

Functional Master Series



En base a la creciente popularidad del rap y las batallas de freestyle en Argentina durante los últimos años, los máximos referentes de la movida nos pidieron ayuda para evaluar sus puestas en escena. Es por esto que nos estuvieron instruyendo en esto de las batallas.

Nos contaron que lo que hacen los artistas es jugar con las palabras armando *versos*, que son frases con sentido, y esos versos se agrupan en *estrofas*, generalmente de a 4 pero puede variar. Las estrofas expresan alguna idea, agite para el público, o respuesta que el artista quiere transmitir.

A partir de este primer relevamiento el equipo de desarrollo decidió tener las siguientes abstracciones a partir de las cuales debemos desarrollar nuestro programa:

```
type Palabra = String
type Verso = String
type Estrofa = [Verso]
type Artista = String -- Solamente interesa el nombre

esVocal :: Char -> Bool
esVocal = flip elem "aeiou"

tieneTilde :: Char -> Bool
tieneTilde = flip elem "áéíóú"

cumplen :: (a -> b) -> (b -> b -> Bool) -> a -> a -> Bool
cumplen f comp v1 v2 = comp (f v1) (f v2)
```

Requerimientos

Se pide resolver los siguientes requerimientos en un programa funcional, aprovechando las ideas del paradigma (orden superior, aplicación parcial, composición, etc) y manteniendo buena calidad de código (ser declarativo, no repetir ideas, nombres expresivos, etc).

Jugando con las palabras

Lo primero que nos dijeron es entender cómo es el juego de palabras que comúnmente se ven en las batallas.¹

¹ Cabe destacar que las definiciones que damos en este enunciado no son del todo correctas, es un modelo simplificado de las verdaderas [figuras literarias](#).

Rimas

Existen varias formas de clasificar las rimas, la más sencillas son:

- Rima asonante: se cumple cuando las dos últimas vocales de la palabra coinciden. *Por ejemplo: parcial - estirar*
- Rima consonante: se cumple cuando las tres últimas letras de la palabra coinciden. *Por ejemplo: función - canción*

1. Se pide

- a. Determinar si dos palabras riman. Es decir, si generan una rima, ya sea asonante o consonante, pero teniendo en cuenta que dos palabras iguales no se consideran una rima.
- b. Enumerar todos los casos de test representativos (clases de equivalencia) de la función anterior. No hace falta escribir los tests (serían sólo sus nombres).

Conjugaciones

Pero solamente rimar dos palabras sueltas no basta, los artistas tienen que armar versos completos y conjugarlos de manera que queden agradables al oírlo. Existen varias conjugaciones de versos posibles, por ahora nos dijeron estas:

- Por medio de rimas: dos versos se conjugan con rima cuando logran rimar las últimas palabras de cada uno. *Por ejemplo:*

*no hace falta un programa que genere una **canción**
para saber que esto se resuelve con una **función***

- Haciendo anadiplosis: sucede cuando el segundo verso comienza con la misma palabra con la que termina el primero. *Por ejemplo:*

*este examen no se aprueba sin aplicación **parcial**
parcial lindo y divertido si rendiste todas las katas*

- ### 2. Modelar las conjugaciones anteriores. Tener en cuenta que debe ser sencillo agregar más conjugaciones al sistema, ya que se planea hacerlo en próximas iteraciones.

Tip: Haskell tiene la función `words` que dado un `string` lo separa por espacios. Ejemplo:

> words "hola soy pepita"

["hola", "soy", "pepita"]

Patrones

Poder jugar con las palabras y versos de manera agradable es solo el primer paso. Los artistas llevan eso a otro nivel al armar patrones en cada estrofa. Un patrón es una forma de articular los versos dentro de una estrofa y existen muchas formas posibles de hacer esto, describiremos las siguientes:

- Simple: es un patrón en el que riman 2 versos, especificados por su posición en la estrofa.

Por ejemplo, la siguiente estrofa tiene un patrón simple de 1 y 4, pero no de 1 y 3:

- (1) esta rima es fácil como patear un **penal**
- (2) solamente tiene como objetivo servir de ejemplo
- (3) los versos del medio son medio fríos
- (4) porque el remate se retoma al **final**

- Esdrújulas: Todos los versos terminan con palabras en esdrújula. Diremos que una palabra es esdrújula cuando la antepenúltima vocal está acentuada. *Un ejemplo de este patrón sería:*

a ponerse los guantes y subir al cuadrilátero
que después de este parcial acerca el paradigma lógico
no entiendo por qué está fallando mi código
si todas estas frases terminan en esdrújulas

- Anáfora: Todos los versos comienzan con la misma palabra. *Por ejemplo:*

paradigmas hay varios, recién vamos por funcional
paradigmas de programación es lo que analizamos acá
paradigmas que te invitan a otras formas de razonar
paradigmas es la materia que más me gusta cursar

- Cadena: Es un patrón que se crea al conjugar cada verso con el siguiente, usando siempre la misma conjugación. La conjugación usada es elegida por el artista mientras está rapeando. *Por ejemplo, una cadena de anadiplosis sería:*

este es un ejemplo de un parcial **compuesto**
compuesto de funciones que se operan entre **ellas**
ellas también pueden pasarse por **parámetro**
parámetro que recibe otra función de alto orden

Tip: puede hacerse utilizando recursividad.

