

## BIZCOCHUELOS – LA FUNCIONAL DEL OESTE

---

### PRIMERA PARTE (YA HECHA)

---

"La Funcional del Oeste" es una conocida repostería de Merlo que se especializa en la venta de tortas caseras hechas a pedido. Un factor clave el éxito que las mismas tienen desde hace varios años son sus excelsos bizcochuelos, de diversos sabores y consistencias, y que los clientes pueden elegir al momento de ordenar. Para obtener distintos tipos de bizcochuelos, los maestros panaderos, basándose en una receta tradicional, modifican ligeramente las proporciones de los ingredientes. Las recetas de bizcochuelos se representan como 5-uplas, de la forma: `(Int, Int, Int, Int, Int)` que representan: `(cucharadasDeHarina, cucharadasDeAzucar, cucharadasDeLeche, cucharadasDeAceite, cantidadDeHuevos)`

Se sabe que la receta tradicional es:

```
recetaTradicional = (20,20,20,20,2)
```

Y que la receta promo es:

```
recetaPromo = (10,10,10,2,5)
```

Los clientes de la repostería suelen pasar por su local a lo largo del día y dejar una orden que retiran horas más tarde, especificando cual tipo de bizcochuelo quieren y en qué cantidad. Ocasionalmente, puede suceder que después de hacerla la cancelen un rato más tarde. Las órdenes se representan de la siguiente forma: `(String, Receta, Int, Bool)` que representan: `(nombreCliente, receta, cantidad, fueCancelada)`. Por ejemplo:

```
ordenPepe = ("pepe", recetaTradicional, 2, False)
ordenPepe2= ("pepe", recetaPromo, 20, True)
ordenPepe3= ("pepe", (15, 5, 25, 9, 8), 4, False)
ordenPepe4= ("pepe", recetaPromo, 3, False)
ordenRoque= ("roque", (0, 1, 1, 2, 3), 5, True)
```

### SEGUNDA PARTE

---

Los clientes son gente que puede ser más o menos impaciente para retirar las tortas, por lo que es un tema a considerar. Los clientes se modelan de la siguiente forma: `(nombre, criterioDePaciencia)`, donde `criterioDePaciencia` puede ser:

- **constante:** espera siempre hasta 8 horas sin importar las órdenes que haya hecho
- **lineal:** espera una cantidad en horas propia de su paciencia lineal más 1 hora por cada torta pedida.
- **mucha:** si encargó una ó 2 tortas, espera 2 horas. Si no, 3 horas por la cantidad de tortas encargadas

Por ejemplo:

```
pepe= ("pepe", constante)
guille= ("guille", mucha)
roque= ("roque", (lineal 5))
laura= ("laura", (lineal 1))
```

## TRABAJO PRACTICO – PARADIGMA FUNCIONAL – 2da ENTREGA

Independientemente de la paciencia de los clientes, lo cierto es que el tiempo que toma una orden en ser entregada no depende de la paciencia del cliente, sino de cuántas ordenes haya pendientes: se tarda 2 horas por torta si tienen poco trabajo, o 3 horas por torta en caso contrario (ver más adelante la definición de “poco trabajo”)

Las órdenes de la pastelería las encontramos en una lista de órdenes:

```
ordenesPasteleria=[ordenPepe, ordenRoque, ordenPepe2, ordenPepe3, ordenPepe4]
```

1- Codificar los criterios de paciencia. Algunos ejemplos de uso son:

```
> constante [ordenPepe1, ordenPepe2, ordenPepe3, ordenPepe4]
8
> lineal 2 [ordenPepe1, ordenPepe2, ordenPepe3, ordenPepe4]
6
> mucha [ordenPepe1, ordenPepe2, ordenPepe3, ordenPepe4]
12
```

2- Programar una función que reciba una dupla (cliente, criterio) y calcule el tiempo máximo que la persona va a esperar, antes que se enoje y deje sus pedidos. Los pedidos de una persona se pueden obtener de `ordenesPasteleria`.

```
> cuantoPuedoTardar pepe
8
> cuantoPuedoTardar laura
1
> cuantoPuedoTardar roque
6
> cuantoPuedoTardar guille
2
```

3- Saber si la pastelería tiene poco trabajo. Tener en cuenta que tiene poco trabajo si hay menos de 10 órdenes (no canceladas) o si todas las órdenes salen bien. Una orden sale bien cuando la receta para su bizcochuelo sale bien.

4- Dado un ingrediente (función de acceso a la tupla), saber cuánto se necesita de ese ingrediente para cubrir la totalidad de órdenes pendientes (no canceladas).

```
> cuantoSeNecesita harina
130 (20*2+15*4+10*3)
> cuantoSeNecesita aceite
82 (20*2+9*4+2*3)
```