

## **Diccionarios en Python y Java**

Diego Casallas Díaz

Santiago Cespedes

Escuela de ciencias exactas e ingeniería, Universidad Sergio Arboleda

Estructura de datos no lineales

Ing. José Marcial Téllez

25 de febrero de 2025

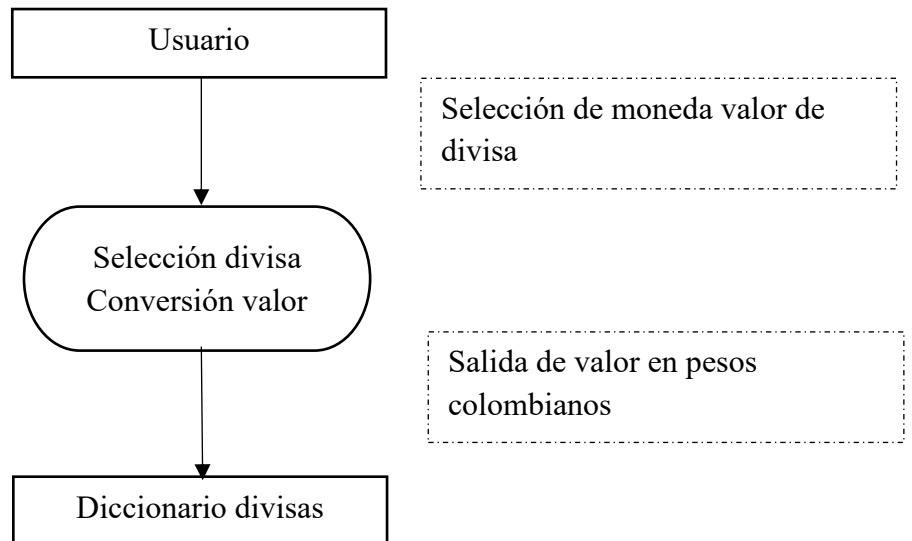
## Ejercicio 1

Realizar un programa que guarde en un diccionario el nombre de las 10 monedas más importantes y su equivalente en pesos colombianos, ej. {'Euro':4.444, 'Dollar':4.422, 'Yen':31.86, ...}, pregunte al usuario por una divisa y muestre su valor en pesos colombianos y realice la conversión de un valor ingresado por el usuario, emita un mensaje de aviso si la divisa no está en el diccionario.

Diagrama de caja negra

Entradas	Procesos	Salida
nombre divisa	Búsqueda de la divisa	valor de conversión en COP
Valor divisa a convertir	conversión de un valor	

Diagrama DFD





```

J ConversorDivisas.java > ...
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class ConversorDivisas {
    Run | Debug
    public static void main(String[] args) {
        // Diccionario de divisas y sus valores en pesos colombianos
        Map<String, Double> divisas = new HashMap<>();
        divisas.put(key:"Euro", value:4444.0);
        divisas.put(key:"Dollar", value:4422.0);
        divisas.put(key:"Yen", value:31.86);
        divisas.put(key:"Libra", value:5157.0);
        divisas.put(key:"Franco Suizo", value:4811.0);
        divisas.put(key:"Dólar Canadiense", value:3271.0);
        divisas.put(key:"Peso Mexicano", value:220.0);
        divisas.put(key:"Real Brasileño", value:870.0);
        divisas.put(key:"Rublo Ruso", value:55.45);
        divisas.put(key:"Yuan Chino", value:677.5);

        // Preguntar al usuario por una divisa
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Ingrese el nombre de la divisa: ");
        String divisa = scanner.nextLine().trim().toLowerCase();

        // Convertir la primera letra a mayúscula
        divisa = divisa.substring(beginIndex:0, endIndex:1).toUpperCase() + divisa.substring(beginIndex:1);

        if (divisas.containsKey(divisa)) {
            System.out.print("Ingrese el valor en " + divisa + ": ");
            double valor = scanner.nextDouble();
            double conversion = valor * divisas.get(divisa);
            System.out.println(valor + " " + divisa + "(s) equivale a " + conversion + " pesos colombianos.");
        } else {
            System.out.println(x:"Divisa no encontrada en el diccionario.");
        }

        scanner.close();
    }
}

```

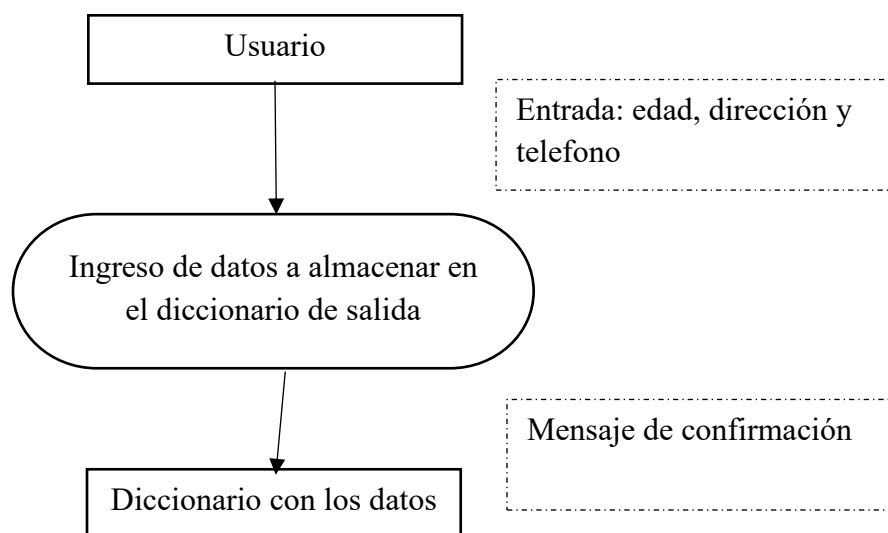
PS C:\Users\User\Desktop\ConversorDivisas> & 'C:\Program Files\Eclipse Adoptium\j
 'ConversorDivisas'
 Ingrese el nombre de la divisa: Euro
 Ingrese el valor en Euro: 5000
 5000.0 Euro(s) equivale a 2.222E7 pesos colombianos.

## Ejercicio 2

Realizar 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 tiene años, vive en y su número de teléfono es .

Entradas	Procesos	Salida
nombre	almacenamiento de la información	Monólogo de los datos en el formato especificado
Edad		

Dirección	Monólogo de la salida con la información	
Teléfono		



Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_2.py > ...

```

1 # Solicitar los datos al usuario
2 nombre = input("Ingrese su nombre: ")
3 edad = input("Ingrese su edad: ")
4 direccion = input("Ingrese su dirección: ")
5 telefono = input("Ingrese su teléfono: ")
6
7 # Almacenar los datos en un diccionario
8 datos_usuario = {
9     'Nombre': nombre,
10    'Edad': edad,
11    'Dirección': direccion,
12    'Teléfono': telefono
13 }
14
15 # Mostrar el mensaje con la información
16 print(f"{datos_usuario['Nombre']} tiene {datos_usuario['Edad']} años, vive en {datos_usuario['Dirección']} y su número de teléfono es {datos_usuario['Teléfono']}.")
17

```

```

Ingrese su nombre: Santiago
Ingrese su edad: 19
Ingrese su dirección: calle 81 bis
Ingrese su teléfono: 3138899955
Santiago tiene 19 años, vive en calle 81 bis y su número de teléfono es 3138899955.

