

SPRINT 7

Edoardo Brega

★ NIVELL 1

→ exercici 1

- **Calculadora de l'índex de massa corporal:**

Escriu una funció que calculi l'IMC ingressat per l'usuari/ària, és a dir, qui ho executi haurà d'ingressar aquestes dades.

"""IMC"""

Función para calcular el Índice de Masa Corporal (IMC) y determinar su categoría
def calc_imc(peso, altura):

Creo la variable imc dividiendo el peso por la altura elevada al cuadrado
imc = peso / (altura ** 2)

Controlo si el IMC es menor de 18.5 y creo la variable cat = "Bajo Peso"
if imc < 18.5: cat = "Bajo Peso"

Controlo si el IMC es entre 18.5 y 25 y creo la variable cat = "Peso normal"
elif 18.5 <= imc < 25: cat = "Peso Normal"

Controlo si el IMC es entre 25 y 30 y creo la variable cat = "Sobrepeso"
elif 25 <= imc < 30: cat = "Sobrepeso"

Controlo si el IMC es mayor de 30 y creo la variable cat = "Obesidad"
else: cat = "Obesidad"

Devuelve el valor del imc y la categoría correspondiente
return imc, cat

Función para ingresar los datos por el usuario de peso y altura
def obtener_datos_imc():

Bucle para el peso
while True:

Mensaje para el input del peso
**peso = input("Ingresa el peso en 'Kg' o presiona ESC para volver al principio: ").replace(",",
".").lower().replace("kg", "")**

try:

Controlo si el valor ingresado es un número
valor_peso = float(peso)

Verifico si el peso ingresado está dentro del rango permitido (30-250 kg)
if 29.9 <= valor_peso <= 250.1: break

Muestro un mensaje de error si el peso no está dentro del rango
**else: print(f"El valor ingresado para el peso es {peso}. La masa corporal se certifica solo para valores
entre 30 kg y 250 kg. Ingresa otra vez el peso.")**

except ValueError:

Gestiono el error en el caso que el peso no se haya reconocido como número

Si se ha presionado "ESC", termina la función sin devolver valores
if keyboard.is_pressed('esc'): print("Has presionado Esc. ") ; return None, None

```

# Si no se ha ingresado nada, vuelvo al empezar del bucle del peso
elif not peso: print("No has ingresado el peso.")

# Si el valor no es un número, muestro un mensaje de error y vuelvo al empezar del bucle del peso
else: print(f"El valor ingresado no es valido: {peso}. Ingresa otra vez el peso.")

# Bucle para la altura
while True:

    # Mensaje para el input de la altura
    altura = input("Ingresa la altura en 'm' o presiona ESC para volver al principio: ").replace(" ",
".").lower().replace("m", "")

    try:

        # Controlo si el valor ingresado es un número
        valor_altura = float(altura)

        # Verifico si la altura ingresada está dentro del rango permitido (1-2.5 m)
        if 0.9 < valor_altura < 2.6: break

        # Muestra un mensaje de error si la altura no está dentro del rango
        else: print(f"El valor ingresado para la altura es {valor_altura}. La masa corporal se certifica solo para
valores entre 1 m y 2.5 m. Ingresa otra vez la altura.")

    except ValueError:

        # Si se ha presionado "ESC", termina la función sin devolver valores
        if keyboard.is_pressed('esc'): print("Has presionado Esc ") ; return None, None

        # Si no se ha ingresado nada, vuelvo al empezar del bucle de la altura
        elif not altura: print("No has ingresado la altura")

        # Si el valor no es un número, muestro un mensaje de error y vuelvo al empezar del bucle de la altura
        else: print(f"El valor ingresado no es valido: {altura}. Ingresa otra vez la altura.")

# Devuelvo el peso y la altura validados
return valor_peso, valor_altura

# Función principal
def main_imc():

    # Mensaje de introducción al programa
    print("Programa para el calculo del Índice de masa corporal.\n")

    # Bucle para empezar el programa
    while True:

        # Pido al usuario si quiere calcular un IMC
        respuesta = input("Quieres calcular el IMC de una persona? (y/n): ").strip().lower()

        # Si se ha presionado "ESC" o 'n', salgo del bucle y termino el programa
        if keyboard.is_pressed('esc') or respuesta == 'n': print("Salida del programa.") ; break

        # Si se ha presionado 'y', llamo la primera función
        elif respuesta == "y":

            # Creo las variables peso y altura con los valores devueltos de la función obtener_datos_imc()
            peso, altura = obtener_datos_imc()

            # Controlo que la función de obtener datos haya devuelto valores, si no vuelvo al empezar del bucle
            if peso is not None and altura is not None:

                # Creo las variables imc y categoria con los valores devueltos de la función calc_imc
                imc, categoria = calc_imc(peso, altura)

