

Caso 1

- A- Naturaleza del caso: la función hace dos cosas siendo una sola función
- B- Lo que hace la función o método: saca el dígito del centro de un número de dígitos impar
- C- Código no limpio

```
1 def centronum(num):
2     num=abs(num)
3     copianum = num
4     if type(num)==int:
5         print("Ingrese un numero entero")
6     elif type(num)==int:
7         largo=0
8         while copianum!=0:
9             copianum=copianum//10
10            largo=largo+1
11            largo=(largo//2)-1
12            if largo//2 != 0:
13                while largo>=0:
14                    num=num//10
15                    centro=num%10
16                    largo=largo-1
17                return centro
18            else:
19                print("Ingrese un numero de digitos impar")
20        else:
21            print("Ingrese un numero entero")
```

- D- Código limpio

```
1 def largo(num):
2     num=abs(num)
3     copianum=num
4     if isinstance(copianum,int):
5         digitos=0
6         while copianum!=0:
7             copianum=copianum//10
8             digitos=digitos+1
9         return digitos
10    else:
11        return "no entero"
12
13 def centernum(num):
14     num=abs(num)
15     largo=largo(num)
16     if type(num)==int and largo==1:
17         print("Ingrese un numero entero de digitos impar")
18     elif type(num)==int and largo%2!=0:
19         largo=(largo//2)-1
20         while largo>=0:
21             num=num//10
22             centro=num%10
23             largo=largo-1
24         return centro
25     else:
26         print("Ingrese un numero entero de digitos impar")
```

Caso 2

- A- Naturaleza del caso: la función utiliza nombres poco descriptivos en sus variables
- B- Lo que hace la función o método: divide un numero en dígitos y estos los ingresa en una lista
- C- Código no limpio

```
1 def paslist (n):  
2     if n==0:  
3         return([0])  
4     else:  
5         nlist=[]  
6         while n!=0:  
7             d=n%10  
8             nlist= [d]+nlist  
9             n=n//10  
10        return nlist
```

- D- Código limpio

```
1 def pasarDigitosDeNumeroALista (numero):  
2     if numero==0:  
3         return([0])  
4     else:  
5         listaDeDigitos=[]  
6         while numero!=0:  
7             digitoAIngresar = numero % 10  
8             listaDeDigitos = [digitoAIngresar] + listaDeDigitos  
9             numero = numero // 10  
10        return listaDeDigitos
```

Caso 3

- A- Naturaleza del caso: la función no realiza la validación de la lista y al ser en el lenguaje Python podría generar un error ya que no hay que declarar el tipo de dato
- B- Lo que hace la función o método: invierte el orden de los elementos de una lista
- C- Código no limpio

```
1 def invertirLista(lista):
2     if lista == []:
3         return []
4     else:
5         nuevaLista = []
6         for elemento in lista:
7             nuevaLista = [elemento] + nuevaLista
8     return nuevaLista
```

- D- Código limpio

```
1 def invertirLista(lista):
2     if type(lista) == list:
3         if lista == []:
4             return []
5         else:
6             nuevaLista = []
7             for elemento in lista:
8                 nuevaLista = [elemento] + nuevaLista
9             return nuevaLista
10    else:
11        print("Ingrese una lista")
```