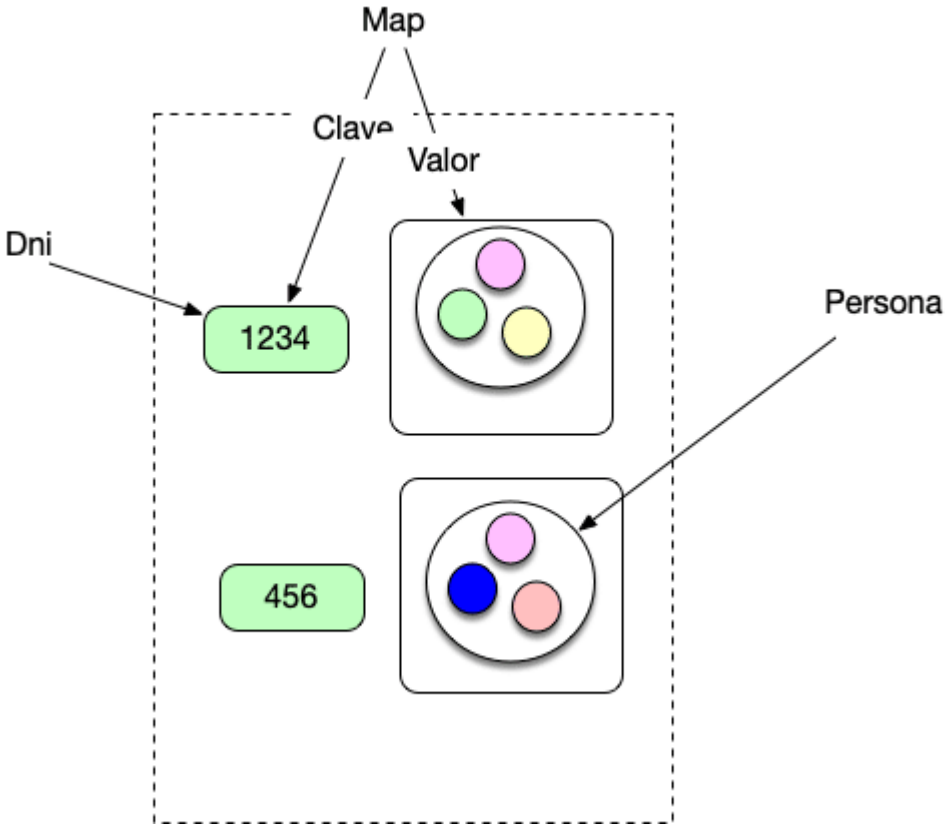
¿Cómo implementar un Java Map for loop o un bucle dentro de una estructura de diccionario o Map? . Los Java Map son estructuras que almacenan parejas de clave y valor .



Es decir por ejemplo pueden almacenar el DNI de una Persona .

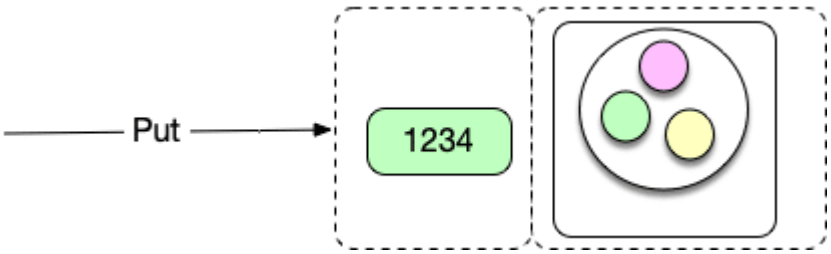
```
package com.arquitecturajava;

public class Persona {

	private String dni;
	private String nombre;
	private int edad;
	public String getDni() {
		return dni;
	}
```

```
	public void setDni(String dni) {
		this.dni = dni;
	}
	public String getNombre() {
		return nombre;
	}
	public void setNombre(String nombre) {
		this.nombre = nombre;
	}
	public int getEdad() {
		return edad;
	}
	public void setEdad(int edad) {
		this.edad = edad;
	}
	public Persona(String dni, String nombre, int edad) {
		super();
		this.dni = dni;
		this.nombre = nombre;
		this.edad = edad;
	}
}
```

Usaremos el DNI como clave y el objeto Persona como valor .



