de una biblioteca'
8
9     name = fields.Char('Nombre')
10    capacidad = fields.Integer('Capacidad')
11
12    libros_ids=fields.One2many('moduloapg.libro','biblioteca_id',string='Libros')
13
14 class libro(models.Model):
15     _name = 'moduloapg.libro'
16     _description= 'Permite definir las características de un libro'
17
18     name = fields.Char('Nombre')
19     publicado = fields.Date('Fecha de publicación')
20     precio = fields.Float('Precio',(4,2),default=0.0, help='P.V.P. del libro')
21     genero = fields.Selection(string='genero',selection=[('f','Ficción'),('n','Novela'),('p','Poesía')],default='f')
22
23     biblioteca_id = fields.Many2one('moduloapg.biblioteca', string='Biblioteca')
24     autor_ids=fields.Many2many('moduloapg.autor', string='Autores')
25
26
27 class autor(models.Model):
28     _name = 'moduloapg.autor'
```

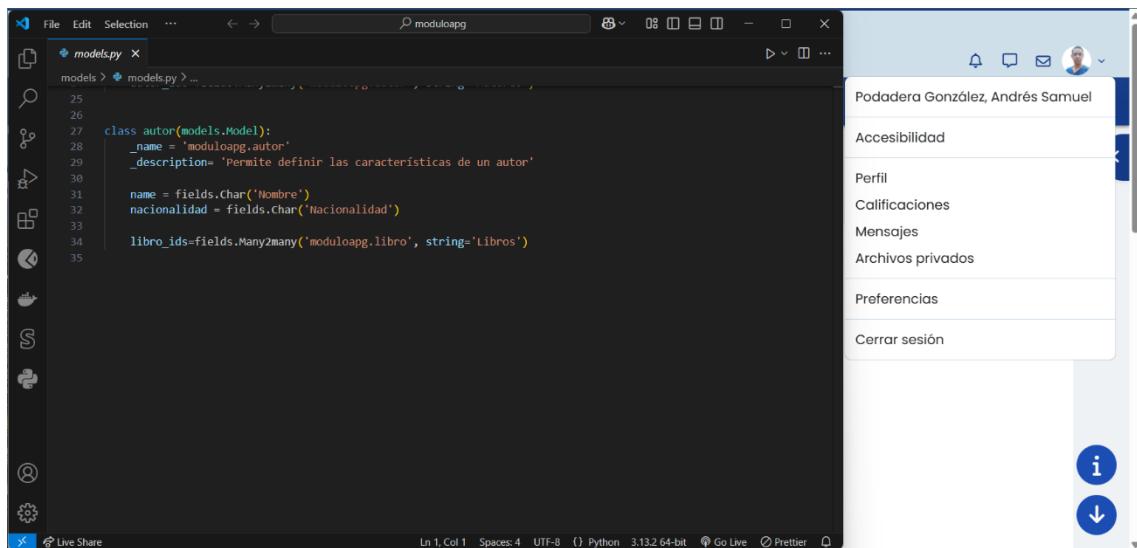
Libro:

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025