```

```

src > J DatosUsuario.java > ...
1  import java.util.HashMap;
2  import java.util.Map;
3  import java.util.Scanner;
4
5  public class DatosUsuario {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9
10         // Solicitar los datos al usuario
11         System.out.print(s:"Ingrese su nombre: ");
12         String nombre = scanner.nextLine();
13
14         System.out.print(s:"Ingrese su edad: ");
15         String edad = scanner.nextLine();
16
17         System.out.print(s:"Ingrese su dirección: ");
18         String direccion = scanner.nextLine();
19
20         System.out.print(s:"Ingrese su teléfono: ");
21         String telefono = scanner.nextLine();
22
23         // Almacenar los datos en un mapa
24         Map<String, String> datosUsuario = new HashMap<>();
25         datosUsuario.put(key:"Nombre", nombre);
26         datosUsuario.put(key:"Edad", edad);
27         datosUsuario.put(key:"Dirección", direccion);
28         datosUsuario.put(key:"Teléfono", telefono);
29
30         // Mostrar el mensaje con la información
31         System.out.println(datosUsuario.get(key:"Nombre") + " tiene " + datosUsuario.get(key:"Edad") +
32                             " años, vive en " + datosUsuario.get(key:"Dirección") +
33                             " y su número de teléfono es " + datosUsuario.get(key:"Teléfono") + ".");
34         scanner.close();
35     }
36 }

```

```

PS C:\Users\User\Desktop\ConversorDivisas\DatosUsuario> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.8.
rio\bin' 'DatosUsuario'
Ingrese su nombre: Camilo Cometa
Ingrese su edad: 65
Ingrese su dirección: calle 10 # 7-42
Ingrese su teléfono: 3225335695
Camilo Cometa tiene 65 años, vive en calle 10 # 7-42 y su número de teléfono es 3225335695.

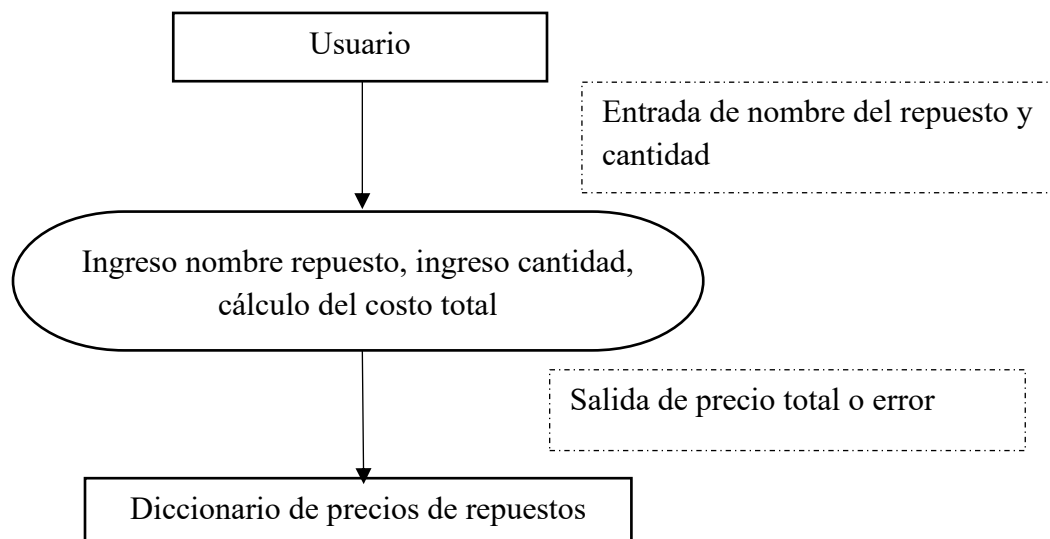
```

### Ejercicio 3

Realizar un programa que guarde en un diccionario los precios de repuestos de carro de la siguiente tabla, pregunte al usuario por un repuesto, la cantidad y muestre por pantalla el precio de ese número de repuestos de carro. Si el repuesto no está en el diccionario debe mostrar un mensaje informando de ello.

Entradas	Procesos	Salida
Nombre repuesto	Buscar en el dic el precio del repuesto	Mostrar total precio de los repuestos ó mostrar error
Cantidad repuesto	Cálculo del precio Total	

## Diagrama DFD



Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_3.py > ...

```

1  # Diccionario de precios de repuestos
2  precios_repuestos = {
3      'Bujia': 1.40,
4      'Pastillas de freno': 5.80,
5      'Radiador': 300.50,
6      'Rodamiento': 10.70
7  }
8
9  # Solicitar al usuario el nombre del repuesto
10 repuesto = input("Ingrese el nombre del repuesto: ").capitalize()
11
12 # Verificar si el repuesto está en el diccionario
13 if repuesto in precios_repuestos:
14     cantidad = int(input("Ingrese la cantidad requerida: "))
15     precio_total = precios_repuestos[repuesto] * cantidad
16     print(f"El precio total de {cantidad} {repuesto}(s) es {precio_total:.2f} pesos.")
17 else:
18     print("El repuesto no se encuentra en el inventario.")
19

```

```

Ingrese el nombre del repuesto: Bujia
Ingrese la cantidad requerida: 90
El precio total de 90 Bujia(s) es 126.00 pesos.

```

```

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class InventarioRepuestos {
    Run | Debug
    public static void main(String[] args) {
        // Diccionario de precios de repuestos
        Map<String, Double> preciosRepuestos = new HashMap<>();
        preciosRepuestos.put(key:"Bujia", value:1.40);
        preciosRepuestos.put(key:"Pastillas de freno", value:5.80);
        preciosRepuestos.put(key:"Radiador", value:300.50);
        preciosRepuestos.put(key:"Rodamiento", value:10.70);

        Scanner scanner = new Scanner(System.in);

        // Solicitar al usuario el nombre del repuesto
        System.out.print(s:"Ingrese el nombre del repuesto: ");
        String repuesto = scanner.nextLine().trim().toLowerCase();

        // Convertir la primera letra a mayúscula
        repuesto = repuesto.substring(beginIndex:0, endIndex:1).toUpperCase() + repuesto.substring(beginIndex:1);

        // Verificar si el repuesto está en el diccionario
        if (preciosRepuestos.containsKey(repuesto)) {
            System.out.print(s:"Ingrese la cantidad requerida: ");
            int cantidad = scanner.nextInt();
            double precioTotal = preciosRepuestos.get(repuesto) * cantidad;
            System.out.printf(format:"El precio total de %d %s(s) es %.2f pesos.%n", cantidad, repuesto, precioTotal);
        } else {
            System.out.println(x:"El repuesto no se encuentra en el inventario.");
        }

        scanner.close();
    }
}

```

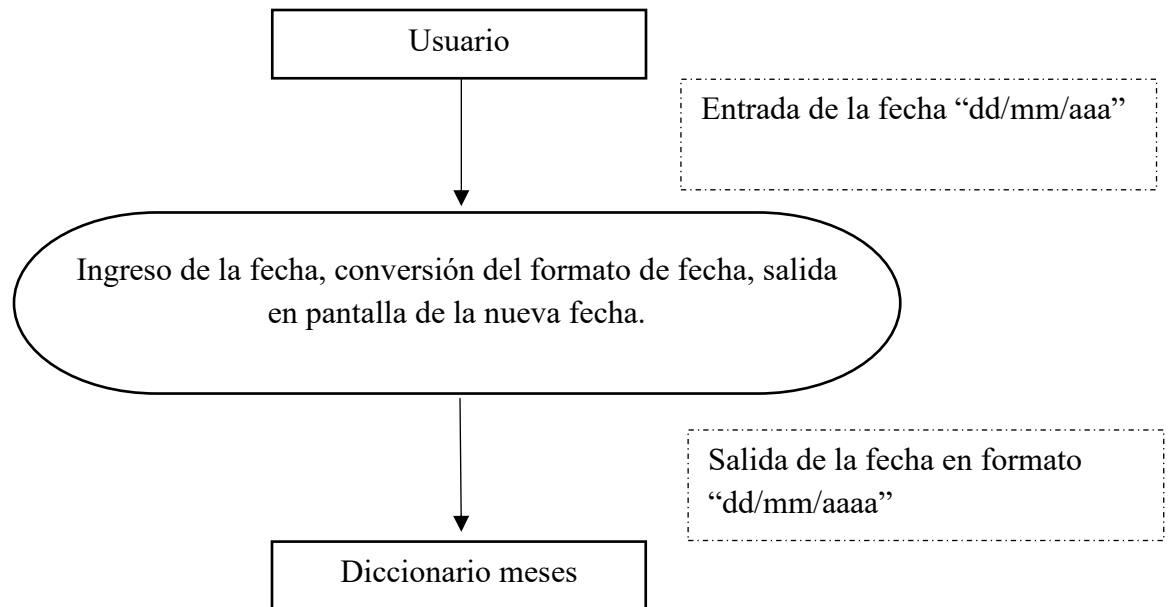
#### Ejercicio 4

Realizar un programa que pregunte una fecha en formato dd/mm/aaaa y muestre por pantalla la misma fecha en formato dd de de aaaa donde es el nombre del mes.