De tal forma que cada vez que queramos añadir un ejemplo al mapa realicemos una

operación de PUT e insertemos una clave y un valor.

```
package com.arquitecturajava;

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

public class Principal2 {

	public static void main(String[] args) {
		Persona p= new Persona("123","pepe",30);
		Persona p2= new Persona("456","juan",40);
		Persona p3= new Persona("231","ana",50);
		Map<String, Persona>  mapa= new
HashMap<String,Persona>();
		mapa.put(p.getDni(),p);
		mapa.put(p2.getDni(),p2);
		mapa.put(p3.getDni(),p3);

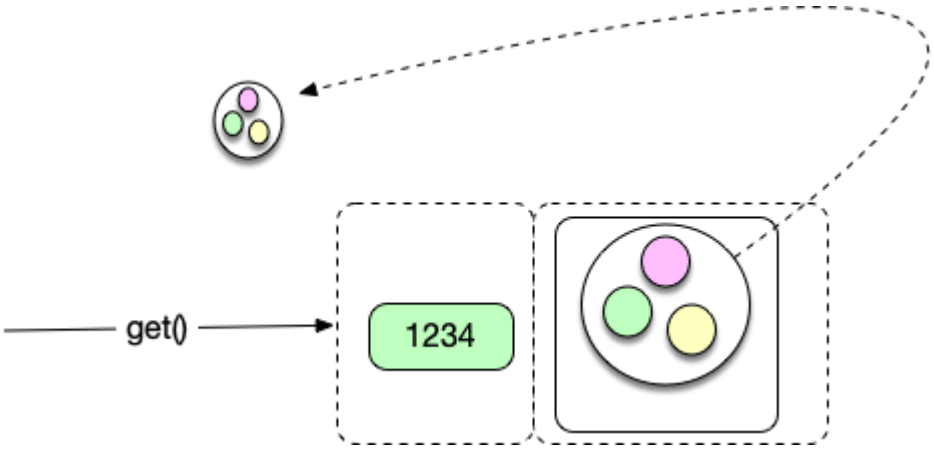
	}

}
```

Usando Java Map for loop

En un principio lo que es muy rápido es insertar claves y valores en un mapa ya que nos encontramos con su funcionamiento habitual.

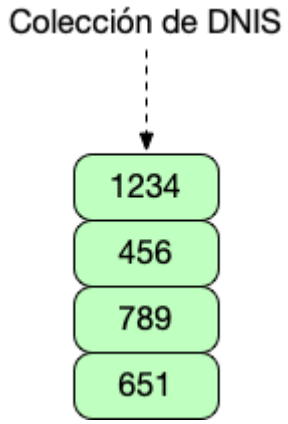
De igual forma nosotros podemos acceder a cualquier elemento del mapa a traves de su método get ya que este nos solicita la clave que introdujimos anteriormente y nos devuelve el valor que en este caso no es ni más ni menos que la Persona que habiamos guardado .



Veamoslo en código:

```
Persona persona4= mapa.get("123");
                System.out.println(persona4.getDni());
                System.out.println(persona4.getDni());
                System.out.println(persona4.getDni());
```

Ahora bien ¿Como podemos recorrer el mapa por completo y acceder a todos sus valores ? Para recorrer el mapa deberemos usar un bucle que accede a cada una de las entradas del mapa (entries). Estas entradas componen un conjunto de elementos no repetidos dentro del mapa como si fueran claves primarias de una tabla.



El código es algo mas complejo:

```
                for(Map.Entry<String,Persona> entrada:
mapa.entrySet()) {
                                Persona otra= entrada.getValue();
                                System.out.println(otra.getDni());
                                System.out.println(otra.getNombre());
                                System.out.println(otra.getEdad());
                }
```

El resultado se imprime por la consola:



¿Podemos simplificarlo? : La realidad es que sí ya que a partir del JDK 10 podemos apoyarnos en la nueva palabra reservada var que nos dejara las cosas más sencillas:

```
for(var entrada: mapa.entrySet()) {
                                Persona otra= entrada.getValue();
                                System.out.println(otra.getDni());
                                System.out.println(otra.getNombre());
                                System.out.println(otra.getEdad());
                }
```

Java siempre sigue evolucionando y en este caso nos aporta a la hora de realizar un Java Map for loop.

Otros artículos relacionados

- Java ArrayList for y sus opciones
- Java 9 Factory Methods (List,Set,Map)
- Java List Directory en Java 8 con Streams
- Java List to Map y el uso de Collectors