```
models > models.py > ...
12 |     libros_ids=fields.OneToMany('moduloapg.libro','biblioteca_id',string='Libros')
13 |
14 | class libro(models.Model):
15 |     _name = 'moduloapg.libro'
16 |     _description= 'Permite definir las características de un libro'
17 |
18 |     name = fields.Char('Nombre')
19 |     publicado = fields.Date('Fecha de publicación')
20 |     precio = fields.Float('Precio',(4,1),default=0.0, help='P.V.P. del libro')
21 |     genero = fields.Selection(string='genero',selection=[('f','Ficción'),('n','Novela'),('p','Poesía')],default='f')
22 |
23 |     biblioteca_id = fields.ManyToOne('moduloapg.biblioteca', string='Biblioteca')
24 |     autor_ids=fields.ManyToMany('moduloapg.autor', string='Autores')
25 |
26 |
27 | class autor(models.Model):
28 |     _name = 'moduloapg.autor'
29 |     _description= 'Permite definir las características de un autor'
30 |
31 |     name = fields.Char('Nombre')
32 |     nacionalidad = fields.Char('Nacionalidad')
33 |
34 |     libro_ids=fields.ManyToMany('moduloapg.libro', string='Libros')
35 
```

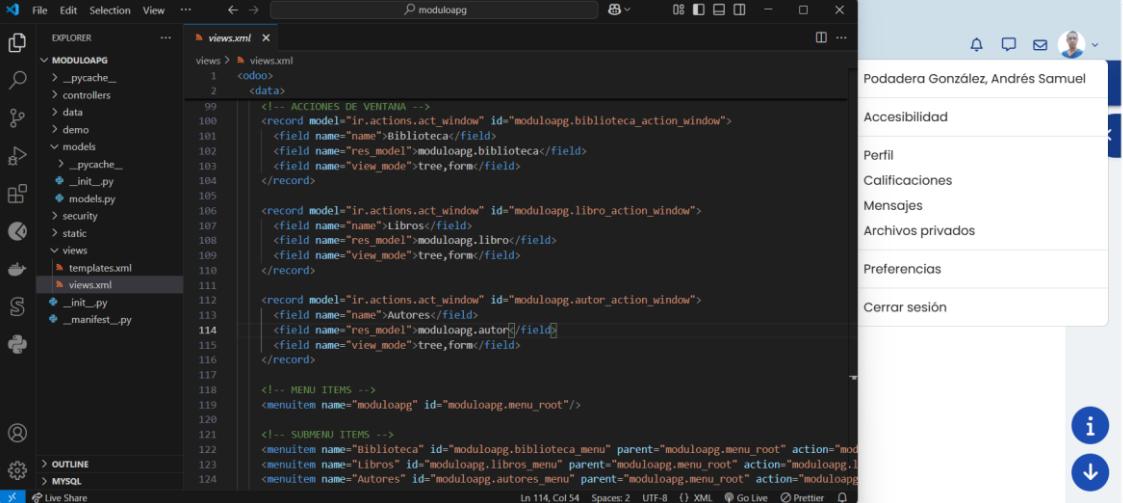
Autor:



```
models > models.py > ...
25 |
26 |
27 | class autor(models.Model):
28 |     _name = 'moduloapg.autor'
29 |     _description= 'Permite definir las características de un autor'
30 |
31 |     name = fields.Char('Nombre')
32 |     nacionalidad = fields.Char('Nacionalidad')
33 |
34 |     libro_ids=fields.ManyToMany('moduloapg.libro', string='Libros')
35 
```

Actividad 03: Menú

- Crea una opción de menú para tu módulo y los submenús necesarios según los modelos creados. Además, debes crear una acción de ventana para cada uno.



```

<!-- ACCIONES DE VENTANA -->
<record model="ir.actions.act_window" id="moduloapg.biblioteca_action_window">
    <field name="name">Biblioteca</field>
    <field name="res_model">moduloapg.biblioteca</field>
    <field name="view_mode">tree,form</field>
</record>

<record model="ir.actions.act_window" id="moduloapg.libro_action_window">
    <field name="name">Libros</field>
    <field name="res_model">moduloapg.libro</field>
    <field name="view_mode">tree,form</field>
</record>

