



Maestros PdePizzeros

Nos pidieron escribir un programa que permita analizar y predecir los resultados de un concurso de pizzeros.

Cada participante del concurso tiene que preparar una pizza que será evaluada por un distinguido jurado. Aquel participante que reciba un mejor puntaje, gana.

La manera en la que cada pizzero prepara una pizza es parecida:

- Comienzan preparando una prepizza, que es una pizza de 500g, con tomate como único ingrediente y cuyo grado de cocción es 0.3.
- Le agrega el ingrediente muzzarella.
- Le dan su toque, que es algo que depende de cada pizzero.
- La cocinan por 10 minutos. Por cada minuto de cocción, la pizza aumenta su grado de cocción en $(10/\text{masa de la pizza al comenzar a cocinarla})$ y también pierde un 1% de su masa **inicial** (con la que comenzó “de fábrica”).

Por ejemplo, si se cocina una de 500g que tenía 500g inicialmente por 10 minutos, termina con 450g y su grado de cocción aumenta en 0.2.

Queremos representar a los siguientes pizzeros:

- Carla, cuyo toque es agregarle provolone y luego cocinarla durante 1 minuto.
- Facu, cuyo toque depende de el humor con el que se encuentre:
 - Si ese día se siente arriesgado, es agregar ananá.
 - Si ese día se siente conservador, es agregar orégano.
- los pizzeros de la pizzería “La Marítima”, cuyo toque es agregar su ingrediente favorito (cada pizzero tiene el suyo) y también agregarle el ingrediente del día del local (que se quiere poder cambiar día a día y es el mismo para todos). Además, en La Marítima los pizzeros hacen la prepizza de 650g en vez de 500g.

Hay ciertas características de la pizza que pueden importar al jurado, por ejemplo:

- Que no esté cruda ni quemada. Una pizza está cruda si su grado de cocción es menor a 0.4 y está quemada si su grado de cocción es mayor a 1.0.
- Que tenga cierto ingrediente.
- Que esté cargada, es decir, que tenga más de una vez el mismo ingrediente

El jurado está formado por un grupo de críticos. Todo crítico veta pizzas que estén crudas o que tengan un ingrediente que no les gusta.

Algunos ejemplos de jurados son:

- **Zeff:** no le gusta el atún, y el puntaje que le pone a la pizza se calcula como la masa final de la pizza sobre la masa inicial con la que se hizo.
- **Anton:** no le gustan los morrones. Además de vetar las pizzas que suelen vetar los críticos, también veta pizzas con menos de 400g de masa. Su forma de puntuar pizzas es un poco más particular: Parte de un puntaje perfecto (1.0) y a eso **le resta** 0.5 si la pizza está quemada, **le suma** 0.2 si la pizza está cargada, y **le resta** 0.1 por cada 100 gramos por encima de 500g que tenga la pizza.
- **Contrera:** no le gusta el queso azul, y su ingrediente favorito es el ananá. Este crítico siempre toma a otro crítico de referencia y le lleva la contra. Su puntaje será:

1 - el puntaje del otro crítico + 0.2 si la pizza tiene su ingrediente favorito.

Luego, cada uno tiene un criterio según el cual puntúa la pizza. **Nota:** el puntaje del criterio siempre tiene como máximo 1.0, si la fórmula aplicada a una pizza pudiese superar ese valor, debe limitarse a 1.0 de todas formas. De la misma manera, nunca puede bajar de 0.0.

Requerimientos:

1. Queremos poder pedir a un pizzero que cocine una pizza.
2. Sobre una pizza, queremos poder saber:

1. Si está cruda.
2. Si está quemada.
3. Si está cargada.

3. Jurados

1. Queremos poder preguntarle a un jurado si vetaría cierta pizza.
2. Queremos poder pedirle a un jurado el puntaje para cierta pizza.

4. Concurso

1. En un concurso pueden participar varios pizzeros y son evaluados por un equipo de críticos.

Dar un ejemplo de cómo se crearía un concurso en el que participan:

- como pizzeros: Facu, Carla y 2 pizzeros de La Marítima (de uno de ellos su ingrediente favorito es el atún y de otro los morrones).
- como jurados: Zeff y Contrera (que está tomando como referencia a Zeff).

1. La forma en la que funciona el concurso es la siguiente:

- Cada pizzero prepara una pizza.
- Se hace una ronda de vetos donde los críticos eliminan a todos los pizzeros cuyas pizzas fueron vetadas.

- En la ronda final, se comparan las pizzas comparando la sumatoria de puntajes que le dieron los críticos. Aquel pizzero cuya pizza sea la mejor puntuada gana el concurso.
1. Queremos, dado un concurso, obtener al pizzero ganador. Siempre tiene que ganar uno sí o sí, por lo que si llegase a haber empate, o si todos los pizzeros fueron vetados, no hay ganador posible y queremos informar esto de alguna manera.

Por ejemplo, en el concurso de ejemplo del punto a:

