

Tema 7 Complejidad Condicional:

La complejidad condicional hace que el código sea más complicado de entender y, por lo tanto, de mantener. Además, la complejidad condicional suele ser un indicador de que el código está acoplado.

Problema:

La complejidad condicional se presenta cuando en un caso inicial se crea un código complejo y/o poco entendido para resolver un problema que se da en un momento, pero con el paso del tiempo y al querer actualizar el código este presenta un gran problema ya que por su complejidad no se puede actualizar o reemplazar partes del mismo haciendo que pierda calidad.

Consecuencias:

- El código se vuelve complicado de entender y mantener.
- La corrección de errores del código se vuelve casi imposible.
- El código corre el riesgo de volverse obsoleto.

Causas:

- Sobre analizar las formas de resolver un problema.
- Mal análisis e implementación del código por falta de tiempo.
- Falta de ética del programador

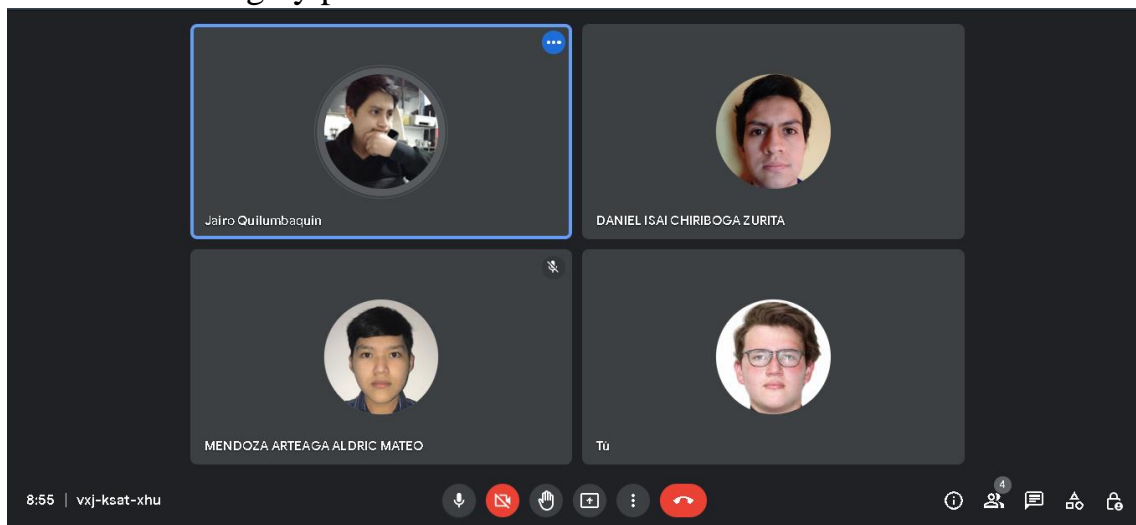
Ejemplo:

```
1  function getPayAmount() {
2      let result;
3      if (isDead){
4          result = deadAmount();
5      }else {
6          if (isSeparated){
7              result = separatedAmount();
8          } else {
9              if (isRetired){
10                 result = retiredAmount();
11             }else{
12                 result = normalPayAmount();
13             }
14         }
15     }
16     return result;
17 }
```

Como podemos observar en el primer ejemplo el código contiene una gran cantidad de condicionales “if” anidados dificultando la lectura y comprensión del código en cuestión.

```
1  function getPayAmount() {  
2    if (isDead) return deadAmount();  
3    if (isSeparated) return separatedAmount();  
4    if (isRetired) return retiredAmount();  
5    return normalPayAmount();  
6  }
```

Al limpiar el código y retirar los condicionales “else” resulta más fácil entender el código y por ende en un futuro mantenerlo será fácil



Code with Conditional Complex:

```

public static void main(String[] args) {
    Scanner input;
    int type;

    input= new Scanner(System.in);

    System.out.println("Seleccione el tipo de archivo que desea crear");
    System.out.println("1.PDF\n");
    System.out.println("2.DOC\n");
    System.out.println("3.CSV\n");
    System.out.println("4.TXT\n");
    type=input.nextInt();

    if(type==1){
        System.out.println("PDF was created");
    }else{
        if(type==2){
            System.out.println("DOC was created");
        }else{
            if(type==3){
                System.out.println("CSV was created");
            }else{
                if(type==4){
                    System.out.println("TXT was created");
                }else{
                }
            }
        }
    }
}

```

Code refactores:

```

public static void main(String[] args) {
    Scanner input;
    int type;

    input= new Scanner(System.in);

    System.out.println("Seleccione el tipo de archivo que desea crear");
    System.out.println("1.PDF\n");
    System.out.println("2.DOC\n");
    System.out.println("3.CSV\n");
    System.out.println("4.TXT\n");
    type=input.nextInt();

    switch(type){
        case 1:
            System.out.println("PDF was created");break;
        case 2:
            System.out.println("DOC was created");break;
        case 3:
            System.out.println("CSV was created");break;
        case 4:
            System.out.println("TXT was created");break;
    }
}

```