



Has completado este test el *07/06/2022, 09:07*  
Tu calificación es 100.00%

CORRECTO

¿Cual es la mala práctica y como se corrige?

```
let str = '';  
str += 'foo';  
str += 'bar';  
str += 'baz';
```

La mala práctica es haber llamado la variable "str" tres veces para concatenarle caracteres que se podían haber puesto al declarar la variable o solamente llamándola una vez.

CORRECTO

¿Cual es la mala práctica y como se corrige?

```
class Employee {  
  eat() {  
    //...  
  }  
  drink() {  
    //...  
  }  
  work() {  
    //...  
  }  
  sleep() {  
    ///...  
  }  
  walk() {  
    //...  
  }  
}
```

La mala práctica en este caso es hacer que la clase "Employee" se encargue de muchas cosas, esto se podría solucionar creando otra clase llamada "Human" que se encargue de las tareas como (eat, drink, sleep y walk) y dejar únicamente al "Employee" la responsabilidad de work, haciendo que esta herede de "Human".

CORRECTO

## ¿Cual es la mala práctica y como se corrige?

```
void printOwing() {  
    printBanner();  
    // Print details.  
    System.out.println("name: " + name);  
    System.out.println("amount: " + getOutstanding());  
}
```

En este caso la mala práctica estaría en el anidamiento de funciones, ya que si "printBanner" solamente imprime un Banner no es necesario llamarlo dentro de la función de "printOwing" puesto que la podríamos llamar junto con esta y obtendríamos el mismo resultado.

Ejemplo:

```
printBanner();  
printOwing();
```

---

CORRECTO

## ¿Cual es la mala práctica y como se corrige?

```
boolean hasDiscount(Order order) {  
    double basePrice = order.basePrice();  
    return basePrice > 1000;  
}
```

La mala práctica estaría en que es un método muy simple que fácilmente se podría sustituir su uso por una condición y se llegaría al mismo resultado. Se podría solucionar cambiando los llamados a este método con una condicional que verifique si `order.basePrice() > 1000` y listo.

---

CORRECTO

## ¿Cual es la mala práctica y como se corrige?

```
class Range {  
    private int low, high;  
    boolean includes(int arg) {  
        return arg >= low && arg <= high;  
    }  
}
```

La mala práctica aquí es que se está creando una clase para resolver un problema que se puede resolver utilizando una función o método que haga lo mismo. Esto se puede solucionar haciendo una función/método que reciba dos int como parámetros y que haga la misma comprobación, de esta manera no se estaría creando una clase solamente para resolver este problema.

---

CORRECTO

¿Cual es la mala práctica y como se corrige?

```
public String board() {
    StringBuilder buf = new StringBuilder();

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            buf.append(data[i][j]);
        }
        buf.append("\n");
    }
}
```

Aquí la mala practica está en la indentación del código, ya que se tiene un bucle for dentro de otro. Esto se puede solucionar extrayendo el for interior y colocándolo en un método aparte, de esta manera en el primer for simplemente se llama al método recién creado y se mantiene un orden de indentación ideal, bajando el nivel de complejidad al código.

---

CORRECTO

¿Cual es la mala práctica y como se corrige?

```
public void login(String username, String
password) { if
(userRepository.isValid(username, password)) {
redirect("homepage");
} else {
    addFlash("error", "Bad credentials");

    redirect("login");
}
}
```

En este caso la mala práctica se encuentra en el uso del "else", ya que este puede generarnos un nivel de complejidad innecesario. Esto se puede solucionar dejando como acción por defecto el método "addFlash" y el "redirect" y en caso de cumplirse la condición en el if que haga el redireccionamiento a la homepage.