Entradas	Procesos	Salida
Fecha en formato : "dd/mm/aaaa"	Obtener el día, mes y año del input del usuario	Mostrar fecha en formato texto
	Conversión de fecha a formato texto	



## Diagrama DFD



```

Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio_4.py > ...
1  # Diccionario de meses
2  meses = {
3      '01': 'enero',
4      '02': 'febrero',
5      '03': 'marzo',
6      '04': 'abril',
7      '05': 'mayo',
8      '06': 'junio',
9      '07': 'julio',
10     '08': 'agosto',
11     '09': 'septiembre',
12     '10': 'octubre',
13     '11': 'noviembre',
14     '12': 'diciembre'
15 }
16
17 # Solicitar la fecha al usuario
18 fecha = input("Ingrese la fecha en formato dd/mm/aaaa: ")
19
20 # Separar la fecha en día, mes y año
21 dia, mes, año = fecha.split('/')
22
23 # Verificar si el mes es válido
24 if mes in meses:
25     # Mostrar la fecha en el nuevo formato
26     print(f"{dia} de {meses[mes]} de {año}")
27 else:
28     print("Mes no válido. Asegúrese de ingresar una fecha en el formato correcto.")
29
Ingrese la fecha en formato dd/mm/aaaa: 15/01/2005
15 de enero de 2005
PS C:\Users\SANT\OneDrive\Escritorio\Cuarto Semestre>

```

```
Ejercicio4.java > Ejercicio4
import java.util.HashMap;
import java.util.Scanner;

public class Ejercicio4 {
    Run | Debug
    public static void main(String[] args) {
        // Diccionario de meses
        HashMap<String, String> meses = new HashMap<>();
        meses.put(key:"01", value:"enero");
        meses.put(key:"02", value:"febrero");
        meses.put(key:"03", value:"marzo");
        meses.put(key:"04", value:"abril");
        meses.put(key:"05", value:"mayo");
        meses.put(key:"06", value:"junio");
        meses.put(key:"07", value:"julio");
        meses.put(key:"08", value:"agosto");
        meses.put(key:"09", value:"septiembre");
        meses.put(key:"10", value:"octubre");
        meses.put(key:"11", value:"noviembre");
        meses.put(key:"12", value:"diciembre");

        // Solicitar la fecha al usuario
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Ingrese la fecha en formato dd/mm/aaaa: ");
        String fecha = scanner.nextLine();

        // Separar la fecha en día, mes y año
        String[] partes = fecha.split(regex:"/");
        if (partes.length != 3) {
            System.out.println(x:"Formato de fecha incorrecto. Asegúrese de ingresar una fecha en el formato dd/mm/aaaa.");
            return;
        }

        String día = partes[0];
        String mes = partes[1];
        String año = partes[2];

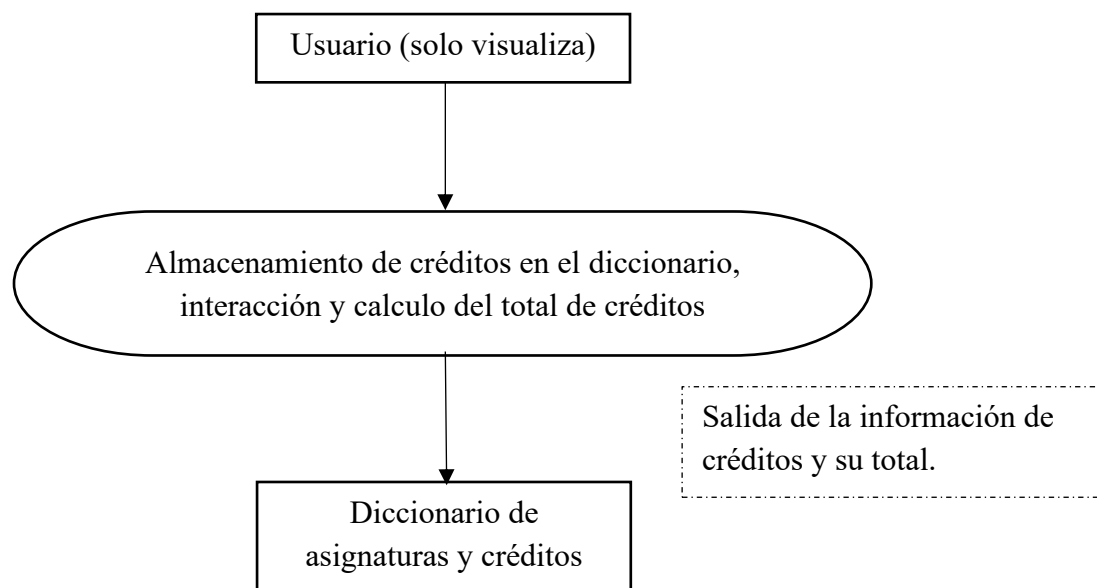
        // Verificar si el mes es válido
        if (meses.containsKey(mes)) {
            // Mostrar la fecha en el nuevo formato
            System.out.println(día + " de " + meses.get(mes) + " de " + año);
        } else {
            System.out.println(x:"Mes no válido. Asegúrese de ingresar una fecha en el formato correcto.");
        }
    }
}
```

## Ejercicio 5

Realizar un programa que almacene el diccionario con los créditos de las asignaturas de un curso {'Matemáticas': 6, 'Física': 4, 'Química': 5} y después muestre por pantalla los créditos de cada asignatura en el formato tiene créditos, donde es cada una de las asignaturas del curso, y son sus créditos. Al final debe mostrar también el número total de créditos del curso.

Entradas	Procesos	Salida
No se requiere entrada del usuario	Créditos de las asignaturas dentro de un diccionario.	Mostrar el total de créditos
	Calcular el total de créditos, según valores en diccionario	

## Diagrama DFD



Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_5.py > ...

```
1  # Diccionario con los créditos de las asignaturas
2  creditos_curso = {
3      'Matemáticas': 6,
4      'Física': 4,
5      'Química': 5
6  }
7
8  # Mostrar los créditos de cada asignatura
9  for asignatura, creditos in creditos_curso.items():
10     print(f"{asignatura} tiene {creditos} créditos.")
11
12 # Calcular y mostrar el total de créditos
13 total_creditos = sum(creditos_curso.values())
14 print(f"El total de créditos del curso es {total_creditos}.")
15
```

Matemáticas tiene 6 créditos.

Física tiene 4 créditos.

Química tiene 5 créditos.

El total de créditos del curso es 15.