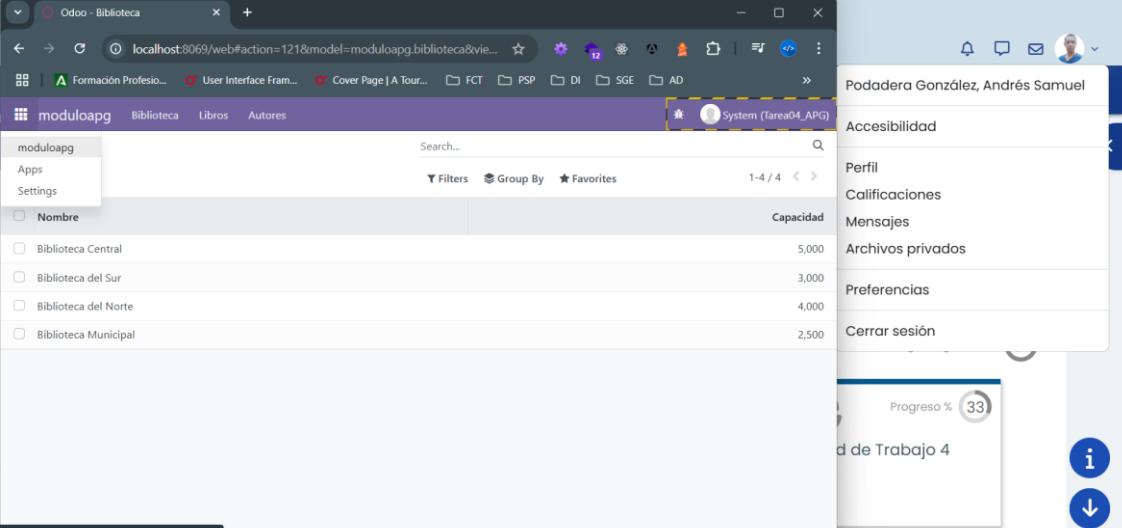
<record model="ir.actions.act_window" id="moduloapg.autor_action_window">
    <field name="name">Autores</field>
    <field name="res_model">moduloapg.autor</field>
    <field name="view_mode">tree,form</field>
</record>

