



ESCUELA SUPERIOR DE COMPUTO



# PROGRAMACIÓN ORIENTADA A OBJETOS

## PRÁCTICA 4

Tipos, Variables, Métodos y Constructores

Comprender, utilizar y aplicar correctamente los tipos en variables utilizados en el lenguaje de programación Java

**Rubio Haro Rodrigo R.**

CDMX. MAYO, 2020.

INSTITUTO POLITÉCNICO NACIONAL

# Práctica 4: Tipos, variables, Métodos y Constructores

## 1. Introducción

El uso apropiado de los tipos de datos dispuestos por el lenguaje de programación Java, le da una potencia enorme, esto a pesar de que Java es un lenguaje orientado a objetos. Un valor constante en Java se crea utilizando una representación literal de él. Java utiliza cinco tipos de elementos: enteros, reales en coma flotante, booleanos, caracteres y cadenas, que se pueden poner en cualquier lugar del código fuente de Java. Cada uno de estos literales tiene un tipo correspondiente asociado con él.

## 2. Desarrollo

### 2.1 Tipos de Datos

#### 2.1.1 Código

```
public class Datos {  
    private int _entero;  
    private float _float;  
    private byte _byte;  
    private char _char;  
    private short _short;  
    private long _long;  
    private double _doble;  
    private boolean _boolean;  
  
    public Datos() {  
        int n=10000;  
        _char= 'c';  
        _entero = (int) ((Math.random() * n) + 1);  
        _float = (float) ((Math.random() * n) + 1);  
        _byte = (byte) ((Math.random() * n) + 1);  
        _short = (short) ((Math.random() * n) + 1);  
        _long = (long) ((Math.random() * n) + 1);  
        _doble = (double) ((Math.random() * n) + 1);  
        _boolean = (boolean) (Math.random() < 0.5);  
    }  
}
```

```
        public void imprimirDatos() {
            System.out.println("dato entero: " + _entero);
            System.out.println("dato float: " + _float);
            System.out.println("dato byte: " + _byte);
            System.out.println("dato char: " + _char);
            System.out.println("dato short: " + _short);
            System.out.println("dato long: " + _long);
            System.out.println("dato double: " + _doble);
            System.out.println("dato bool: " + _boolean);
        }
    }

    public class TiposDeDatos {
        public static void main(String[] args) {
            Datos lista = new Datos();
            lista.imprimirDatos();
        }
    }
}
```

## 2.1.2 Compilación y Ejecución

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/1.TiposDeDatos (master)
$ javac *.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/1.TiposDeDatos (master)
$ java TiposDeDatos
dato entero: 7999
dato float: 640.4004
dato byte: -80
dato char: c
dato short: 2648
dato long: 7108
dato double: 9299.386237300821
dato bool: false

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/1.TiposDeDatos (master)
$
```

Resultado de compilar y ejecutar el código.

## 2.2 Desbordamiento

### 2.2.1 Código

```
public class Datos {
    private int _entero;
    private float _float;
    private byte _byte;
    private char _char;
    private short _short;
    private long _long;
    private double _doble;
```

```

public Datos() {
    _char = (char) 128;
    _entero = (int) 2147483648L;
    _float = (float) (1.79769313486231570E+308 + 1);
    _byte = (byte) 256;
    _short = (short) 32768;
    _long = (long) 9223372036854775808D;
    _doble = (double) 1.79769313486231570E+308 + 1;
}

public void imprimirDatos() {
    System.out.println("dato entero: " + _entero);
    System.out.println("dato float: " + _float);
    System.out.println("dato byte: " + _byte);
    System.out.println("dato char: " + _char);
    System.out.println("dato short: " + _short);
    System.out.println("dato long: " + _long);
    System.out.println("dato double: " + _doble);
}
}

public class TiposDeDatos {
    public static void main(String[] args) {
        Datos lista = new Datos();
        lista.imprimirDatos();
    }
}

```

## 2.2.2 Compilación y Ejecución