PS C:\Users\SANT\OneDrive\Escritorio\Cuarto Semestre> □

```

import java.util.HashMap;
import java.util.Map;

public class Ejercicio5 {
    Run | Debug
    public static void main(String[] args) {
        // Diccionario con los créditos de las asignaturas
        HashMap<String, Integer> creditosCurso = new HashMap<>();
        creditosCurso.put(key:"Matemáticas", value:6);
        creditosCurso.put(key:"Física", value:4);
        creditosCurso.put(key:"Química", value:5);

        // Mostrar los créditos de cada asignatura
        for (Map.Entry<String, Integer> entry : creditosCurso.entrySet()) {
            String asignatura = entry.getKey();
            int creditos = entry.getValue();
            System.out.println(asignatura + " tiene " + creditos + " créditos.");
        }

        // Calcular y mostrar el total de créditos
        int totalCreditos = 0;
        for (int creditos : creditosCurso.values()) {
            totalCreditos += creditos;
        }
        System.out.println("El total de créditos del curso es " + totalCreditos + ".");
    }
}

```

```

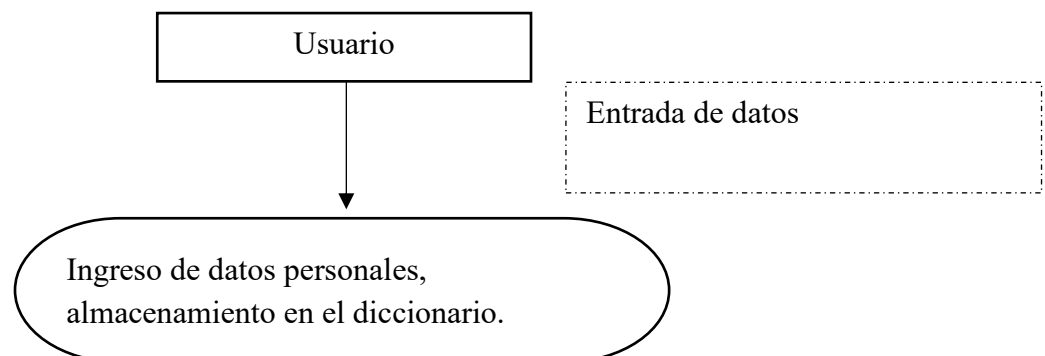
Matemáticas tiene 6 créditos.
Química tiene 5 créditos.
Física tiene 4 créditos.
El total de créditos del curso es 15.
PS C:\Users\User\Desktop\Ejercicio5>

```

Realizar un programa que cree un diccionario vacío y lo vaya llenado con información sobre una persona (por ejemplo, nombre, edad, sexo, teléfono, correo electrónico, etc.) que se le pida al usuario. Cada vez que se añada un nuevo dato debe imprimirse el contenido del diccionario.

Entradas	Procesos	Salida
Valor de la clave	Instanciación de un diccionario	Mostrar en consola el nuevo valor de cada clave del diccionario.
	Agregar valor a cada una de las claves dentro del diccionario	

Diagrama DFD



```

Ejercicio6.java 1 x
src > J Ejercicio6.java > ...
1 import java.util.HashMap;
2 import java.util.Map;
3 import java.util.Scanner;
4
5 public class Ejercicio6 {
6     public static void main(String[] args) {
7         // Crear un diccionario vacío
8         HashMap<String, String> datosPersona = new HashMap<>();
9
10        // Lista de campos a solicitar
11        String[] campos = {"Nombre", "Edad", "Sexo", "Teléfono", "Correo electrónico"};
12
13        // Crear un objeto Scanner para la entrada del usuario
14        Scanner scanner = new Scanner(System.in);
15
16        // Solicitar los datos y llenar el diccionario
17        for (String campo : campos) {
18            System.out.print("Ingrese su " + campo + ": ");
19            String valor = scanner.nextLine();
20            datosPersona.put(campo, valor);
21
22            // Mostrar los datos actuales
23            System.out.println("Datos actuales: " + datosPersona);
24        }
25
26        // Cerrar el scanner
27        scanner.close();
28    }
29 }
30
Ingrese su Nombre: camilo cometa
Datos actuales: {Nombre=camilo cometa}
Ingrese su Edad: 65
Datos actuales: {Nombre=camilo cometa, Edad=65}
Ingrese su Sexo: masculino
Datos actuales: {Nombre=camilo cometa, Edad=65, Sexo=masculino}
Ingrese su Teléfono: 3225335696
Datos actuales: {Nombre=camilo cometa, Edad=65, Sexo=masculino, Teléfono=3225335696}
Ingrese su Correo electrónico: vvvv@gmail.com
Datos actuales: {Nombre=camilo cometa, Edad=65, Correo electrónico=vvvv@gmail.com, Sexo=masculino, Teléfono=3225335696}
PS C:\Users\User\Desktop\Ejercicio6>

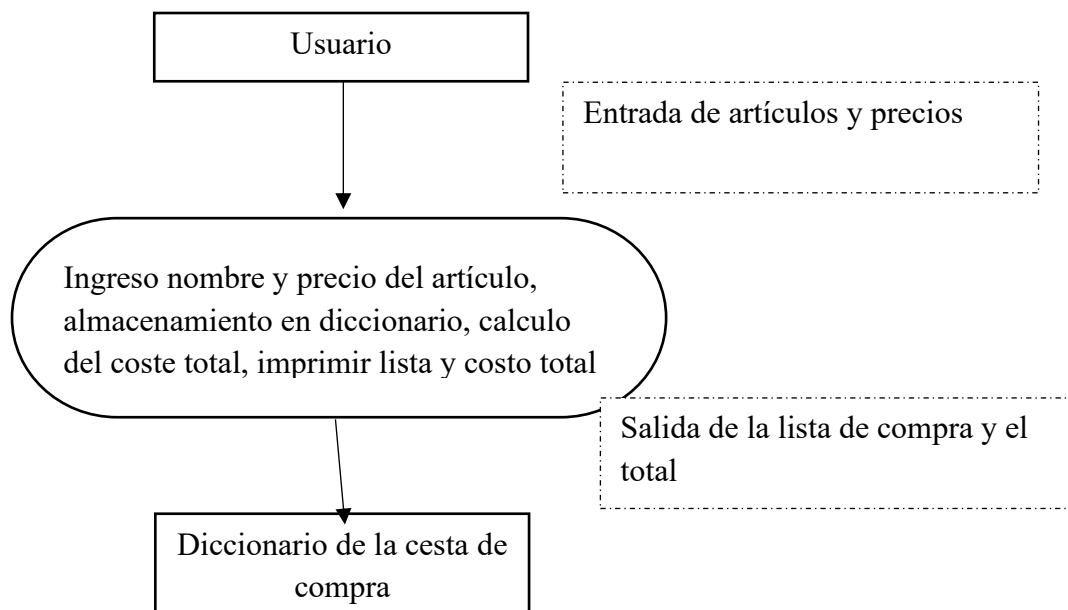
```

## Ejercicio 7

Realizar un programa que cree un diccionario simulando una cesta de la compra. El programa debe preguntar el artículo y su precio y añadir el par al diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato

Entradas	Procesos	Salida
Valor de la clave	Crear diccionario	
	Agregar valor a cada una de las claves dentro del diccionario	Mostrar en consola el nuevo valor de cada clave del diccionario.

Diagrama DFD



```

Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio_7.py > ...
1  # Crear un diccionario vacío para la cesta de la compra
2  cesta_compra = {}
3
4  # Ciclo para añadir artículos a la cesta
5  while True:
6      articulo = input("Ingrese el nombre del artículo (o 'terminar' para finalizar): ")
7      if articulo.lower() == 'terminar':
8          break
9      precio = float(input(f"Ingrese el precio de {articulo}: "))
10     cesta_compra[articulo] = precio
11
12     # Mostrar la lista de la compra y calcular el coste total
13     print("\nLista de la compra:")
14     coste_total = 0
15     for articulo, precio in cesta_compra.items():
16         print(f"{articulo}: {precio:.2f}")
17         coste_total += precio
18
19     print(f"\nTotal Coste: {coste_total:.2f}")
20
  
