El uso de Java varargs es poco conocido dentro del mundo Java y a veces nos pueden ser realmente útiles .¿Para que sirven los varargs? . Son métodos que permiten variar el número de parámetros que reciben permitiendo trabajar de una forma más cómoda. Vamos a ver un ejemplo usando los conceptos de Persona y Factura y relacionandolos, una Persona tiene varias Facturas.

```
package com.arquitecturajava.varargs;

public class Factura {

private int numero;

public int getNumero() {
return numero;
}

public void setNumero(int numero) {
this.numero = numero;
}

public Factura(int numero) {
super();
this.numero = numero;
}

}
```

```
package com.arquitecturajava.varargs;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Persona {

private String nombre;
private List<Factura> facturas= new ArrayList<>();
public String getNombre() {
return nombre;
}

public void setNombre(String nombre) {
this.nombre = nombre;
}

public void addFactura(Factura f) {

facturas.add(f);
}

}
```

Java varargs

Si queremos añadir facturas a un objeto Persona deberemos escribir el siguiente bloque de código:

```
package com.arquitecturajava.varargs;

public class Principal {

public static void main(String[] args) {

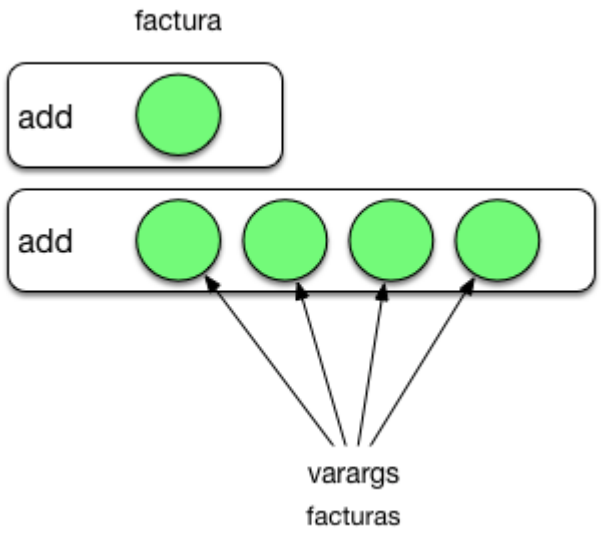
Factura f= new Factura (1);
Factura f2= new Factura (2);
Factura f3= new Factura (3);
Persona p= new Persona();

p.addFactura(f);
p.addFactura(f2);
p.addFactura(f3);

}

}
```

Podemos simplificarlo usando Java varargs y generando un nuevo método en la clase Persona que admita un número variable de argumentos.



Vamos a verlo en código:

```
public void addFactura(Factura ... variasFacturas ){


for(Factura f:variasFacturas) {
facturas.add(f);
}
}
```

Los Java varargs se identifican por ser métodos que contienen varios puntos , permitiendo pasar varios parámetros del mismo tipo. Una vez construido el método añadir facturas a la Persona será más sencillo.

```
package com.arquitecturajava.varargs;

public class Principal {

public static void main(String[] args) {

Factura f= new Factura (1);
Factura f2= new Factura (2);
Factura f3= new Factura (3);
Persona p= new Persona();
p.addFactura(f,f2,f3);

}

}
```

Muchas veces las personas me preguntan si hay algún método vararg en el propio framework de colecciones ya que resultaría práctico añadir varios objetos de golpe a un ArrayList. Esto no esta soportado directamente por el framework pero sí por la clase Collections que es una de las clásicas de utilidades. Podríamos cambiar la implementación del método por la siguiente:

```
public void addFactura(Factura ... variasFacturas ){

Collections.addAll(facturas, variasFacturas);


}
```

De esta forma añadiremos directamente a la colección la lista de facturas.

Otros artículos relacionados: Java Collections Views , Java List vs Set , Java Generics