

# Night Club 2 - Functional Fest<sup>1</sup>



## Indice

### [1 El dominio](#)

### [2 Primer entrega](#)

#### [2.1 Temas a evaluar](#)

#### [2.2 Punto 1](#)

#### [2.3 Punto 2](#)

#### [2.4 Punto 3](#)

#### [2.5 Punto 5](#)

#### [2.6 Punto 6](#)

#### [2.7 Punto 7](#)

### [3 Casos de prueba](#)

#### [3.1 Punto 3](#)

#### [3.2 Punto 4](#)

#### [3.3 Punto 5](#)

#### [3.4 Punto 6](#)

#### [3.5 Punto 7](#)

### [4 Importante](#)

#### [4.1 Restricciones sobre la entrega](#)

#### [4.2 Extras que necesitamos antes de la entrega](#)

---

<sup>1</sup> El de la foto es el señor Haskell Curry: [https://es.wikipedia.org/wiki/Haskell\\_Curry](https://es.wikipedia.org/wiki/Haskell_Curry)

# 1 El dominio

Un amigo nuestro es encargado de un boliche, necesita modelar su negocio para mejorar la forma en que lo administra. En el relevamiento surgieron las siguientes entidades:

- Los clientes son personas mayores de edad. Al ingresar se registra su nombre, la resistencia, y los otros clientes a quienes considera amigos.
- Los clientes van teniendo nuevos amigos. Ningún cliente puede agregar a sí mismo como amigo y no tiene sentido volver a hacerse amigo de alguien a quien ya considera amigo.
- Cuando un cliente toma ciertas bebidas, suele verse afectado de forma diferente:
  - el grog XD deja sin resistencia al cliente.
  - la jarra loca baja 10 unidades de resistencia al cliente y a todos sus amigos.
  - el Klusener tiene diferentes gustos, como "Huevo", "Chocolate" y "Frutilla", entre otros, y disminuyen tanta resistencia como la cantidad de letras que tenga el gusto. Ej: Alguien toma Klusener de Huevo, eso le baja la resistencia 5 unidades.
  - el Tintico (café colombiano) le sube 5 puntos de resistencia al cliente por cada amigo que tenga.
  - la Soda, que dependiendo de la fuerza agrega "erp" al inicio del nombre del cliente, con tantas r como indique la fuerza. Por ejemplo, si Rodri se toma una soda con fuerza 2 y luego una soda con fuerza 5, su nombre queda "errrrrperrpRodri" (y queda con la misma resistencia y amigos)<sup>2</sup>.
- Pero como no es todo jarana, los clientes pueden decidir "rescatarse" un cierto tiempo medido en horas, que siempre es mayor a 0. Eso le devuelve 200 de resistencia si son más de 3 horas o 100 en caso contrario.

## 2 Primer entrega

### 2.1 Temas a evaluar

- Modelado de información
- Composición
- Aplicación parcial
- Definición con Guardas

### 2.2 Punto 1

Modelar el tipo de dato cliente. Justificar el criterio utilizado.

---

<sup>2</sup> Puede serles útil la función replicate

## 2.3 Punto 2

- Modelar a Rodri, que tiene 55 de resistencia y no considera a nadie como amigo.
- También modelar a Marcos, un cliente que tiene resistencia 40 y considera a Rodri como su único amigo.
- Tenemos a Cristian, un cliente cuya resistencia es 2, y no considera a nadie como amigo.
- Por último está Ana, una cliente que tiene 120 de resistencia y considera a Marcos y a Rodri como amigos.

## 2.4 Punto 3

Desarrollar la función **comoEsta** que dice cómo está un cliente

- Si su resistencia es mayor a 50, está “fresco”
- Si no está “fresco”, pero tiene más de un amigo, está “piola”
- En caso contrario, está “duro”

## Punto 4

Hacer que un cliente reconozca como amigo a otro cliente, respetando las restricciones definidas por el negocio (no agregar más de una vez al mismo amigo ni agregarse a sí mismo como amigo, basándose en que dos clientes son iguales si tienen el mismo nombre).

