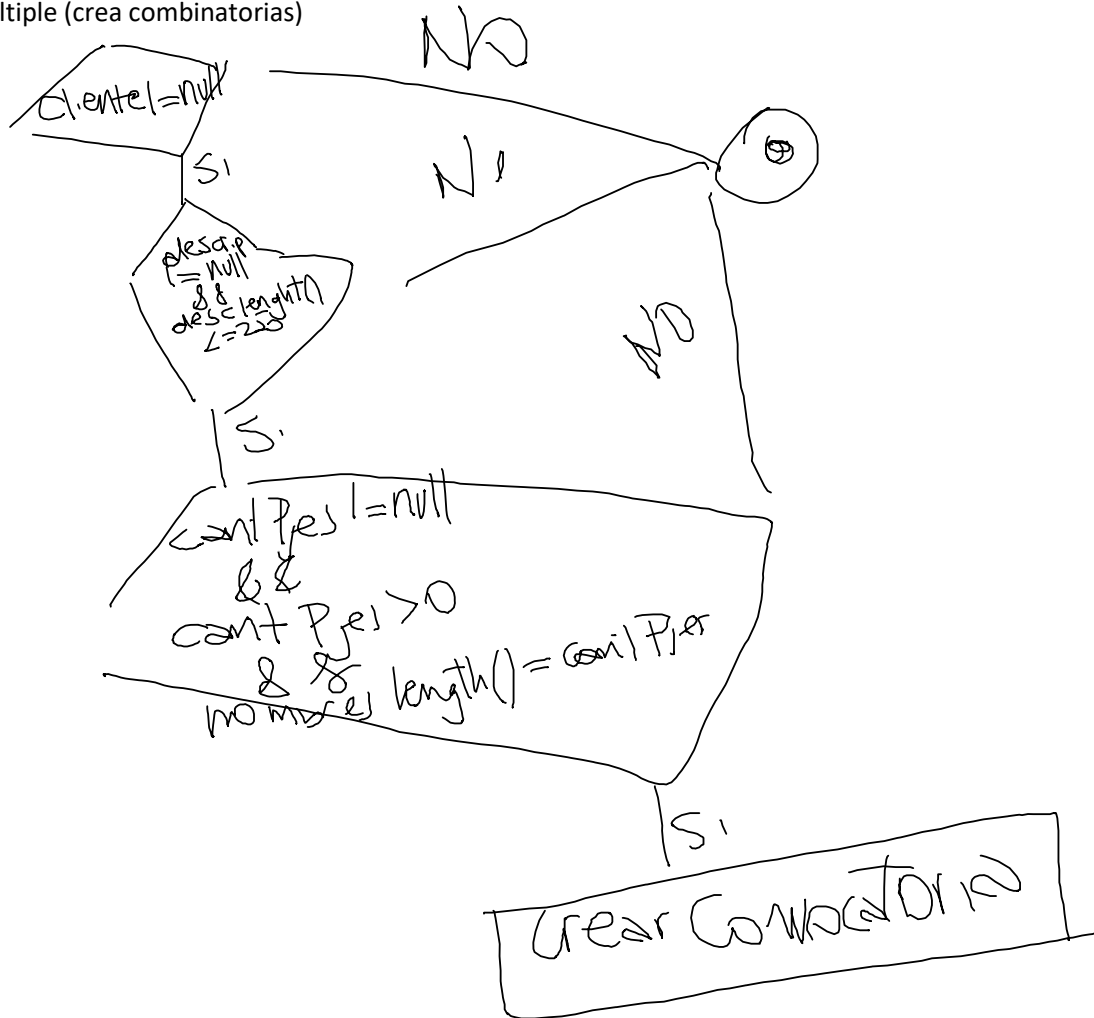


Métodos:

- Sentencias (aquel que no termina en sentencia no nos interesa)
- Decisión: probar las salidas V o F all menos. Condicionales, switch case.
- Condición: Probar cada salida de una condición
- Decisión/Condición (ambas condiciones de cumplen o no)
- Multiple (crea combinatorias)



Por sentencias: 1

Por decisión: Mínimo CP=4

Por condición: 6

Por decisión/condicion: 6

Por multiple: 7

Sentenzier

①

```
cliente = "Juan"
descripcion = x → es var string de 250 caract.
countPersonajes = 1
nombre = [Jack Sparrow]
```

Precondiciones:

① los clientes deben estar registrados

Decision

1) Client = Fran,
description = 'Pet:ivlon',
cunt + Personaje = 1,
nombre = [Voldemort]

2) Idem 1), nombre = []

3) Cliente = Fran,
description = null

4) Cliente = null.

Precondición: Fran es un cliente ya registrado

Cond.

1) Cliente = null

2) Cliente = "PePito"
descripcion = "desc"
cantPersonajes = 2

Nombres = ["ZUKE", "LEIA"]

3) Cliente = "PePito"
descripcion = null

4) Cliente: "Pepe To"
 Descripción: "desc" * TIENE 251 CARACTERES

5) Cliente = "pepito"
description = "desc"

canPersonajes = -1
 nombres = ["LUKE"]

6) cliente = "DEPITO"
descripcion = "DESC"
cantPersonajes = nu

Conda

1) Cliente = null

2) Cliente = "PePito"
 descripcion = "desc"
 cantPersonajes = 2
 nombres = ["LUKE", "LEIA"]

3) Cliente = "PePire"
descripcion = null

4) Cliente: "pepito"
 descripcion: "desc" * 251
 Colores

5) cliente: "pepito"
la creación: "desc"

canPersonajes = -1
ambres = "LUKE"

6) cliente = "DEPITO"

countPersonajes = 0

Multiple

① Cliente = "Pepe"
Desc = "asd."
ant per = 1
membres = {Carlos}

② client: rep^a, Desc: "cst"
Contpos: 1, nombres: [.]

③ cliente: "Pope", Desc: "asd"
camper: 0 nombres: []

④ check: "pepe", desc: "asd"
cant pers: null

⑤ cherries: "pepe", dark red color

② classe: null, desc: null,

8 cliente = "Pepe", car