```

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/2.DesbordamientoTiposDeDatos (master)
$ javac *.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/2.DesbordamientoTiposDeDatos (master)
$ java TiposDeDatos
dato entero: -2147483648
dato float: Infinity
dato byte: 0
dato char: ?
dato short: -32768
dato long: 9223372036854775807
dato double: 1.7976931348623157E308

```

## 2.3 Matrices

### 2.3.1 Código Estructurado

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/P00/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/Estructurado (master)
$ java DosDim
0
1 2
3 4 5
6 7 8 9
```

Se compiló, ejecutó y analizó el primer programa.

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/P00/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/Estructurado (master)
$ javac DosDimensiones.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/P00/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/Estructurado (master)
$ java DosDimensiones
0.0 0.0 0.0 0.0
0.0 1.0 2.0 3.0
0.0 2.0 4.0 6.0
0.0 3.0 6.0 9.0
```

Se compiló, ejecutó y analizó el segundo programa.

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/P00/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/Estructurado (master)
$ java tresDMatriz
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12

0 0 0 0 0
0 2 4 6 8
0 4 8 12 16
0 6 12 18 24
```

Se compiló, ejecutó y analizó el tercer programa.

### 2.3.2 Código Orientado a Objetos

El código estructurado de los tres ejercicios de compiló y solo uno presentó un pequeño error. El código se llevó al paradigma orientado a objetos. El código se podrá encontrar en el repositorio de POO/ESCOM o habrá sido enviado al profesor. Las clases principales fueron (las clases principales se ha omitido en este documento):

```
public class Matriz {
    private double[][] valores;

    public Matriz() {}

    public Matriz(double[][] valores) {
        this.valores = valores;
    }

    public void imprimir() {
        for (int i = 0; i < valores.length; i++) {
            for (int j = 0; j < valores[i].length; j++)
                System.out.print(valores[i][j] + " ");
            System.out.println();
        }
    }
}

public class MatrizTridimensional {

    private double[][][] valores;
    public MatrizTridimensional() {}

    public MatrizTridimensional(double[][][] valores) {
        this.valores = valores;
    }

    public void imprimir() {
        for (int i = 0; i < valores.length; i++) {
            for (int j = 0; j < valores[i].length; j++) {
                for (int k = 0; k < valores[i][j].length; k++)
                    System.out.print(valores[i][j][k] + " ");
                System.out.println();
            }
            System.out.println();
        }
    }
}
```



### 2.3.3 Ejecución del Código Orientado a Objetos

El código, aunque fue modificado, para los tres casos, produjo, prácticamente el mismo resultado. Primer programa:

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/OrientadosAObjetos (master)
$ javac DosDimensiones.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/OrientadosAObjetos (master)
$ java DosDimensiones
0.0
1.0 2.0
3.0 4.0 5.0
6.0 7.0 8.0 9.0
```

Segundo programa:

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/OrientadosAObjetos (master)
$ javac Multiplicaciones.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/OrientadosAObjetos (master)
$ java Multiplicaciones
0.0 0.0 0.0 0.0
0.0 1.0 2.0 3.0
0.0 2.0 4.0 6.0
0.0 3.0 6.0 9.0
```

Tercer programa:

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/OrientadosAObjetos (master)
$ javac TresDimensiones.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departa
tica04_2CV1_Departamental1/3.Matrices/OrientadosAObjetos (master)
$ java TresDimensiones
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0
0.0 1.0 2.0 3.0 4.0
0.0 2.0 4.0 6.0 8.0
0.0 3.0 6.0 9.0 12.0