## 2.5 Punto 5

Representar con la abstracción que crea conveniente

- a cada una de las bebidas mencionadas
- y cómo queda un cliente luego de tomar cualquiera de las bebidas mencionadas.

## 2.6 Punto 6

Hacer que un cliente pueda rescatarse<sup>3</sup>.

## 2.7 Punto 7

Escribir la *consulta en la consola* que permita realizar el siguiente itinerario con Ana: tomarse una jarra loca, un klusener de chocolate, rescatarse 2 horas y luego tomar un klusener de huevo.

---

<sup>3</sup> Si ve que hay ideas que se repiten, sería bueno encontrar abstracciones que eviten esa redundancia.

## 3 Casos de prueba

### 3.1 Punto 3

- Cristian debe estar "duro"
- Rodri debe estar "fresco"
- Marcos debe estar "duro"
- Si Marcos se hace amigo de Ana y Rodri, está "piola"

### 3.2 Punto 4

- Intentar agregar a Rodri como amigo de Rodri. ¿Qué debe pasar?
- Hacer que Marcos reconozca a Rodri como amigo (que ya lo conoce). ¿Qué debe pasar?
- Hacer que Rodri reconozca a Marcos como amigo. Debe arrancar con 0 amigos y luego agregarlo a Rodri como único amigo.

### 3.3 Punto 5

- Ana toma GrogXD. Queda con resistencia 0.
- Ana toma la Jarra Loca. Queda con resistencia 110, su amigo Marcos queda con 30 de resistencia y su amigo Rodri queda con 45 de resistencia.
- Ana toma un Klusener de huevo, queda con 115 de resistencia
- Ana toma un Klusener de chocolate, queda con 111 de resistencia
- Cristian toma un Tintico, queda con 2 de resistencia por no tener amigos.
- Ana toma un Tintico, pasa de 120 a 130 de resistencia (tiene 2 amigos)
- Rodri toma una Soda de fuerza 2, queda con nombre "errpRodri"
- Ana toma una Soda de fuerza 10, queda con nombre "errrrrrrrrpAna"
- Ana toma una Soda de fuerza 0, queda con nombre "epAna"

### 3.4 Punto 6

- Rodri se rescata 5 horas, queda con 255 de resistencia ( $55 + 200$  ya que son más de 3 horas)
- Rodri se rescata 1 hora, queda con 155 de resistencia ( $55 + 100$  porque son menos de 3 horas)

### 3.5 Punto 7

Luego de evaluar el itinerario de Ana, queda con 196 de resistencia, como amigos a Marcos (30 de resistencia) y Rodri (45 de resistencia).

## 4 Importante

Lo que sigue es importante para los cursos que vayan a validar la entrega mediante tests unitarios automatizados, en particular

- Miércoles a la noche, cursos K2052 | K2152

### 4.1 Restricciones sobre la entrega

Definir la abstracción Cliente como quieran. Solo debe cumplir:

- que debe haber una función **nombre** que permita conocer el nombre de un cliente
- que debe haber una función **resistencia** que permita conocer la resistencia de un cliente
- que debe haber una función **amigos** que permita conocer los amigos de un cliente
- el nombre de Rodri debe ser "Rodri", el de Marcos "Marcos" y el de Ana "Ana" respetando mayúsculas y minúsculas
- las expresiones que se esperan: rodri, marcos, ana y cristian

### 4.2 Extras que necesitamos antes de la entrega

El archivo .hs se puede llamar como ustedes quieran, solo que la primera línea debe ser la siguiente definición:

`module TP where`

(luego escriben todas las funciones propias del TP)

También necesitamos que descarguen [Haskell Platform](#), que incluye Cabal, una herramienta de manejo de dependencias de Haskell y , y que instalen [Hspec](#), con los que vamos a correr las pruebas unitarias automatizadas de tu TP Eso se hace

1. Abrir la consola del sistema operativo.
2. Ejecutar el comando: `cabal update && cabal install hspec.`