```



```

rc > J CestaCompra.java > ...
1  import java.util.HashMap;
2  import java.util.Map;
3  import java.util.Scanner;
4
5  public class CestaCompra {
6
7      Run | Debug
8      public static void main(String[] args) {
9          Scanner scanner = new Scanner(System.in);
10         Map<String, Double> cestaCompra = new HashMap<>();
11
12         // Ciclo para añadir artículos a la cesta
13         while (true) {
14             System.out.print("Ingrese el nombre del artículo (o 'terminar' para finalizar): ");
15             String articulo = scanner.nextLine().toLowerCase();
16             if (articulo.equals(anObject:"terminar")) {
17                 break;
18             }
19             System.out.print("Ingrese el precio de " + articulo + ": ");
20             double precio = Double.parseDouble(scanner.nextLine());
21             cestaCompra.put(articulo, precio);
22         }
23
24         // Mostrar la lista de la compra y calcular el coste total
25         System.out.println("Lista de la compra:");
26         double costeTotal = 0;
27         for (Map.Entry<String, Double> entry : cestaCompra.entrySet()) {
28             String articulo = entry.getKey();
29             double precio = entry.getValue();
30             System.out.printf(format:"%s: %.2f%n", articulo, precio);
31             costeTotal += precio;
32         }
33
34         System.out.printf(format:"\nTotal Coste: %.2f%n", costeTotal);
35     }
36

```

```

Ingrese el precio de 2 libras de arroz: 2000
Ingrese el nombre del artículo (o 'terminar' para finalizar): pan bimbo
Ingrese el precio de pan bimbo: 5000
Ingrese el nombre del artículo (o 'terminar' para finalizar): galletas
Ingrese el precio de galletas: 8000
Ingrese el nombre del artículo (o 'terminar' para finalizar): pasta
Ingrese el precio de pasta: 4000
Ingrese el nombre del artículo (o 'terminar' para finalizar): cereal
Ingrese el precio de cereal : 8000
Ingrese el nombre del artículo (o 'terminar' para finalizar): atun
Ingrese el precio de atun: 5000
Ingrese el nombre del artículo (o 'terminar' para finalizar): terminar

```

```

Ingrese el precio de atun: 5000
Ingrese el nombre del artículo (o 'terminar' para finalizar): terminar

Lista de la compra:
pan bimbo: 5000,00
atun: 5000,00
pasta: 4000,00
cereal : 8000,00
galletas: 8000,00
2 libras de arroz: 2000,00

Total Coste: 32000,00

```

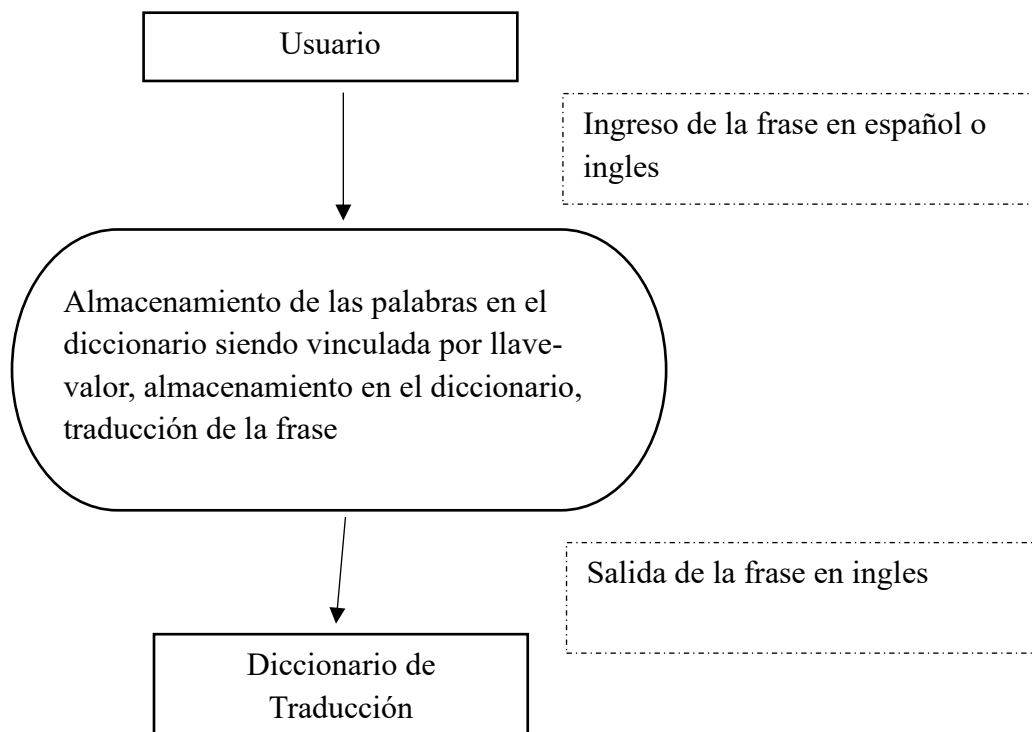


## Ejercicio 8

Realizar un programa que cree un diccionario de traducción español-inglés. El usuario introducirá las palabras en español e inglés separadas por dos puntos, y cada par : separados por comas. El programa debe crear un diccionario con las palabras y sus traducciones. Después pedirá una frase en español y utilizará el diccionario para traducirla palabra a palabra. Si una palabra no está en el diccionario debe dejarla sin traducir.

Entradas	Procesos	Salida
Lista de palabra en español e ingles	Crear diccionario	
Frase en español	Traducción de la frase utilizando el diccionario	Mostrar en consola frase traducida en inglés palabra por palabra.

## Diagrama DFD



Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_8.py > ...

```

1  # Solicitar la lista de palabras y sus traducciones
2  entrada = input("Ingrese las palabras en formato 'español:inglés' separadas por comas: ")
3
4  # Crear el diccionario de traducción
5  diccionario_traducccion = {}
6  pares = entrada.split(',')
7
8  for par in pares:
9      espanol, ingles = par.split(':')
10     diccionario_traducccion[espanol.strip()] = ingles.strip()
11
12 # Solicitar la frase en español para traducir
13 frase = input("Ingrese una frase en español: ")
14
15 # Traducir la frase palabra por palabra
16 palabras = frase.split()
17 frase_traducida = []
18
19 for palabra in palabras:
20     if palabra in diccionario_traducccion:
21         frase_traducida.append(diccionario_traducccion[palabra])
22     else:
23         frase_traducida.append(palabra) # Dejar sin traducir si no está en el diccionario
24
25 # Unir la frase traducida
26 print("Frase traducida:", ' '.join(frase_traducida))
27

```

Ingrese las palabras en formato 'español:inglés' separadas por comas: gato:cat,perro:dog,casa:house,libro:book

Ingrese una frase en español: El gato está en la casa con un libro

Frase traducida: El cat está en la house con un book

PS C:\Users\SANT\OneDrive\Escritorio\Cuarto Semestre>

```

3  import java.util.Scanner;
4
5  public class Traductor {
6
7      Run | Debug
8      public static void main(String[] args) {
9          Scanner scanner = new Scanner(System.in);
10
11          // Solicitar la lista de palabras y sus traducciones
12          System.out.print(s:"Ingrese las palabras en formato 'español:inglés' separadas por comas: ");
13          String entrada = scanner.nextLine();
14
15          // Crear el diccionario de traducción
16          Map<String, String> diccionarioTraducccion = new HashMap<>();
17          String[] pares = entrada.split(regex:",");
18
19          for (String par : pares) {
20              String[] idiomas = par.split(regex:"");
21              String espanol = idiomas[0].trim();
22              String ingles = idiomas[1].trim();
23              diccionarioTraducccion.put(espanol, ingles);
24          }
25
26          // Solicitar la frase en español para traducir
27          System.out.print(s:"Ingrese una frase en español: ");
28          String frase = scanner.nextLine();
29
30          // Traducir la frase palabra por palabra
31          String[] palabras = frase.split(regex:" ");
32          StringBuilder fraseTraducida = new StringBuilder();
33
34          for (String palabra : palabras) {
35              if (diccionarioTraducccion.containsKey(palabra)) {
36                  fraseTraducida.append(diccionarioTraducccion.get(palabra)).append(str:" ");
37              } else {
38                  fraseTraducida.append(palabra).append(str:" "); // Dejar sin traducir si no está en el diccionario
39              }
40          }
41
42          // Mostrar la frase traducida
43          System.out.println("Frase traducida: " + fraseTraducida.toString().trim());
44      }
45  }