0.0 0.0 0.0 0.0 0.0
0.0 2.0 4.0 6.0 8.0
0.0 4.0 8.0 12.0 16.0
0.0 6.0 12.0 18.0 24.0
```

## 2.4 Conversiones en JAVA

Se crearon 3 clases, la de la interfaz (main), una para el menú y la de las conversiones, compiló sin problemas.

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/D
roblemas/1.Conversiones (master)
$ javac *.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/D
roblemas/1.Conversiones (master)
$ java InterfazConversiones

*****
CONVERSIONES DE TIPOS EN JAVA
*****

Que tipo de conversiones quieres ver?
1, Conversiones Automaticas
2, Conversiones Explicitas

Ingresa una opcion: 1

Conversiones Automaticas
***** Generales *****
byte -> short -> int-> long -> float -> double

original byte:58
byte to short:58
short to int:58
int to long:58
lonh to float:58.0
float to double:58.0
```

Se dividieron en dos las conversiones, las automáticas y las explícitas, seleccionando entre una u otra desde el menú.



```
problemas/1.Conversiones (master)
```

```
$ java InterfazConversiones
```

```
*****
```

## CONVERSIONES DE TIPOS EN JAVA

```
*****
```

Que tipo de conversiones quieres ver?

1, Conversiones Automaticas

2, Conversiones Explicitas

Ingrese una opcion: 2

### Conversiones Explicitas

```
***** Generales *****
```

```
double -> float -> long-> int -> short -> byte
```

En estas conversiones podemos tener problemas de desbordamiento

```
***** Generales *****
```

```
original double:58.6
```

```
double -> float:58.6
```

```
float -> long:58
```

```
long -> int:58
```

```
int -> short:58
```

```
short -> byte:58
```

Ademas podemos realizar las conversiones intermedias:

```
***** double *****
```

```
new double:59.6
```

```
double -> long:59
```

```
double -> int:59
```

```
double -> short:59
```

```
double -> byte:59
```

```
***** float *****
```

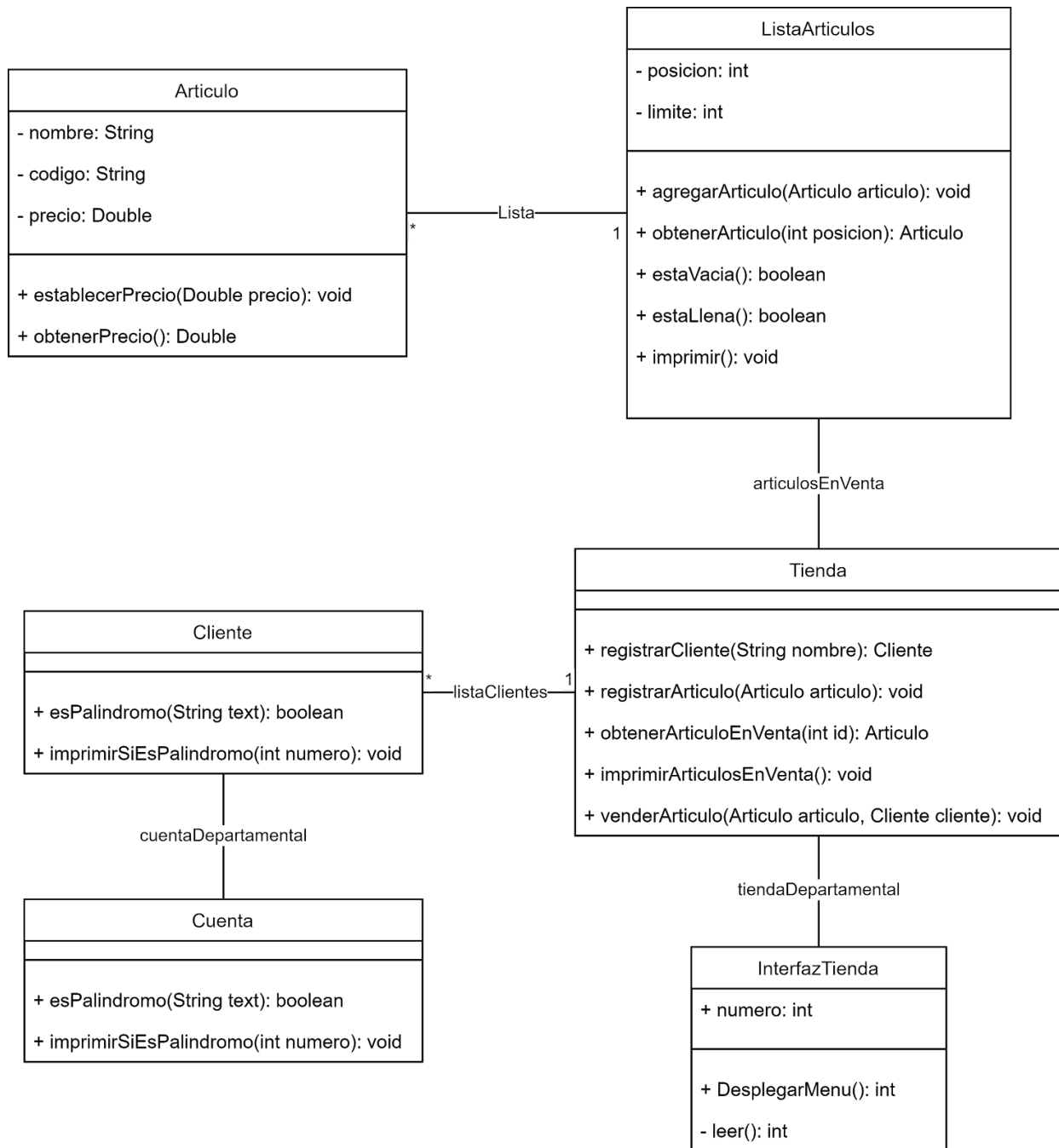
```
new float:60.6
```

```
float -> int:60
```

```
float -> short:60
```

```
float -> byte:60
```