```

```
# Muestro el resultado y salgo del bucle
print(f"Datos de peso y altura recibidos: {peso}, {altura}\nIMC = {imc:.2f}\nCategoria = {categoria}."); break

# Si no se ha presionado ni 'Esc' ni 'y' ni 'n' muestro un mensaje de error y vuelvo al empezar de bucle
else: print("Respuesta no reconocida. Presiona 'y' o 'n' o 'Esc'.")

# Importa la biblioteca para reconocer si se presiona 'Esc'
import keyboard

# Función principal para iniciar el programa
main_imc()
```

→ exercici 2

- **Convertidor de temperatures:**

Existeixen diverses unitats de temperatura utilitzades en diferents contextos i regions. Les més comunes són Celsius (°C), Fahrenheit (°F) i Kelvin (K). També existeixen altres unitats com Rankine (°Ra) i Réaumur (°Re). Selecciona almenys 2 conversors, de tal manera que en introduir una temperatura retorni, com a mínim, dues conversions.

```
"""Temperaturas"""
```

```
# Funciones de conversión
```

```
def celsius_2_fahrenheit(celsius): return (celsius * 9/5) + 32
```

```
def celsius_2_kelvin(celsius): return celsius + 273.15
```

```
def fahrenheit_2_celsius(fahrenheit): return (fahrenheit - 32) * 5/9
```

```
def fahrenheit_2_kelvin(fahrenheit): return (fahrenheit - 32) * 5 / 9 + 273.15
```

```
def kelvin_2_celsius(kelvin): return kelvin - 273.15
```

```
def kelvin_2_fahrenheit(kelvin): return (kelvin - 273.15) * 9 / 5 + 32
```

```
# Función para la conversión según la unidad de medida ingresada
```

```
def convertir_temperatura(temp_in):
```

```
    # Cada 'if' reconoce el ultimo caracter 'temp_in[-1]' y devuelve una cadena de texto llamando a las funciones correspondientes pasando el valor de la temperatura como número sin la unidad de medida 'float(temp_in[:-1])' y lo muestra con dos decimales con '.2f'
```

```
    if temp_in[-1].upper() == 'C': return f"{float(temp_in[:-1])}° Celsius equivalen a :  
{celsius_2_fahrenheit(float(temp_in[:-1])):.2f}° Fahrenheit y equivalen a :  
{celsius_2_kelvin(float(temp_in[:-1])):.2f}° Kelvin."
```

```
    elif temp_in[-1].upper() == 'F': return f"{float(temp_in[:-1])}° Fahrenheit equivalen a :  
{fahrenheit_2_celsius(float(temp_in[:-1])):.2f}° Celsius y equivalen a :  
{fahrenheit_2_kelvin(float(temp_in[:-1])):.2f}° Kelvin."
```

```
    elif temp_in[-1].upper() == 'K': return f"{float(temp_in[:-1])}° Kelvin equivalen a :  
{kelvin_2_celsius(float(temp_in[:-1])):.2f}° Celsius y equivalen a :  
{kelvin_2_fahrenheit(float(temp_in[:-1])):.2f}° Fahrenheit."
```

```
# Función para obtener la temperatura y controlar el valor ingresado
```

```
def obtener_temperatura():
```

```
    # Bucle para pedir la temperatura y controlar el valor ingresado
```

```
    while True:
```

```
        # Pido al usuario que ingrese la temperatura
```

```
        temp_in = input("Ingresa la temperatura que quieres convertir con su unidad de medida\n(ejemplos: 25C, 77F, 300K): ").strip().replace(' ', '.').replace("°", "")
```

```
        # Si el usuario presiona 'Esc' sale del bucle y no devuelve ningun valor a la función principal  
        if keyboard.is_pressed('esc'): print("Has presionado Esc.") ; return None
```

```
        # Si no se ha ingresado nada vuelve al bucle