- CombinaDos: Dos patrones cualesquiera se pueden combinar para crear un patrón más complejo, y decimos que una estrofa lo cumple cuando cumple ambos patrones a la vez. *Por ejemplo, si contemplamos el patrón combinado de esdrújulas y anáfora, una estrofa que cumpliría podría ser:*

estrofa que sirve como caso ejémplico
estrofa dedicada a la gente fanática
estrofa comenzada toda con anáfora
estrofa que termina siempre con esdrújulas

3. Se pide

- a. Modelar los patrones anteriores.
- b. Usar el punto a para definir los siguientes patrones combinados:
 - aabb = patrón simple entre 1 y 2 + otro simple entre 3 y 4
 - abab = patrón simple entre 1 y 3 + otro simple entre 2 y 4

- abba = patrón simple entre 1 y 4 + otro simple entre 2 y 3
 - hardcore = patrón de cadena de rimas + esdrújulas
- c. ¿Se podría saber si una estrofa con infinitos versos cumple con el patrón *hardcore*? ¿Y el *aabb*? Justifique en cada caso específicamente por qué (no valen respuestas genéricas).

Puestas en escena

¡Genial! La gente ya comenzó a usar nuestro sistema en las batallas y dicen que hay que hacerle unos ajustes. Nos comentaron que lo que hicimos hasta ahora es importante pero las estrofas no lo son todo en una batalla, ya que cuando los artistas “tiran freestyle” entran más elementos en juego.

Por ahora pudimos identificar las siguientes variables significativas de una puesta en escena: si el público está exaltado o no, la potencia (un número), además de, claro está, la estrofa del freestyle (una sola, la puesta es por estrofa) y el artista.

Además nos dimos cuenta que en cada puesta en escena cada artista utiliza un estilo distinto, dependiendo del mensaje que quiere transmitir, que altera la puesta en escena original. Identificamos los siguientes casos:

- Gritar: aumenta la potencia en un 50%
- Responder un acote: conociendo su efectividad, aumenta la potencia en un 20%, y además el público queda exaltado si la respuesta fue efectiva, sino no lo queda.
- Tirar técnicas: se refiere a cuando el artista deja en evidencia que puede lograr algún patrón en particular, aumenta la potencia en un 10%, además el público se exalta si la estrofa cumple con dicho patrón, sino no.

4. Hacer que un artista se tire un freestyle a partir de la estrofa que quiere decir y el estilo que le quiera dar a su puesta en escena. Para ello se parte siempre de una puesta base que tiene potencia 1 y el público tranquilo, la que luego varía según el estilo utilizado. El resultado de que un artista se tire un freestyle es una puesta en escena.

Jurados

Nuestro sistema llegó a las manos del jurado de las batallas, generalmente personas con trayectoria en el ambiente, son los que puntúan las puestas en escena de los artistas y al final eligen al ganador. Nos pidieron agregar configuraciones extras a nuestro sistema para ayudarles con su tarea.

Nos contaron que cada jurado define sus propios criterios para puntuar una puesta en escena, estos criterios se basan en alguna condición que debe cumplir la puesta y un puntaje que se le otorga si la cumple. Para que nos demos una idea, nos comentaron algunos ejemplos de estos criterios: que el freestyle cumpla con un patrón determinado, si el público está exaltado, si la potencia cumple algún requisito, etc.

Es importante entender que cada jurado define sus propias condiciones y puntajes para armar los criterios que le parecen decisivos.

5. Se pide
- a. Definir el jurado *a/Toke* con los siguientes criterios:

- Si el freestyle (estrofa de la puesta) cumple con el patrón aabb, entonces suma 0.5 punto
 - Si el freestyle (estrofa de la puesta) cumple con el patrón combinado de esdrújulas y simple entre 1 y 4, entonces suma 1 punto
 - Si el público está exaltado, entonces suma 1 punto
 - Si tiene una potencia mayor a 1.5, entonces suma 2 puntos
- b. Calcular el puntaje que le daría un jurado a una puesta en escena, considerando que es la suma de los puntajes de todos los criterios que cumpla la puesta, teniendo como máximo 3 puntos en total.

BONUS: 3, 2, 1... ¡Tiempo! (este punto es opcional)

Por último, llega el momento de las batallas.

Una batalla se da siempre entre dos artistas. Cada artista deberá presentar diferentes puestas en escenas y la batalla consistirá de todas esas puestas en escena. El primer artista comenzará haciendo una puesta en escena, luego el segundo hará la suya, y de esta manera se irán turnando (de forma indefinida) hasta que la batalla termine.

Al final, el artista ganador es quien haya sumado más puntos por parte del conjunto de jurados de la batalla y es quien se lleva el cinto a la casa.

6. A partir de una batalla y un conjunto de jurados, saber qué artista se lleva el cinto a la casa.