

Simulacro Parcial

Pensemos casos de tests antes de resolverlos

Ejercicios simulacro

Para empezar a diseñar la novedosa y rupturista red social Y el famoso Elio Mark nos ha pedido que desarrollemos algunas funciones básicas, que tendrán como objetido representar algunas relaciones e interacciones entre los usuarios. Para esto nos envió las siguientes especificaciones en lenguaje semiformal y nos pidió que hagamos el desarrollo enteramente en Haskell, utilizando los tipos requeridos y solamente las funciones que se ven en Introducción a la Programación de Exactas-UBA.

Ejercicios simulacro

```
problema relacionesValidas (relaciones: seq<StringxString>) : Bool {  
  requiere: { True }  
  asegura: {  $res = True \iff relaciones$  no contiene ni tuplas repetidas1,  
             ni tuplas con ambas componentes iguales }  
}
```

¹ A los fines de este problema consideraremos que dos tuplas son iguales si el par de elementos que las componen (sin importar el orden) son iguales.

```
problema personas (relaciones: seq<StringxString>) : seq<String> {  
  requiere: { relacionesValidas(relaciones) }  
  asegura: {  $res$  no tiene elementos repetidos }  
  asegura: {  $res$  tiene exactamente los elementos que figuran en alguna  
             tupla de  $relaciones$ , en cualquiera de sus posiciones }  
}
```

Ejercicios simulacro

```
problema amigosDe (personas: String, relaciones: seq<StringxString>) :  
  seq<String> {  
    requiere: { relacionesValidas(relaciones) }  
    asegura: { res tiene exactamente los elementos que figuran en las tuplas  
              de relaciones en las que una de sus componentes es persona }  
}
```

```
problema personaConMasAmigos (relaciones: seq<StringxString>) : String {  
  requiere: { relaciones no vacía }  
  requiere: { relacionesValidas(relaciones) }  
  asegura: { res es el Strings que aparece más veces en las tuplas de  
            relaciones (o alguno de ellos si hay empate) }  
}
```