```

```
        elif not temp_in: print("No has ingresado la temperatura.")
```

```
        # Controlo si el valor ingresado es de un solo carácter, podría dar problemas ingresando solo la unidad de medida
```

```
        elif len(temp_in) == 1: print(f"El valor ingresado no es valido: ' {temp_in} ', recuerda ingresar la temperatura seguida de su unidad de medida (ejemplos: 25C, 77F, 300K)")
```

```
        # Controlo que el último carácter sea la unidad de medida
```

```
        elif temp_in[-1].upper() not in ['C', 'F', 'K']: print("Unidad de medida no reconocida. Recuerda añadir C, F o K despues del valor de la temperatura.")
```

```
        # Si se han pasado los otros controles sigo intentado convertir el valor ingresado a un número excluyendo el último carácter
```

```

else:

    try: temp_valor = float(temp_in[:-1])

    # Si el valor sin el ultimo caracter no es un numero vuelvo al empezar del bucle
    except ValueError: print("Ingresa un valor numerico válido para la temperatura seguido de su unidad
de medida.")

    # Verifica el rango para cada unidad de medida
    if temp_in[-1].upper() == 'C' and -273.15 <= temp_valor <= 1000: return temp_in

    elif temp_in[-1].upper() == 'F' and -459.67 <= temp_valor <= 1832: return temp_in

    elif temp_in[-1].upper() == 'K' and 0 <= temp_valor <= 1273.15: return temp_in

    else:

        # Muestro un mensaje de error por cada unidad de medida para valores fuera del rango
        if temp_in[-1].upper() == 'C': print("Las temperaturas en Celsius tienen que ser entre -273.15°C y
1000°C.")

        if temp_in[-1].upper() == 'F': print("Las temperaturas en Fahrenheit tienen que ser entre -459.67°F y
1832°F.")

        if temp_in[-1].upper() == 'K': print("Las temperaturas en Kelvin tienen que ser entre 0K y 1273.15K.")

# Función principal
def main_temp():

    # Muestra un mensaje de inicio
    print("Programa para la conversion de temperaturas entre Celsius (c), Fahrenheit (f) e Kelvin (k):\n")

    # Bucle para pedir al usuario si quiere hacer una conversión
    while True:

        respuesta = input("Quieres convertir temperaturas entre Celsius (c), Fahrenheit (f) e Kelvin (k)? (y/n):
").strip().lower()

        # Si el usuario presiona 'Esc' o 'n' salgo del bucle y se acaba el programa
        if keyboard.is_pressed('esc') or respuesta == 'n': print("Salida del programa.") ; break

        # Si el usuario presiona 'y' el programa sigue llamando a la función
        elif respuesta == 'y':

            # Creo la variable temp_in con el valor devuelto por la función obtener_temperatura
            temp_in = obtener_temperatura()

            # Si la función ha devuelto un valor, llamo la función de conversion pasandole el valor temp_in y la muestro
            en pantalla
            if temp_in is not None: print(convertir_temperatura(temp_in)) ; break

            # Si el usuario no ha presionado ni 'y' ni 'n' ni 'Esc' muestro un mensaje de error y vuelvo al empezar del bucle
            else: print("Respuesta no reconocida. Presiona 'y' o 'n' o 'Esc'.")

# Biblioteca para la presión de 'Esc'
import keyboard

main_temp()