```

PS C:\Users\User\Desktop\Traductor> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.8.7-hotspot\bin\

Ingrese las palabras en formato 'español:inglés' separadas por comas: gato:cat,rata:rat,cocinar:co

Ingrese una frase en español: el perro y la rata le van a cocinar al gato

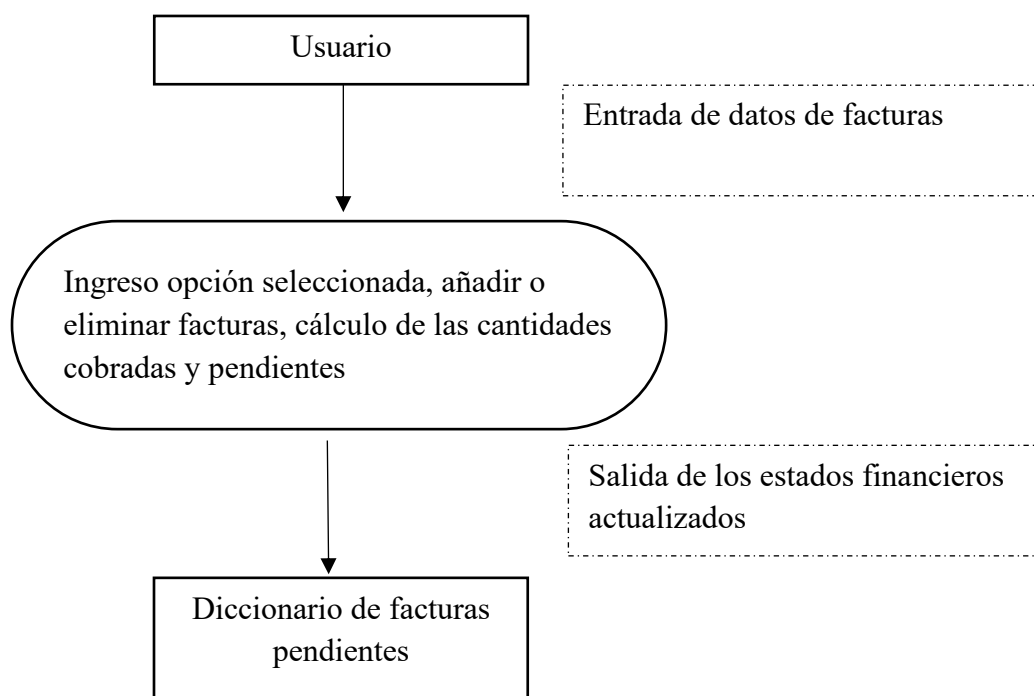
Frase traducida: el perro y la rat le van a cook al cat

## Ejercicio 9

Realizar un programa que gestione las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.

Entradas	Procesos	Salida
Opción seleccionada por el usuario		
No. de factura y coste	Añadir nueva factura al diccionario	
No. De facturas al pagar	Eliminar factura pagada, calcular, mostrar la cantidad cobrada y cantidad pendiente de cobro.	

## Diagrama DFD



Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_9.py > ...

```
1  # Crear un diccionario vacío para almacenar las facturas
2  facturas = {}
3
4  # Inicializar las cantidades cobradas y pendientes
5  cantidad_cobrada = 0
6  cantidad_pendiente = 0
7
8  while True:
9      # Mostrar el menú de opciones al usuario
10     print("\nSeleccione una opción:")
11     print("1. Añadir nueva factura")
12     print("2. Pagar una factura")
13     print("3. Terminar")
14     opcion = input("Ingrese su opción: ")
15
16     if opcion == '1':
17         # Añadir una nueva factura
18         numero_factura = input("Ingrese el número de la factura: ")
19         coste = float(input("Ingrese el coste de la factura: "))
20         facturas[numero_factura] = coste
21         cantidad_pendiente += coste
22         print(f"Factura {numero_factura} añadida con un coste de {coste:.2f}.")
23
24     elif opcion == '2':
25         # Pagar una factura existente
26         numero_factura = input("Ingrese el número de la factura a pagar: ")
27         if numero_factura in facturas:
28             coste = facturas.pop(numero_factura)
29             cantidad_cobrada += coste
30             cantidad_pendiente -= coste
31             print(f"Factura {numero_factura} pagada con un coste de {coste:.2f}.")
32         else:
33             print(f"La factura {numero_factura} no existe.")
```

Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_9.py > ...

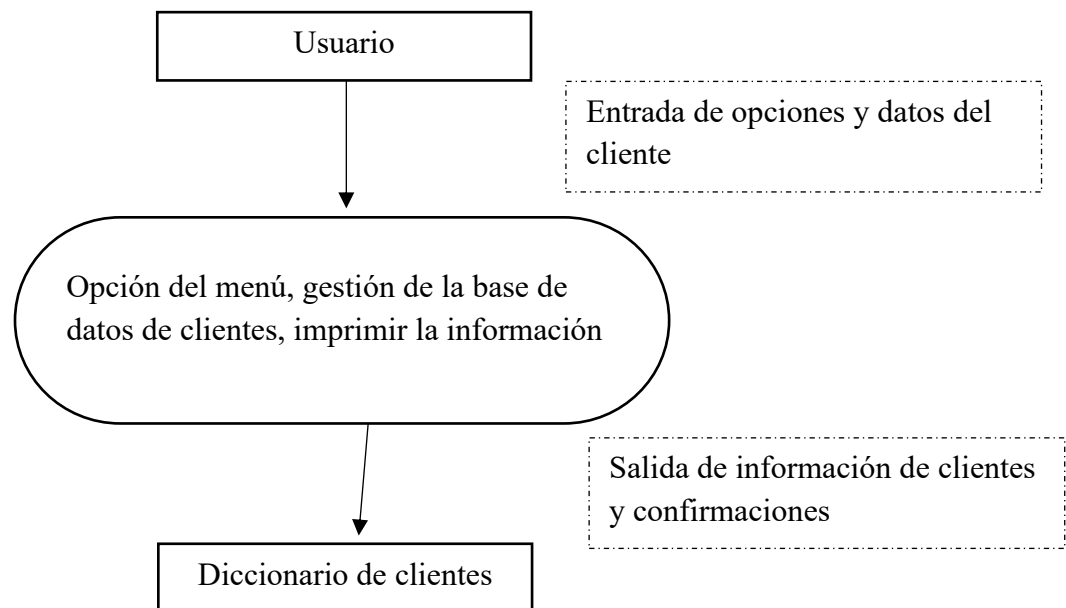
```
1  # Crear un diccionario vacío para almacenar las facturas
2  facturas = {}
3
4  # Inicializar las cantidades cobradas y pendientes
5  cantidad_cobrada = 0
6  cantidad_pendiente = 0
7
8  while True:
9      # Mostrar el menú de opciones al usuario
10     print("\nSeleccione una opción:")
11     print("1. Añadir nueva factura")
12     print("2. Pagar una factura")
13     print("3. Terminar")
14     opcion = input("Ingrese su opción: ")
15
16     if opcion == '1':
17         # Añadir una nueva factura
18         numero_factura = input("Ingrese el número de la factura: ")
19         coste = float(input("Ingrese el coste de la factura: "))
20         facturas[numero_factura] = coste
21         cantidad_pendiente += coste
22         print(f"Factura {numero_factura} añadida con un coste de {coste:.2f}.")
23
24     elif opcion == '2':
25         # Pagar una factura existente
26         numero_factura = input("Ingrese el número de la factura a pagar: ")
27         if numero_factura in facturas:
28             coste = facturas.pop(numero_factura)
29             cantidad_cobrada += coste
30             cantidad_pendiente -= coste
31             print(f"Factura {numero_factura} pagada con un coste de {coste:.2f}.")
32         else:
33             print(f"La factura {numero_factura} no existe.")
```