## 2.5 Tienda Departamental

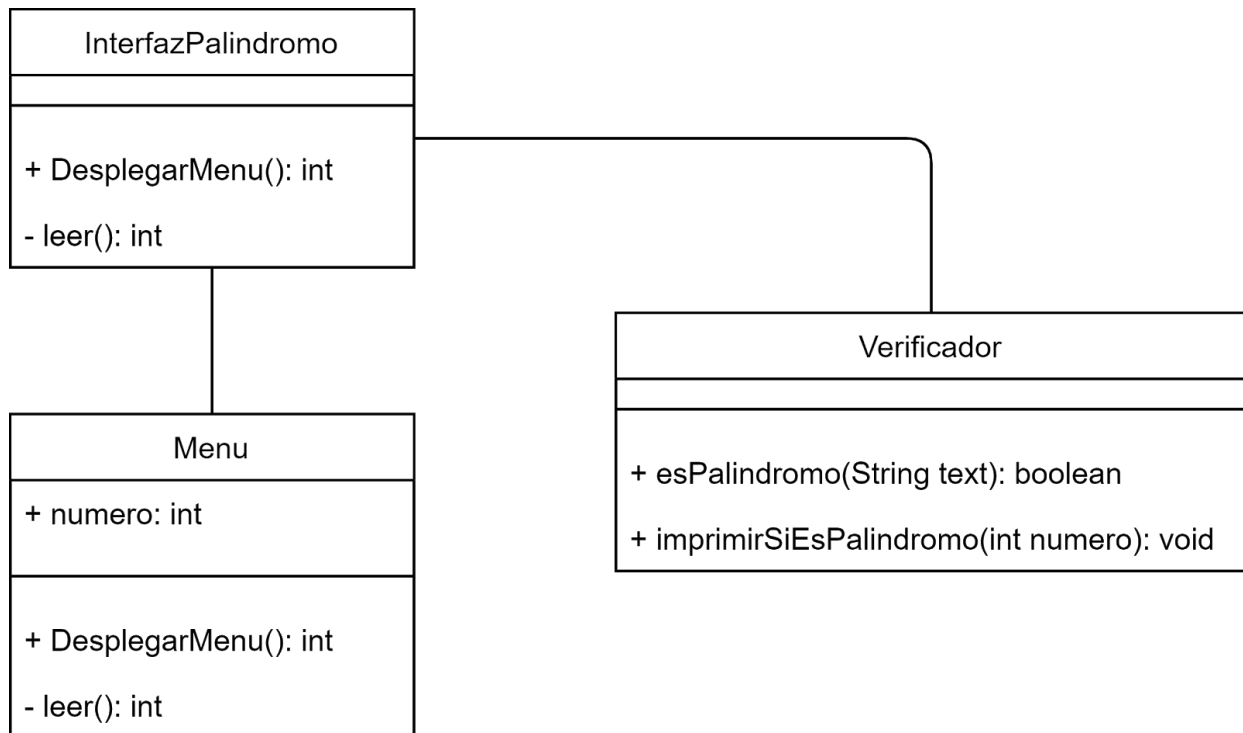


## 2.5.1 Ejecución del Código

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_D
rtamental1/4.Problemas/2.Departamental (master)
$ javac *.java
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_D
rtamental1/4.Problemas/2.Departamental (master)
$ java InterfazTienda
Mov. cargado:Articulo: OMEN 15, codigo:HPO15X, precio: 15000.0 El nuevo saldo es:15000.0 Cliente: Nombre: Jhon Snow
Mov. cargado:Articulo: USB 3.0 Kingston, codigo:KGSB23, precio: 200.0 El nuevo saldo es:15200.0 Cliente: Nombre: Jhon Snow
Mov. cargado:Articulo: EPSON 374, codigo:EPS374, precio: 1000.0 El nuevo saldo es:16200.0 Cliente: Nombre: Jhon Snow
Mov. cargado:Articulo: EPSON 374, codigo:EPS374, precio: 1000.0 El nuevo saldo es:17200.0 Cliente: Nombre: Jhon Snow
Mov. cargado:Articulo: EPSON 374, codigo:EPS374, precio: 1000.0 El nuevo saldo es:18200.0 Cliente: Nombre: Jhon Snow
Se ha excedido el limite de credito, no se ha podido cargar:Articulo: OMEN 15, codigo:HPO15X, precio: 15000.0 Tu saldo es: 18200.0mont
oACargar:33200.0 Cliente: Nombre: Jhon Snow
*****
Datos del cliente:
ESTADO DE CUENTA: 992-122-212-111
Articulo: OMEN 15, codigo:HPO15X, precio: 15000.0
Articulo: USB 3.0 Kingston, codigo:KGSB23, precio: 200.0
Articulo: EPSON 374, codigo:EPS374, precio: 1000.0
Articulo: EPSON 374, codigo:EPS374, precio: 1000.0
Articulo: EPSON 374, codigo:EPS374, precio: 1000.0
Saldo Inicial: $0.0 pesos al 1/03/2020 12:00:00 AM
Saldo Actual: $18200.0 pesos al 18/05/2020 10:44:13 PM
limite de credito: $30000.0 pesos. Credito restante: $11800.0
*****
```

## 2.6 Palíndromo



Se compiló y ejecutó el programa.

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/4.Problemas/3.Palindromo (master)
$ javac *.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/4.Problemas/3.Palindromo (master)
$ java InterfazPalindromo
Ingresa un palindromo:111111
111111 Es un palindromo

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/Practicas/Rubio_Haro_Rodrigo_Rodolfo_Practica04_2CV1_Departamental1/4.Problemas/3.Palindromo (master)
$ java InterfazPalindromo
Ingresa un palindromo:25424
25424 NO es un palindromo
```

El programa funcionó sin ningún problema.

### 3. Conclusión

Manejar tipos de datos en java es más complicado sobre otros lenguajes, sin embargo nos permite tener más control y certeza sobre estos, fue la parte, en mi opinión más importante de la práctica.