

PdePesadilla



En el vasto paisaje de la programación nos encontramos a menudo desafíos inesperados. Hoy nos adentramos en un universo paralelo donde los conceptos familiares se entrelazan con lo desconocido. Nos adentramos en la mente de Suki, quien ha despertado de un sueño peculiar donde los errores y los fallos de programación son las criaturas de la noche.

En este mundo alternativo, el lenguaje Haskell es el vehículo que guía nuestra exploración. Aquí, las pesadillas son más que simples malos sueños; son manifestaciones de bugs y errores, desde fallos de segmentación hasta fugas de memoria que acechan en la oscuridad de cada línea de código.

La tarea que enfrenta Suki después de su despertar es monumental: modelar un sistema de pesadilla utilizando las herramientas que nos brinda la programación funcional

Las personas tienen en su memoria diferentes recuerdos ordenados de cierta manera. Durante la noche estos recuerdos pueden alterarse como consecuencia de las pesadillas que tenga.

```
suki = UnaPersona {  
    nombre = "Susana Kitimporta",  
    recuerdos = ["abuelita", "escuela primaria", "examen  
aprobado", "novio"]}
```

- Pesadilla de movimiento. Intercambia la ubicación de dos recuerdos. No es necesario verificar que la persona tenga una cantidad de recuerdos suficiente. *Por*

*ejemplo, si a los recuerdos de suki se le dice **mover** 1 3, hace que los recuerdos queden ["abuelita", "novio", "examen aprobado", "escuela primaria"]*

- Pesadilla de sustitución. Reemplaza el recuerdo de una determinada ubicación de su memoria por un recuerdo dado. No es necesario verificar que la persona tenga una cantidad de recuerdos suficiente. *Por ejemplo, si a los recuerdos de suki se le dice **reemplazar** 2 "examen desaprobado", hace que los recuerdos queden ["abuelita", "novio", "examen desaprobado", "escuela primaria"]*
- Pesadilla desmemorizadora. Hace que se olvide un determinado recuerdo. *Por ejemplo, si a los recuerdos de suki se le dice **quitar** "novio", hace que los recuerdos queden ["abuelita", "examen desaprobado", "escuela primaria"] si no tuviera dicho recuerdo no cambia nada*
- Pesadilla espejo: Hace que los recuerdos queden en el orden inverso al que estaban. *Por ejemplo, si a los recuerdos de suki se le dice **invertir**, hace que los recuerdos queden ["escuela primaria", "examen desaprobado", "novio", "abuelita"]*
- Sueño (no pesadilla) No provoca nada. *Por ejemplo, si a los recuerdos de suki se le dice **nop**, hace que los recuerdos queden ["abuelita", "novio", "examen aprobado", "escuela primaria"]*

1. Hacer que una persona pase una noche en la que se suceden una serie de pesadillas. Se quiere obtener cómo se encuentra la persona por la mañana siguiente.
2. Agregar una nueva pesadilla, utilizando una expresión lambda

Situaciones excepcionales

Se quiere experimentar con grupos de voluntarios y analizar cómo cada persona pasa la noche.

Por ahora se sabe que se pueden detectar las siguientes situaciones excepcionales, no excluyentes, pero podría haber otras.

- **Segmentation fault**: cuando la cantidad de pesadillas en la noche es mayor a la cantidad de recuerdos de la persona.
- **Bug inicial**: sucede cuando ya la primera de las pesadillas de la noche haría que a la persona se le modifique su primer recuerdo
- **Falsa alarma**: cuando pese a que al menos una de las pesadillas de la noche de manera aislada provocaría un cambio en la memoria, por cómo se da la secuencia de toda la noche, la persona se levantaría igual que como se fue a dormir.

Para un grupo de personas diferentes, asumiendo que todas van a soñar lo mismo durante la noche, se quiere saber:

- En cuántas se detectaría una determinada situación.
- Si alguna de estas situaciones se detecta en todas las personas.

Pesadillas infinitas

- ¿Si a una persona pasa una noche con infinitas pesadillas, ¿podríamos saber si hubo una falsa alarma? ¿Y un bug inicial? Justificar conceptualmente