## Ejercicio 10

Realizar un programa que permita gestionar la base de datos de clientes de una empresa. Los clientes se guardarán en un diccionario en el que la clave de cada cliente será su NIF, y el valor será otro diccionario con los datos del cliente (nombre, dirección, teléfono, correo, preferente), donde preferente tendrá el valor True si se trata de un cliente preferente. El programa debe preguntar al usuario por una opción del siguiente menú: (1) Añadir cliente, (2) Eliminar cliente, (3) Mostrar cliente, (4) Listar todos los clientes, (5) Listar clientes preferentes, (6) Terminar. En función de la opción elegida el programa tendrá que hacer lo siguiente:

Entradas	Procesos	Salidas
Opción seleccionada	Añadir, eliminar o mostrar cliente	Información sobre cliente
Nif del cliente	Mostrar todos los clientes	Listar clientes
Datos del cliente		

## Diagrama DFD





Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio\_10.py > ...

```
1  # Crear un diccionario para almacenar la base de datos de clientes
2  clientes = {}
3
4  while True:
5      # Mostrar el menú de opciones
6      print("\nSeleccione una opción:")
7      print("1. Añadir cliente")
8      print("2. Eliminar cliente")
9      print("3. Mostrar cliente")
10     print("4. Listar todos los clientes")
11     print("5. Listar clientes preferentes")
12     print("6. Terminar")
13     opcion = input("Ingrese su opción: ")
14
15     if opcion == '1':
16         # Añadir un nuevo cliente
17         nif = input("Ingrese el NIF del cliente: ")
18         nombre = input("Ingrese el nombre del cliente: ")
19         direccion = input("Ingrese la dirección del cliente: ")
20         telefono = input("Ingrese el teléfono del cliente: ")
21         correo = input("Ingrese el correo electrónico del cliente: ")
22         preferente = input("¿Es un cliente preferente? (sí/no): ").lower() == 'sí'
23         clientes[nif] = {
24             'Nombre': nombre,
25             'Dirección': direccion,
26             'Teléfono': telefono,
27             'Correo': correo,
28             'Preferente': preferente
29         }
30         print(f"Cliente {nombre} añadido con éxito.")
31
32     elif opcion == '2':
33         # Eliminar un cliente existente
```

```

32 elif opcion == '2':
33     # Eliminar un cliente existente
34     nif = input("Ingrese el NIF del cliente a eliminar: ")
35     if nif in clientes:
36         del clientes[nif]
37         print(f"Cliente con NIF {nif} eliminado con éxito.")
38     else:
39         print("Cliente no encontrado.")
40
41 elif opcion == '3':
42     # Mostrar los datos de un cliente
43     nif = input("Ingrese el NIF del cliente: ")
44     if nif in clientes:
45         print(f"Datos del cliente {nif}: {clientes[nif]}")
46     else:
47         print("Cliente no encontrado.")
48
49 elif opcion == '4':
50     # Listar todos los clientes
51     if clientes:
52         print("Lista de todos los clientes:")
53         for nif, datos in clientes.items():
54             print(f"NIF: {nif}, Nombre: {datos['Nombre']}")
55     else:
56         print("No hay clientes en la base de datos.")
57
58 elif opcion == '5':
59     # Listar clientes preferentes
60     preferentes = {nif: datos for nif, datos in clientes.items() if datos['Preferente']}
61     if preferentes:
62         print("Lista de clientes preferentes:")
63
64 elif opcion == '5':
65     # Listar clientes preferentes
66     preferentes = {nif: datos for nif, datos in clientes.items() if datos['Preferente']}
67     if preferentes:
68         print("Lista de clientes preferentes:")
69         for nif, datos in preferentes.items():
70             print(f"NIF: {nif}, Nombre: {datos['Nombre']}")
71     else:
72         print("No hay clientes preferentes en la base de datos.")
73
74 elif opcion == '6':
75     # Terminar el programa
76     print("Terminando el programa.")
77     break
78
79 else:
80     print("Opción no válida. Intente nuevamente.")

```



```

1  import java.util.HashMap;
2  import java.util.Map;
3  import java.util.Scanner;
4
5  public class ClienteDatabase {
6      Run | Debug
7      public static void main(String[] args) {
8          Map<String, Map<String, String>> clientes = new HashMap<>();
9          Scanner scanner = new Scanner(System.in);
10
11          while (true) {
12              // Mostrar el menú de opciones
13              System.out.println(x:"\nSelecione una opción:");
14              System.out.println(x:"1. Añadir cliente");
15              System.out.println(x:"2. Eliminar cliente");
16              System.out.println(x:"3. Mostrar cliente");
17              System.out.println(x:"4. Listar todos los clientes");
18              System.out.println(x:"5. Listar clientes preferentes");
19              System.out.println(x:"6. Terminar");
20              System.out.print(s:"Ingrese su opción: ");
21              String opcion = scanner.nextLine();
22
23              switch (opcion) {
24                  case "1":
25                      // Añadir un nuevo cliente
26                      System.out.print(s:"Ingrese el NIF del cliente: ");
27                      String nif = scanner.nextLine();
28                      System.out.print(s:"Ingrese el nombre del cliente: ");
29                      String nombre = scanner.nextLine();
30                      System.out.print(s:"Ingrese la dirección del cliente: ");
31                      String direccion = scanner.nextLine();
32                      System.out.print(s:"Ingrese el teléfono del cliente: ");
33                      String telefono = scanner.nextLine();
34                      System.out.print(s:"Ingrese el correo electrónico del cliente: ");
35                      String correo = scanner.nextLine();
36
37                      System.out.print(s:"¿Es un cliente preferente? (sí/no): ");
38                      String preferente = scanner.nextLine().equalsIgnoreCase(anotherString:"sí") ? "sí" : "no";
39
40                      Map<String, String> cliente = new HashMap<>();
41                      cliente.put(key:"Nombre", nombre);
42                      cliente.put(key:"Dirección", direccion);
43                      cliente.put(key:"Teléfono", telefono);
44                      cliente.put(key:"Correo", correo);
45                      cliente.put(key:"Preferente", preferente);
46
47                      clientes.put(nif, cliente);
48                      System.out.println("Cliente " + nombre + " añadido con éxito.");
49                      break;
50                  case "2":
51                      // Eliminar un cliente existente
52                      System.out.print(s:"Ingrese el NIF del cliente a eliminar: ");
53                      nif = scanner.nextLine();
54                      if (clientes.containsKey(nif)) {
55                          clientes.remove(nif);
56                          System.out.println("Cliente con NIF " + nif + " eliminado con éxito.");
57                      }
58                  }
59              }
60          }
61      }
62  }

```

```

55         System.out.println("Cliente con NIF " + nif + " eliminado con éxito.");
56     } else {
57         System.out.println(x:"Cliente no encontrado.");
58     }
59     break;
60
61     case "3":
62         // Mostrar los datos de un cliente
63         System.out.print(s:"Ingrese el NIF del cliente: ");
64         nif = scanner.nextLine();
65         if (clientes.containsKey(nif)) {
66             System.out.println("Datos del cliente " + nif + ": " + clientes.get(nif));
67         } else {
68             System.out.println(x:"Cliente no encontrado.");
69         }
70         break;
71
72     case "4":
73         // Listar todos los clientes
74         if (!clientes.isEmpty()) {
75             System.out.println(x:"Lista de todos los clientes:");
76             for (Map.Entry<String, Map<String, String>> entry : clientes.entrySet()) {
77                 System.out.println("NIF: " + entry.getKey() + ", Nombre: " + entry.getValue().get(key:"Nombre"));
78             }
79         } else {
80             System.out.println(x:"No hay clientes en la base de datos.");
81         }
82         break;
83
84     case "5":
85         // Listar clientes preferentes
86         boolean hayPreferentes = false;
87         for (Map.Entry<String, Map<String, String>> entry : clientes.entrySet()) {
88             if (entry.getValue().get(key:"Preferente").equals(anObject:"si")) {
89                 if (!hayPreferentes) {
90                     System.out.println(x:"Lista de clientes preferentes:");
91                     hayPreferentes = true;
92                 }
93                 System.out.println("NIF: " + entry.getKey() + ", Nombre: " + entry.getValue().get(key:"Nombre"));
94             }
95         }
96         if (!hayPreferentes) {
97             System.out.println(x:"No hay clientes preferentes en la base de datos.");
98         }
99         break;
100
101     case "6":
102         // Terminar el programa
103         System.out.println(x:"Terminando el programa.");
104         scanner.close();
105         return;
106
107     default:
108         System.out.println(x:"Opción no válida. Intente nuevamente.");
109         break;
110 }
111 }
112 }
113 }

