

# Cuaderno de ejercicios: Código limpio

[Solución: Cuaderno de ejercicios: Código limpio](#)

Ejercicio 1:	1
Ejercicio 2:	2
Ejercicio 3:	3
Ejercicio 4	4
Ejercicio 5	4
Ejercicio 6	5
Ejercicio 7	6

## Ejercicio 1:

Corrige el código para que sea código limpio.

```
import java.util.Arrays;
public class Book {
    String ISBN;
    // Titulo del libro
    String t;
    String[] autores;
    // Editorial del libro
    String e;
    Edition[] editions;

    public Book(String ISBN) {
        this.ISBN=ISBN;
    }

    int count_auhors() {
        return autores.length;
    }

    void addNewEdition(String name, String data, int id) {
        Edition[] newArray = Arrays.copyOf(editions,
        editions.length+1);
```

```

        newArray[editions.length] = new Edition(name, data, id);
        editions = newArray;
    }
}

class Edition{
    String name;
    String data;
    int id;

    public Edition(String name, String data, int id) {
        super();
        this.name = name;
        this.data = data;
        this.id = id;
    }
}

```

## Ejercicio 2:

Corrige el código para que sea código limpio.

```

public class Car {
    String model;
    String brand;
    String registration;
    // tipo de vehiculo. Matricula antigua y moderna
    String type = "NEW";

    boolean validateRegistration() {
        if(type.equals("OLD")) {
            if(registration.length() == 8) {
                return true;
            }
            else {
                return false;
            }
        }else {
            if(registration.length() == 7) {

```

```

        return true;
    }
    else {
        return false;
    }
}
}
}
}

```

### Ejercicio 3:

Corrige el código para que sea código limpio.

```

public class User {
    String name;
    // Tipo de usuario: Profesor, alumno o director
    String type;

    public User(String name, String type) {
        this.name=name;
        this.type=type;
    }

    float calculateCostOfUseOfTheLibrary() {
        switch(type) {
            case "USER": return 10;
            case "TEACHER": return 2;
            case "HEAD_MASTER": return 0;
            default: return 0;
        }
    }
}
}

```

## Ejercicio 4

Corrige el código para que sea código limpio.

```
public class Factura {
    String cod;        String nombre;    String direccion;
    String cif;        String fecha;    float total;
    String nombre1;    float precio1;
    String nombre2;    float precio2;
    String nombre3;    float precio3;
    String nombre4;    float precio4;

    public static String getDayOfWeekByNumber(int numberDayOfWeek) {
        if(numberDayOfWeek == 1) {
            return "Lunes";
        }else if(numberDayOfWeek == 2) {
            return "Martes";
        }else if(numberDayOfWeek == 3) {
            return "Miercoles";
        }else if(numberDayOfWeek == 4) {
            return "Jueves";
        }else if(numberDayOfWeek == 5) {
            return "Viernes";
        }else if(numberDayOfWeek == 6) {
            return "Sabado";
        }else if(numberDayOfWeek == 7) {
            return "Domingo";
        }
        return "error";
    }
}
```

## Ejercicio 5

Corrige el código para que sea código limpio.

Haz también el testing

```
public class Person {
    // Nombre de la persona
    String n;
    String surname;
    String direction;
```

```

public void getPersonInitials(){
    // limpiamos el campo name
    String nameInLowerCase = name.toLowerCase();
    String nameWithoutSpaces = nameInLowerCase.trim();
    String firstLetterOfName = nameWithoutSpaces.charAt(0)+"";

    // limpiamos el campo surname
    String surnameInLowerCase = surname.toLowerCase();
    String surnameWithoutSpaces = surnameInLowerCase.trim();
    String firstLetterOfSurname =
surnameWithoutSpaces.charAt(0)+"";

    System.out.println( firstLetterOfName + " " +
firstLetterOfSurname );
}
}

```

## Ejercicio 6

Corrige el código para que sea código limpio.

```

public class Robot {
    String name = "";
    String type = "biped";
    int numLegs = 2;
    int leg1;
    int leg2;
    int leg3;
    int leg4;
    int leg5;
    int leg6;

    public Robot(String name, String type, int numLegs) {
        this.name=name;
        this.type=type;
        this.numLegs = numLegs;
    }
    public void move() {
        if(type.equals("biped")) {
            // mover las 2 piernas
            leg1+=1;

```

```

        leg2+=1;
    }else if(type.equals("cuadruped")) {
        // mover las 4 piernas
        leg1+=1;
        leg2+=1;
        leg3+=1;
        leg4+=1;
    }else if(type.equals("hexapod")) {
        // mover las 6 piernas
        leg1+=1;
        leg2+=1;
        leg3+=1;
        leg4+=1;
        leg5+=1;
        leg6+=1;
    }
}
}
}

```

## Ejercicio 7

Corrige el código para que sea código limpio.

```

public class CoffeeMaker {

    public boolean makeCoffe(String coffeeType, float temp, float
waterAmount) {
        if(!coffeeType.equals("")) {
            if(temp>0 && temp<100) {
                if(waterAmount > 10) {
                    // hacer café
                    return true;
                }else {
                    System.out.println("nivel agua
incorrecto");

                    return false;
                }
            }else {
                System.out.println("temperatura incorrecta");
                return false;
            }
        }else {
            System.out.println("tipo de café incorrecto");
        }
    }
}