```

→ exercici 3

- **Comptador de paraules d'un text:**

Escriu una funció que donat un text, mostri les vegades que apareix cada paraula.

""Cuenta Palabras""

Función que contará las palabras en un texto

def comptador_paraules(texto):

Convierto el texto a minúsculas, elimino puntuación y caracteres especiales reemplazándolos por espacios, y convierto texto en una lista de valores separados por el espacio

palabras = texto.lower().translate(str.maketrans(string.punctuation + "¡¿", ' ' * (len(string.punctuation) + 2))).split()

Creo un diccionario donde las claves serán las palabras y los valores serán el numero que cada palabra aparece

contador = {}

Recorro cada palabra en la lista, con '.get' busco cada palabra y si no se encuentra ya en el diccionario le doy valor 0, y en todos caso sumo 1 a su valor

for word in palabras: contador[word] = contador.get(word, 0) + 1

La función devuelve el diccionario con las palabras y las veces que aparecen

return contador

Función para pedir el texto al usuario

def obtener_texto():

Bucle para pedir el texto

while True:

Solicita al usuario que introduzca un texto

texto_in = input("Escribe el texto:\n")

Si el usuario presiona 'Esc' salgo de la función sin devolver nada

if keyboard.is_pressed('esc'): print("Has presionado Esc."); return None

Si el usuario no ha ingresado ningún texto, vuelvo al empezar del bucle

elif not texto_in: print("No has ingresado ningun texto")

Si se ha ingresado un texto, la función lo devuelve

else: return texto_in

Función principal

def main_cuenta_palabras():

Mensaje inicial

print("Programa para contar las veces que aparece cada palabra o cada numero en un texto")

Bucle para pedir al usuario si quiere ingresar un texto y contar las palabras

while True:

respuesta = input("Quieres ingresar un texto para contar las palabras? (y/n): ").strip().lower()

Si el usuario presiona 'Esc' o 'n' salgo del bucle y se acaba el programa

if keyboard.is_pressed('esc') or respuesta == 'n': print("Salida del programa "); break

Si el usuario presiona 'y' llamo a la función para obtener el texto

elif respuesta == 'y':

Creo la variable texto_in con el valor devuelto de la función obtener_texto

texto_in = obtener_texto()

```

# Si la función ha devuelto un texto lo muestro en pantalla y llamo a la función para el conteo
if texto_in is not None:

    print(f"\nHas ingresado:\n{texto_in}\n\nResultado:")

    # Recorro el diccionario devuelto de la función comptador_paraules tratándolo por parejas de clave/valor
    con '.items' y lo muestro en pantalla ordenado por frecuencia y alfabéticamente
    for palabra, count in sorted(comptador_paraules(texto_in).items(), key=lambda item: (-item[1],
item[0])):
        print(f"'{palabra}' = {count}")

    break

# Si el usuario no ha presionado ni 'y' ni 'n' ni 'Esc' muestro un mensaje de error y vuelvo al empezar del bucle
else: print("Respuesta no reconocida. Presiona 'y' o 'n'.")

# La biblioteca string es para quitar la puntuación del texto, keyboard es para la presión de 'Esc'
import string
import keyboard

main_cuenta_palabras()

```


→ exercici 4

- **Diccionari invers:**

Resulta que el client té una enquesta molt antiga que s'emmagatzema en un diccionari i els resultats els necessita al revés, és a dir, intercanviats les claus i els valors. Els valors i claus en el diccionari original són únics; si aquest no és el cas, la funció hauria d'imprimir un missatge d'avertiment.

```
"""Diccionario inverso con mensaje de error"""

# Función para invertir un diccionario
def diccionario_inverso(diccionario_original):

    # Creo un diccionario para almacenar los valores invertidos
    diccionario_inv = {}

    # Recorro el diccionario original tratándolo por parejas de clave/valor con '.items'
    for key, value in diccionario_original.items():

        # Si un valor ya está en el diccionario invertido como clave, salgo de la función sin devolver el diccionario
        if value in diccionario_inv: return None

        # Si el valor no está ya como clave en el diccionario invertido lo añado con su clave como valor
        else: diccionario_inv[value] = key

    # Si ningún valor ha sido encontrado doble al ser transformado en clave la función devuelve el diccionario
    return diccionario_inv

# Función principal
def main_invertir_diccionario_con_error():

    # Diccionarios de ejemplo
    diccionario1 = {'a': 1, 'b': 2, 'c': 3} # Diccionario sin valores repetidos
    diccionario2 = {'x': 'apple', 'y': 'banana', 'z': 'banana'} # Diccionario con valores repetidos

    # Mensaje de inicio del programa
    print("Programa para invertir un diccionario dado con mensaje de error en caso de llaves dobles\n")

    # Por cada diccionario de ejemplo llamo a la función de inversión
    for diccionario in [diccionario1, diccionario2]:

        # Creo la variable diccionario_inv con el valor devuelto por la función
        diccionario_inv = diccionario_inverso(diccionario)

        # Si la función devuelve un diccionario, lo imprime
        if diccionario_inv: print(f"Diccionario original:\n{diccionario}\nDiccionario\ninvertido:\n{diccionario_inv}\n\n")

        # Si la función no ha devuelto un diccionario significa que ha encontrado valores por transformar en claves dobles y muestro un mensaje de error, indica un error debido a valores repetidos
        else: print(f"Diccionario original:\n{diccionario}\nError: multiple keys for one value\n")

    main_invertir_diccionario_con_error()
```