```

seleccione una opción:

1. Añadir cliente
2. Eliminar cliente
3. Mostrar cliente
4. Listar todos los clientes
5. Listar clientes preferentes
6. Terminar

ingrese su opción: 5

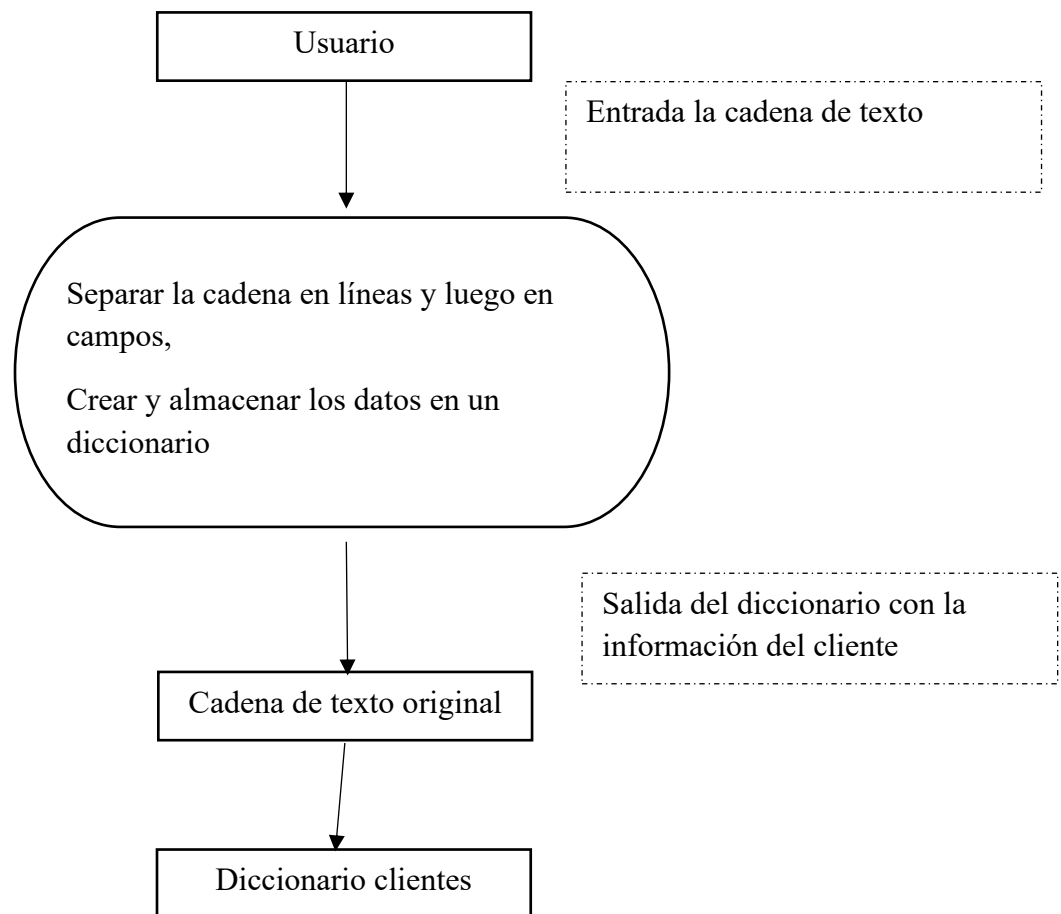
No hay clientes preferentes en la base de datos.

## Ejercicio 11

El directorio de los clientes de una empresa está organizado en una cadena de texto como la de más abajo, donde cada línea contiene la información del nombre, email, teléfono, cédula, y el descuento que se le aplica. Las líneas se separan con el carácter de cambio de línea \n y la primera línea contiene los nombres de los campos con la información contenida en el directorio.

Entradas	Procesos	Salidas
Información clientes	Separa cadena en líneas y luego en campos	Diccionario con la información de los clientes
	Crear un diccionario donde la cédula es la llave	

## Diagrama DFD



```

Estructuras De Datos No Lineales > CR-1 T-2 Ejercicios de diccionarios > Python > Ejercicio_11.py > ...
1 # Cadena de texto de ejemplo
2 directorio = "cedula;nombre;email;telefono;descuento\n1001234567;Luis González;luisgonzalez@mail.com;656343576;12.5\n1007654321K;Ana López;analopez@mail.com;612345678;10.0"
3
4 # Separar la cadena en líneas
5 lineas = directorio.split('\n')
6
7 # La primera línea contiene los nombres de los campos
8 campos = lineas[0].split(';')
9
10 # Crear un diccionario para almacenar la información de los clientes
11 clientes = {}
12
13 # Procesar cada línea (excepto la primera)
14 for linea in lineas[1:]:
15     datos = linea.split(';')
16     cedula = datos[0]
17     # Crear un diccionario para el cliente actual
18     info_cliente = {campos[i]: datos[i] for i in range(1, len(campos))}
19     # Añadir el cliente al diccionario principal
20     clientes[cedula] = info_cliente
21
22 # Mostrar el diccionario resultante
23 print("Diccionario de clientes:")
24 for cedula, info in clientes.items():
25     print(f'{cedula}: {info}')
26

```

```

import java.util.HashMap;
import java.util.Map;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        // Cadena de texto de ejemplo
        String directorio = "cedula;nombre;email;telefono;descuento\n1001234567;Luis González;luisgonzalez@mail.com;656343576;12.5\n1007654321;Ana L

        // Separar la cadena en líneas
        String[] lineas = directorio.split(regex:"\\n");

        // La primera línea contiene los nombres de los campos
        String[] campos = lineas[0].split(regex:";");

        // Crear un diccionario para almacenar la información de los clientes
        Map<String, Map<String, String>> clientes = new HashMap<>();

        // Procesar cada línea (excepto la primera)
        for (int i = 1; i < lineas.length; i++) {
            String[] datos = lineas[i].split(regex:";");
            String cedula = datos[0];

            // Crear un diccionario para el cliente actual
            Map<String, String> infoCliente = new HashMap<>();
            for (int j = 0; j < campos.length; j++) {
                infoCliente.put(campos[j], datos[j]);
            }

            // Añadir el cliente al diccionario principal
            clientes.put(cedula, infoCliente);
        }

        // Mostrar el diccionario resultante
        System.out.println(x:"Diccionario de clientes:");
        for (Map.Entry<String, Map<String, String>> cliente : clientes.entrySet()) {
            System.out.println("Cédula: " + cliente.getKey() + " Info: " + cliente.getValue());
        }
    }
}

```

```

● PS C:\Users\User\Desktop\Main> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.8.7-hotspot\bin\jav
Diccionario de clientes:
Cédula: 1001234567 Info: {cedula=1001234567, descuento=12.5, telefono=656343576, nombre=Luis Gonz
Cédula: 1007654321 Info: {cedula=1007654321, descuento=10.0, telefono=612345678, nombre=Ana López
○ PS C:\Users\User\Desktop\Main>

```