



Pipper++

Las redes sociales hoy en día están tan de moda que se generan una cada día. En el día de hoy nos han pedido a nosotros hacer una más para agregar a la lista. Nos dicen que esta red social va a permitir que un usuario pueda seguir personas, lo cual modelaremos de la siguiente forma:

```
data Usuario = Usuario{
    nombre :: Nombre,
    seguidos :: [Nombre]
}
type Nombre = String

marsupial = Usuario "@marsupialRengo" ["@don_churrascao", "@titoOk"]
churrascao = Usuario "@don_churrascao" []
tito = Usuario "@titoOk" ["@lapipi", "@marsupialRengo"]
pipi = Usuario "@lapipi" ["@titoOk"]
```

Cada usuario puede realizar Pips. Un Pip es un mensaje que podrán leer sus seguidores, marcarlos como favoritos o Repipearlos. Repipear un Pip es hacer que ese Pip ahora sea un mensaje tuyo. Un mensaje tiene la siguiente estructura:

```
data Mensaje = Mensaje {
    usuario :: Nombre,
    texto :: Texto,
    favs :: Cantidad
}
type Texto = String
type Cantidad = Int

unMensaje = Mensaje "@titoOk" "Las personas que viajan en tren se quejan de llenos" 100
otroMensaje = Mensaje "@lapipi" "En mi mundo todos son un pony" 0
otroMensajeMas = Mensaje "@lapipi" "y comen arcoiris y su popó son mariposas" 0
elUltimoMensaje = Mensaje "@titoOk" "No hay problema en cometer errores, el secreto es no pasarlos a producción" 3
```

1. Definir una función `valoracion` que dado un mensaje retorne la cantidad de favs menos la suma de la cantidad de caracteres del texto más la del nombre del usuario

```
> valoracion unMensaje
40 (surge de hacer 100 - (53 + 7))
```

2. Queremos saber, dados dos usuarios, si el primero sigue al segundo.

```
> sigueA tito pipi
True (Porque tito tiene en su la lista de seguidos a "@lapipi")
> sigueA churrascao marsupial
False (Porque churrascao no sigue a nadie)
```

3. Hacer la función `seguidores` que, dado un usuario y una lista de otros usuarios, devuelve los nombres de los seguidores del primer usuario.

```
> seguidores tito [marsupial, churrascao, pipi]
["@marsupialRengo", "@lapipi"] (Marsupial y pipi siguen a tito, churrascao no)
```

4. a. Definir una función para favear un mensaje que permita agregar un favorito al mensaje

```
> favear unMensaje
Mensaje "@titoOk" "Las personas que viajan en tren se quejan de llenos" 101
```

b. Definir una función `editar`, que permite modificar el texto de un mensaje, agregándole al inicio una nueva porción de texto dado.

```
> editar ";Ojo! " unMensaje
Mensaje "@titoOk" ";Ojo! Las personas que viajan en tren se quejan de llenos" 1
00
```

c. Definir una función `repipear` un mensaje por parte de otro usuario, que devuelve el nuevo mensaje con 0 favoritos, y cuyo contenido sea como el del mensaje original, sólo que editado automáticamente con el nombre del usuario original y un ": " al principio.

```
> repipear churrasco unMensaje
Mensaje "@don_churrasco" "@titoOk: Las personas que viajan en tren se quejan de
llenos" 0
```

5. Los usuarios también dudan en que acción realizar sobre un mensaje que vieron. Por ejemplo, no sabe si repipearlo o favearlo, o tal vez editarlo, o todo junto...

Definir una función que dado un mensaje y dos posibles acciones, realice la que permite obtener el pip con la mayor valoración, obteniendo como respuesta el mensaje resultante. Mostrar dos ejemplos de invocación usando acciones diferentes.