★ NIVELL 2

→ exercici 1

- **Diccionari invers amb duplicats:**

Continuant amb l'exercici 4 del nivell 1: al client es va oblidar de comentar un detall i resulta que els valors en el diccionari original poden duplicar-se i més, per la qual cosa les claus intercanviades poden tenir duplicats. En aquest cas, en l'exercici anterior imprimies un missatge d'avertiment, ara, els valors del diccionari resultant hauran d'emmagatzemar-se com una llista. Tingues en compte que si és un valor únic no ha de ser una llista.

```
"""Diccionario inverso con lista de dobles"""
```

```
# Función para invertir un diccionario
```

```
def diccionario_inverso(diccionario_original):
```

```
    # Creo un diccionario para almacenar los valores invertidos
```

```
    diccionario_inv = {}
```

```
    # Itera a través de los pares clave-valor del diccionario original
```

```
    for key, value in diccionario_original.items():
```

```
        # Controllo si el valor ya está en el diccionario invertido como clave
```

```
        if value in diccionario_inv:
```

```
            # Controllo si el valor asociado al valor puesto como clave ya es una lista y lo añado
```

```
            if isinstance(diccionario_inv[value], list): diccionario_inv[value].append(key)
```

```
            # Si no es ya una lista, convierto el valor existente en una lista y lo añado
```

```
            else: diccionario_inv[value] = [diccionario_inv[value], key]
```

```
        # Si el valor no está en el diccionario invertido como clave lo añado con su ex clave como valor
```

```
        else: diccionario_inv[value] = key
```

```
    # La función devuelve el diccionario invertido
```

```
    return diccionario_inv
```

```
# Función principal
```

```
def main_invertir_diccionario_con_dobles():
```

```
    # Mensaje de inicio del programa
```

```
    print("Programa para invertir un diccionario dado con listas para valores dobles\n")
```

```
    # Diccionarios de ejemplo
```

```
    diccionario1 = {'a': 1, 'b': 2, 'c': 3} # Diccionario sin valores repetidos
```

```
    diccionario2 = {'x': 'apple', 'y': 'banana', 'z': 'banana'} # Diccionario con valores repetidos
```

```
    # Por cada diccionario de ejemplo llamo a la función de inversión
```

```
    for diccionario in [diccionario1, diccionario2]:
```

```
        # Creo la variable diccionario_inv con el valor devuelto por la función
```

```
        diccionario_inv = diccionario_inverso(diccionario)
```

```
        # Imprimo el diccionario original
```

```
        print(f"Diccionario original:\n{diccionario}\n")
```

```
        # Verifico si alguno de los valores en el diccionario invertido es una lista
```

```
        if any(isinstance(value, list) for value in diccionario_inv.values()):
```

```
# Si alguno de los valores en el diccionario invertido es una lista, imprimo el mensaje de error y imprimo el
diccionario invertido
    print(f"Error: multiple keys for one value\nDiccionario invertido:\n{diccionario_inv}\n\n")

# Si ninguno de los valores del diccionario es una lista, imprimo el mensaje y el diccionario
else:
    print(f"El diccionario ha sido invertido correctamente\nDiccionario invertido:\n{diccionario_inv}\n\n")

main_invertir_diccionario_con_dobles()
```

