

## UNIDAD TEMÁTICA 1: Introducción a JAVA

### PRACTICOS INDIVIDUALES – USO DEL SISTEMA DE VERSIONADO

NOTA: PARA HACER ESTE EJERCICIO DEBE HABER AL MENOS 2 COMPUTADORAS SEPARADAS “*MáquinaA*” y “*MáquinaB*”...

## Paso 1

En dos máquinas separadas

En *Máquina A*:

Crear la siguiente clase Perro

```
1  public class Perro {  
2  
3      private String nombre;  
4  
5      public Perro(String nombre){  
6          this.nombre = nombre;  
7      }  
8      public String getNombre(){  
9          return this.nombre;  
10     }  
11     public void ladrar(){  
12         System.out.println("wof");  
13     }  
14     public boolean respondeA(String unNombre){  
15         return this.nombre.equals(unNombre);  
16     }  
17 }
```

- Probarlo con una clase *MainMaquinaA*.
- Subir los cambios

En **Maquina B**:

```
1 public class Gato {  
2  
3     private String nombre;  
4  
5     public Gato(String nombre){  
6         this.nombre = nombre;  
7     }  
8     public String getNombre(){  
9         return this.nombre;  
10    }  
11    public void maullar(){  
12        System.out.println("miau");  
13    }  
14    public boolean respondeA(String unNombre){  
15        return false; //NUNCA RESPONDEN!!!  
16    }  
17 }
```

- Probarlo con una clase **MainMaquinaB**.
- Subir cambios.

## Paso 2

- Sincronizar en las dos máquinas.
- Probar los métodos cruzados.

En la **Máquina B**, probar los métodos de Perro desde la clase **MainMaquinaB** (se unifican el método **Main**).

En este caso no hay conflictos, ya que desde **Máquina A** no se había modificado ese archivo.

## Paso 3

En la **Máquina A**, crear una Interfaz **Mamifero** y subir los cambios.

```
1 public interface Mamifero {  
2  
3     public String getNombre();  
4     public boolean respondeA(String unNombre);  
5  
6     public void caminar();  
7     public void correr();  
8     public void saltar();  
9 }
```

## Paso 4

- En la **Máquina A**, hacer los cambios correspondientes para que **Perro** implemente esa interfaz.
- En la **Máquina B**, bajarse los cambios, y hacer los cambios correspondientes para que **Gato** implemente esa interfaz.
- Al finalizar, ambos deberán sincronizar (subir y bajar los cambios pertinentes)

## Paso 5

En el **MainMaquinaB**, ambos equipos ejecuten los nuevos métodos implementados y suban sus cambios.

Esta acción resultará en un conflicto, por lo que deberán hacer un **merge**.

Solucionen los conflictos **en una máquina**, y suban los cambios.

**Posible resultado de merge:**

```
2
3 public class MainMaquinaB {
4
5     public static void main(String[] args){
6
7         Gato gato = new Gato("Carmela");
8         Perro unPerro = new Perro("Dunkel");
9         System.out.println("El gato responde a Pocho? : "+gato.respondeA("Pocho"));
10        System.out.println("Responde a Juan "+unPerro.respondeA("Juan"));
11        System.out.println("Responde a Dunkel "+unPerro.respondeA("Dunkel"));
12
13
14        //Maquina A
15        unPerro.caminar();
16        unPerro.saltar();
17
18        //Maquina B
19        gato.maullar();
20        gato.correr();
21    }
22 }
```