<!-- MENU ITEMS -->
<menuitem name="moduloapg" id="moduloapg.menu_root"/>

<!-- SUBMENU ITEMS -->
<menuitem name="Biblioteca" id="moduloapg.biblioteca_menu" parent="moduloapg.menu_root" action="moduloapg.biblioteca_action_window"/>
<menuitem name="Libros" id="moduloapg.libros_menu" parent="moduloapg.menu_root" action="moduloapg.libro_action_window"/>
<menuitem name="Autores" id="moduloapg.autores_menu" parent="moduloapg.menu_root" action="moduloapg.autor_action_window"/>

```

Cambia el rol a superusuario (como se indica en el documento de ayuda) y haz una captura de la barra de menú donde se muestren tus opciones nuevas.



Nombre	Capacidad
Biblioteca Central	5,000
Biblioteca del Sur	3,000
Biblioteca del Norte	4,000
Biblioteca Municipal	2,500

ACTIVIDAD 4: Vistas

- Crea una vista árbol (lista) y formulario para cada uno de los modelos.

Incluye una captura de pantalla de cómo se visualizan estas vistas en Odoo (no el código XML). Para poder visualizar las vistas debes tener configurados los menús de forma correcta.

Biblioteca (Tipo árbol):

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

The screenshot shows a list view of library records (Biblioteca) in Odoo. The columns are 'Nombre' (Name) and 'Capacidad' (Capacity). The data is as follows:

Nombre	Capacidad
Biblioteca Central	5,000
Biblioteca del Sur	3,000
Biblioteca del Norte	4,000
Biblioteca Municipal	2,500

Biblioteca(Tipo formulario):

The screenshot shows a form view of a library record ('Biblioteca Central'). The fields are:

- Nombre: Biblioteca Central
- Capacidad: 5,000
- Libros:

Nombre	Fecha de publicación	Precio genero
Cien años de soledad	05/30/1967	19.9 Ficción
Don Quijote de la Mancha	01/16/1605	25.0 Ficción

Libro(Tipo árbol):

The screenshot shows a list view of book records (Libros). The columns are 'Nombre' (Name), 'Fecha de publicación' (Publication Date), and 'Precio genero' (Price Type). The data is as follows:

Nombre	Fecha de publicación	Precio genero
Cien años de soledad	05/30/1967	19.9 Ficción
Rayuela	06/28/1963	15.5 Novela
La casa de los espíritus	01/01/1982	17.0 Novela
La ciudad y los perros	01/01/1962	14.0 Novela
Ficciones	01/01/1944	12.0 Ficción
Don Quijote de la Mancha	01/16/1605	25.0 Ficción
Veinte poemas de amor y una canción desesperada	01/01/1924	10.0 Poesía
Romancero Gitano	01/01/1928	9.5 Poesía

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

Libro(Tipo formulario):

The screenshot shows a browser window for the Odoo library module. The URL is `localhost:8069/web#id=1&cids=1&menu_id=106&action=122&mo...`. The page displays a book record for "Cien años de soledad". The details are as follows:

Nombre	Cien años de soledad				
Fecha de publicación	05/30/1967				
Precio	19.9				
genero	Ficción				
Biblioteca	Biblioteca Central				
Autores	<table border="1"><thead><tr><th>Nombre</th><th>Nacionalidad</th></tr></thead><tbody><tr><td>Gabriel García Márquez</td><td>Colombiana</td></tr></tbody></table>	Nombre	Nacionalidad	Gabriel García Márquez	Colombiana
Nombre	Nacionalidad				
Gabriel García Márquez	Colombiana				

A sidebar on the right shows the user profile of "Podadera González, Andrés Samuel" with options like Accesibilidad, Perfil, Calificaciones, etc.

Autor(Tipo árbol):

The screenshot shows a browser window for the Odoo author module. The URL is `localhost:8069/web?action=123&model=moduloapg.autor&view_ty...`. The page displays a list of authors:

Nombre	Nacionalidad
Gabriel García Márquez	Colombiana
Julio Cortázar	Argentina
Isabel Allende	Chilena
Mario Vargas Llosa	Peruana
Jorge Luis Borges	Argentina
Miguel de Cervantes	Española
Pablo Neruda	Chilena
Federico García Lorca	Española

A sidebar on the right shows the user profile of "Podadera González, Andrés Samuel" with options like Accesibilidad, Perfil, Calificaciones, etc.

Autor(Tipo formulario):

Nombre: Andrés Samuel Podadera González
Ciclo: CFGS Desarrollo de aplicaciones multiplataforma
Curso: 2024/2025

The screenshot shows the Odoo web interface for a module named 'moduloapg'. A sidebar on the right displays user information: 'Podadera González, Andrés Samuel', 'Accesibilidad', 'Perfil', 'Calificaciones', 'Mensajes', 'Archivos privados', 'Preferencias', and 'Cerrar sesión'. The main content area shows an author record for 'Gabriel García Márquez'. The record includes fields: Nombre (Gabriel García Márquez), Nacionalidad (Colombiana), and Libros (with one entry: Nombre (Cien años de soledad), Fecha de publicación (05/30/1967), Precio genero (19.9 Ficción)).

ACTIVIDAD 5: Datos precargados

Incluye un fichero "data.xml" en el que incluyas, al menos, dos elementos de cada uno de los modelos creados.

Incluye una captura de pantalla del contenido del archivo "data.xml".

The screenshot shows a code editor displaying the 'data.xml' file from the 'moduloapg' module. The XML file defines four records for the 'biblioteca' model and one record for the 'autor' model. The 'biblioteca' records are: 'Biblioteca Central' (capacity 5000), 'Biblioteca del Sur' (capacity 3000), 'Biblioteca del Norte' (capacity 4000), and 'Biblioteca Municipal' (capacity 2500). The 'autor' record is for 'Gabriel García Márquez' (nationality Colombian).

```
<?xml version="1.0" encoding="UTF-8"?>
<odoo>
    <!-- Datos de Bibliotecas -->
    <record id="biblioteca_1" model="moduloapg.biblioteca">
        <field name="name">Biblioteca Central</field>
        <field name="capacidad">5000</field>
    </record>
    <record id="biblioteca_2" model="moduloapg.biblioteca">
        <field name="name">Biblioteca del Sur</field>
        <field name="capacidad">3000</field>
    </record>
    <record id="biblioteca_3" model="moduloapg.biblioteca">
        <field name="name">Biblioteca del Norte</field>
        <field name="capacidad">4000</field>
    </record>
    <record id="biblioteca_4" model="moduloapg.biblioteca">
        <field name="name">Biblioteca Municipal</field>
        <field name="capacidad">2500</field>
    </record>
    <!-- Datos de Autores -->
    <record id="autor_1" model="moduloapg.autor">
        <field name="name">Gabriel García Márquez</field>
        <field name="nacionalidad">Colombiana</field>
    </record>
</odoo>
```

Debes incluir en tu entrega un archivo moduloXXX.zip que contenga tu módulo listo para cargarlo en Odoo.