→ exercici 2

- La gerència està interessada a analitzar més a fons les vendes en relació amb el mes. Per tant, et demanen que facis els ajustos necessaris per a mostrar la informació d'aquesta manera.

```
"""Separa numeros"""
```

```
# Función principal para separar números de otros valores
```

```
def separa_numeros(lista):
```

```
    # Creo las dos listas para almacenar los valores convertibles en números y no convertibles
```

```
    lista_de_convertibles = []
```

```
    lista_de_no_convertibles = []
```

```
    # Función para procesar cada elemento de la lista
```

```
    def procesa_elemento(elemento):
```

```
        # Verifico si el elemento de la lista es una lista o una tupla anidada
```

```
        if isinstance(elemento, (tuple, list)):
```

```
            # Si el elemento dentro de la lista es una tupla u otra lista recorro esta por cada elemento y llamo a la función para procesarlo
```

```
            for sub_item in elemento: procesa_elemento(sub_item)
```

```
        # Si el elemento de la lista es un valor simple intento convertirlo a número
```

```
        else:
```

```
            # Si el elemento de la lista se puede convertir a número lo añado a la lista de convertibles
```

```
            try: lista_de_convertibles.append(float(elemento))
```

```
            # Si el elemento de la lista no se puede convertir a número lo añado a la lista de no convertibles
```

```
            except (ValueError, TypeError): lista_de_no_convertibles.append(elemento)
```

```
    # Recorro cada elemento de la lista principal de la función separa_numeros y llamo la función procesa_elemento
```

```
    for item in lista: procesa_elemento(item)
```

```
    # La función devuelve las dos listas
```

```
    return lista_de_convertibles, lista_de_no_convertibles
```

```
# Función principal
```

```
def main_separa_numeros():
```

```
    # Mensaje de inicio del programa
```

```
    print("Programa para separar los números de los otros valores desde una lista dada\n")
```

```
    # Lista de prueba
```

```
    lista_prueba = ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2', 1, 1.4, 'not-a-number'), [1, 2, '3', '3.4']]
```

```
    # Creo dos variables con los resultados devueltos por la función pasándole la lista de prueba
```

```
    lista_convertibles, lista_no_convertibles = separa_numeros(lista_prueba)
```

```
    # Imprimo las dos listas
```

```
    print(f"Elementos convertibles: \n{lista_convertibles}\n")
```

```
    print(f"Elementos no convertibles: \n{lista_no_convertibles}\n")
```

```
main_separa_numeros()
```

★ NIVELL 3

→ exercici 1

- Comptador i endreçador de paraules d'un text:

El client va quedar content amb el comptador de paraules, però ara vol llegir arxius TXT i que calculi la freqüència de cada paraula ordenades dins de les entrades habituals del diccionari segons la lletra amb la qual comencen, és a dir, les claus han d'anar de la A a la Z i dins de la A hem d'anar de la A la Z. Per exemple, per a l'arxiu "tu_me_quieres_blanca.txt" la sortida esperada seria:

```
"""Cuenta palabras de un file"""
```

```
# Función principal
```

```
def cuenta_palabras_file():
```

```
    # Importo la biblioteca `string` para quitar la puntuación
    import string
```

```
    # Paso al programa el archivo de prueba
    file = 'tu_me_quieres_blanca.txt'
```

```
    # Creo un diccionario para contar el numero que aparece cada palabra
    contador = {}
```

```
    try:
```

```
        # Intenta abrir el archivo en modo lectura con codificación UTF-8
        with open(file, 'r', encoding='utf-8') as file:
```

```
            # Leo el contenido del archivo, convierte a minúsculas, elimina puntuación y creo una lista de valores
            # divididos por el espacio
```

```
            palabras = file.read().lower().translate(str.maketrans(string.punctuation + "¿", '' *
            (len(string.punctuation) + 2))).split()
```

```
            # Recorro cada palabra en la lista, con '.get' busco cada palabra y si no se encuentra ya en el diccionario le
            # doy valor 0, y en todos caso sumo 1 a su valor
```

```
            for word in palabras: contador[word] = contador.get(word, 0) + 1
```

```
            # Creo otra variable para el diccionario ordenado
            diccionario_ordenado = {}
```

```
            # Recorro cada palabra del diccionario contador, con.keys tomo en cuenta solo las claves
            for word in sorted(contador.keys()):
```

```
                # Creo una variable con la primera letra de cada palabra recorrida por el diccionario
                primeraletra = word[0]
```

```
                # Si la primera letra de la palabra no está ya en el diccionario, creo la nueva clave con un diccionario como
                # valor
```

```
                if primeraletra not in diccionario_ordenado: diccionario_ordenado[primeraletra] = {}
```

```
                # Añado la palabra y su frecuencia al valor/diccionario de la primera letra
                diccionario_ordenado[primeraletra][word] = contador[word]
```

```
            # Recorro y imprimo las primeras letras del diccionario ordenado
```

```
            for letra, palabras in diccionario_ordenado.items():
                print(f"\nPalabras con '{letra}':")
```

```
# Recorro y imprimo cada palabra y su frecuencia del diccionario anidado
for palabra, count in palabras.items(): print(f"{palabra}: {count}")

# Si el archivo no se encuentra muestro un mensaje de error
except FileNotFoundError: print(f"Error: el archivo '{file}' no fue encontrado.")

cuenta_palabras_file()
```