```

```
        return false;
    }
}
}
```

## Ejercicio 8:

Corrige el código para que sea código limpio.

```
public class EjemploSucio {
    public static float dividir(int x) {
        return x/2f;
    }
    public static int calcular(int x, int y) {
        int z = x+y;
        return x+y;
    }
    public static int calcular2(int x) {
        x=x*x;
        System.out.println(x);
        return x;
    }
    public static void main(String[] args) {
        int b = calcular(3,4);
        System.out.println(b);
        calcular2(4);
    }
}
```

## Ejercicio 9:

Corrige el código para que sea código limpio.

```
class BarcoSuc{
    //Velocidad actual
    int va;
    //Velocidad maxima
    int vmax = 100;
    void arrancar() {
        va=3;
    }
}

class CocheSuc{
    //Velocidad actual
    int va;
    //Velocidad maxima
    int vmax = 120;
    void arrancar() {
        va=10;
    }
}

class PeugeotSuc{
    //Velocidad actual
    int va;
    //Velocidad maxima
    int vmax = 110;
    void arrancar() {
        va=10;
    }
}

class FerrariSuc{
    //Velocidad actual
    int va;
    //Velocidad maxima
    int vmax = 320;
    void arrancar() {
        va=10;
    }
}

public class EjemploSucio {
```



```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
  
}  
  
}
```

## Ejercicio 10:

Corrige el código para que sea código limpio.

```
import java.util.Date;  
  
public class EjemploSucio {  
    public static void matricular(String nombre, String apellido1,  
        String apellido2, Date fechaAlta, String asignatura) {  
        // acceso a la base datos  
        // inserta  
    }  
  
    public static void main(String[] args) {  
        matricular("Jose", "Gonzalez", "Ruiz", new Date(), "Sistemas  
informaticos");  
    }  
}
```

## Ejercicio 11:

Corrige el código para que sea código limpio.

```
public class EjemploSucio {  
  
    public static void main(String[] args) {  
        // Relleno un array  
        int datos[] = {2,9,3,4,1};  
  
        // Ordenamos el array  
    }  
}
```

```

int aux;
for(int i=0;i<datos.length-1; i++) {
    for(int j=0;j<datos.length-i-1; j++) {
        if(datos[j+1] < datos[j]) {
            // datos = aux -1;
            aux = datos[j+1];
            datos[j+1] = datos[j];
            datos[j] = aux;
        }
    }
}

// pintamos en pantalla el array
for(int i=0; i<datos.length;i++) {
    System.out.println(datos[i]);
}
}
}

```

## Ejercicio 12:

Corrige el código para que sea código limpio.

```

class premioSocio {
    String tipo;
    String material;

    float get_coste_material() {
        if (material == "oro")
            return 1000;
        else if (material == "plata")
            return 100;
        else if (material == "bronce")
            return 10;
        else
            return 0;
    }

    float getCosteTipo() {
        switch (tipo) {
            case "medalla":
                return 200;
            case "copa":

```

```

        return 2000;
    case "pin":
        return 2;
    default:
        return 0;
    }
}

float getimporte() {
    return get_coste_material()
        * getCosteTipo() * 0.5f;
}

}

public class EjemploLimpio {

    public static void main(String[] args) {

    }

}

```

## Ejercicio 13

1. Documentarlo
2. Crear los tests
3. Refactorizarlo (código limpio)
4. Optimizarlo

```

public class DocumentoIdentificativo {
    // Puede ser CIF o NIF
    String type;
    String text;

    String getLetter() {
        String letraEncontrada = "";
        if(type == "CIF") {
            char temp = text.charAt(0);
            letraEncontrada = String.valueOf(temp);
        }else if(type == "NIF"){

```

```

        char temp = text.charAt(text.length()-1);
        letraEncontrada = String.valueOf(temp);
    }else {
        System.out.println("Error en el tipo de documento");
    }
    return letraEncontrada;
}

boolean isValid() {

    if(type=="CIF") {
        boolean tieneLetra = false;
        boolean tieneNum1 = false;
        boolean tieneNum2 = false;
        boolean tieneNum3 = false;
        boolean tieneNum4 = false;
        boolean tieneNum5 = false;
        boolean tieneNum6 = false;
        boolean tieneNum7 = false;
        boolean tieneNum8 = false;

        if( Character.isAlphabetic(text.charAt(0)) == true) {
            tieneLetra = true;
        }

        if( Character.isDigit(text.charAt(1)) == true) {
            tieneNum1 = true;
        }

        if( Character.isDigit(text.charAt(2)) == true) {
            tieneNum2 = true;
        }

        if( Character.isDigit(text.charAt(3)) == true) {
            tieneNum3 = true;
        }

        if( Character.isDigit(text.charAt(4)) == true) {
            tieneNum4 = true;
        }

        if( Character.isDigit(text.charAt(5)) == true) {
            tieneNum5 = true;
        }
    }
}

```

```
        if( Character.isDigit(text.charAt(6)) == true) {
            tieneNum6 = true;
        }

        if( Character.isDigit(text.charAt(7)) == true) {
            tieneNum7 = true;
        }
        if( Character.isDigit(text.charAt(8)) == true) {
            tieneNum8 = true;
        }

        if(tieneLetra && tieneNum1 && tieneNum2 && tieneNum3
&& tieneNum4
            && tieneNum5 && tieneNum6 && tieneNum7 &&
tieneNum8) {
            return true;
        }else {
            return false;
        }

    }else if(text == "NIF") {
        //TODO falta por hacer
        return false;
    }
    return false;
}

}
```