- *Gana Facu si se siente arriesgado y el ingrediente del día en La Marítima son las aceitunas.*
 - *Gana el 2° pizzero de La Marítima si Facu se siente conservador y el ingrediente del día en La Marítima es el ananá.*
 - *No hay ganador posible si Facu se siente conservador y el ingrediente del día en La Marítima es el atún.*
-

```
class Pizzero{
    var pizza = new Pizza()

    method verPizza() = pizza

    method crearPizza(){
        pizza = new Pizza()
    }

    method agregarMuzzarella(){
        pizza.agregarIngrediente(muzzarella)
    }

    method agregarToque()

    method cocinarPizza(minutos){
        pizza.aumentarGradoDeCoccion(minutos)
        pizza.perderMasa(minutos)
    }
}
```

```

        method hacerPizza(minutos){
            self.crearPizza()
            self.agregarMuzzarella()
            self.agregarToque()
            self.cocinarPizza(minutos)
            return pizza
        }

    }

    object carla inherits Pizzero{
        override method agregarToque() = pizza.agregarIngrediente(provolone)
    }

    object facu inherits Pizzero{
        var tipoDeDia = diaConservador
        method seSienteArriegado() = tipoDeDia == diaArriesgado
        method cambiarDia(nuevoDia) {
            tipoDeDia = nuevoDia
        }

        override method agregarToque(){
            if(self.seSienteArriegado()){
                pizza.agregarIngrediente(anana)
            } else {
                pizza.agregarIngrediente(oregano)
            }
        }
    }

    class PizzerosLaMaritima inherits Pizzero {
        const ingredienteFavorito
        var ingredienteDelDia
        override method crearPizza(){
            pizza = new Pizza(masa = 650)

```

```

    }

    override method agregarToque() {
        pizza.agregarIngrediente(ingredienteFavorito)
        pizza.agregarIngrediente(ingredienteDelDia)
    }
}

```

```

class Pizza{
    var masa = 500
    const masalInicial = masa
    var ingredientes = [tomate]
    var gradoDeCoccion = 0.3
    var puntaje = 0
    var estaVetada = false

    method ingredientes() = ingredientes
    method masa() = masa
    method masalInicial() = masalInicial
    method puntaje() = puntaje

    method agregarIngrediente(ingrediente) = ingredientes.add(ingrediente)
    method aumentarGradoDeCoccion(minutos){ gradoDeCoccion += (10 / masalInicial)*
minutos }
    method perderMasa(minutos){ masa -= (masalInicial * minutos)/ 100 }

    method estaCruda() = gradoDeCoccion < 0.4
    method estaQuemada() = gradoDeCoccion > 1.0
    method estaCargada() = ingredientes.asSet().size() < ingredientes.size()
}

```

```

method cambiarVetadoDePizza() {
    estaVetada = true
}
method vetadoDePizza() = estaVetada

method conseguirPuntaje(jurado){
    puntaje += jurado.puntaje(self)
}
}

```

```

class Jurado{
    var puntajeTemporal = 0
    method pizzaContiene(pizza, ingrediente) = pizza.ingredientes().contains(ingrediente)
    method vetar(pizza) = pizza.estaCruda()
    method puntaje(pizza)
    method pizzaEstaVetada(pizza) {
        if(self.vetar(pizza)){
            pizza.cambiarVetadoDePizza()
        }
    }
}

```

```

object zeff inherits Jurado{
    const ingredienteQueNoLeGusta = atun

    override method vetar(pizza) = super(pizza) || self.pizzaContiene(pizza,
ingredienteQueNoLeGusta)

    override method puntaje(pizza){
        puntajeTemporal = pizza.masa() / pizza.masalInicial()
        if(puntajeTemporal > 1.0){

```

```

        return 1.0
    } else if (puntajeTemporal < 0.0){
        return 0.0
    } else {
        return puntajeTemporal
    }
}

}

object contrera inherits Jurado{

    const ingredienteQueNoLeGusta = quesoAzul

    const ingredienteFavorito = anana

    var juradoDeReferencia = zeff


    override method vetar(pizza) = super(pizza) || self.pizzaContiene(pizza,
ingredienteQueNoLeGusta)

    override method puntaje(pizza) {
        if(self.pizzaContiene(pizza, ingredienteFavorito)){
            puntajeTemporal = 1 - juradoDeReferencia.puntaje(pizza) + 0.2
        } else {
            puntajeTemporal = 1 - juradoDeReferencia.puntaje(pizza)
        }
        if(puntajeTemporal > 1.0){
            return 1.0
        } else if (puntajeTemporal < 0.0){
            return 0.0
        } else {
            return puntajeTemporal
        }
    }

}

}

```

```

class Concurso {
    var listaDePizzeros = []
    const listaDeJurados = []

    method verPizzeros() = listaDePizzeros
    method verJurado() = listaDeJurados

    method todosLosPizzerosHacenPizza(minutos){
        listaDePizzeros.forEach({ pizzero => pizzero.hacerPizza(minutos) })
    }

    method seVetanPizzas(){
        listaDePizzeros.forEach({ pizzero => listaDeJurados.forEach({ jurado =>
jurado.pizzaEstaVetada(pizzero.verPizza())}}))
        listaDePizzeros.removeAllSuchThat({ pizzero =>
pizzero.verPizza().vetadoDePizza()})
    }

    method lasPizzasConsiguenSuPuntaje(){
        listaDePizzeros.forEach({ pizzero => listaDeJurados.forEach({ jurado =>
pizzero.verPizza().conseguirPuntaje(jurado)}}))
    }

    method hacerRondaFinal(){
        listaDePizzeros.max({pizzero => pizzero.verPizza().puntaje()})
    }

    method ganadorEs() {
        if((listaDePizzeros.get(0).verPizza().puntaje() ==
listaDePizzeros.get(1).verPizza().puntaje()) || listaDePizzeros.isEmpty()){
            throw new UserException(message = "No hay ganador")
        } else {
            return listaDePizzeros.first()
        }
    }
}

```



```
    }  
    method simularConcurso(minutos){  
        self.todosLosPizzerosHacenPizza(minutos)  
        self.seVetanPizzas()  
        self.lasPizzasConsiguenSuPuntaje()  
        self.hacerRondaFinal()  
        return self.ganadorEs()  
    }  
}
```

```
object diaArriesgado{}  
object diaConservador{}
```

```
object tomate {}  
object muzzarella {}  
object provolone {}  
object anana {}  
object oregano {}  
object atun {}  
object morron {}  
object quesoAzul {}  
object aceitunas {}
```

```
class UserException inherits Exception { }
```