La previa

Con motivos del Mundial 2010, tenemos que practicar para hacer un sistema completamente verificable que maneje los resultados del torneo, para lo cual decidimos hacerlo usando Haskell.
Como todavía no nos tienen confianza tenemos que practicar con los torneos locales argentinos, para lo cual decidimos modelarlo de la siguiente manera:

- Un partido es una 4-upla *(equipo local, goles, equipo visitante,goles)*
- Una fecha es una 2-upla *(número de fecha, lista de partidos jugados)*
- Un torneo es una 2-upla *(nombre del torneo, lista de fechas)*
- Finalmente los torneos los metimos en una lista

```
torneos=[("Apertura 2008",partidosApertura), ("Clausura 2008",partidosClausura) ]

partidosApertura= [fecha1Apertura, fecha2Apertura]
partidosClausura= [fecha1Clausura, fecha2Clausura]

fecha1Apertura = (1, [("Boca",2, "Colon", 1),("Lanus",2, "Tigre", 0),
     ("Velez",0, "River", 1),("Racing",2, "Banfield", 2)])
fecha2Apertura = (2, [("Boca",2, "Velez", 1),("Lanus",2, "River", 0),
     ("Racing",0, "Tigre", 2),("Colon",2, "Banfield", 2)])
fecha1Clausura = (1, [("Boca",2, "Lanus", 1),("Racing",2, "River", 0),
     ("Velez",0, "Tigre", 1),("Colon",2, "Arsenal", 2)])
fecha2Clausura = (2,[("Boca",2, "Racing", 1),("Colon",2, "River", 0),
     ("Arsenal",0, "Tigre", 1),("Lanus",2, "Velez", 2)])
```

En función de los datos ya presentados, se solicita que se programen las funciones que se indican a continuación. Para su resolución, utilizar al menos una vez cada concepto e indicar dónde se hizo:

- Listas por comprensión
- Composición
- Aplicación parcial
- Función de orden superior
- Expresiones lambda

Consejos:

- Con todas las funciones armar una tabla con nombre, dominio, codominio, concepto/s usados
- Codificar funciones auxiliares para acceder a cada elemento de las tuplas con nombres expresivos para el dominio (ejemplo: local, golesLocal, etc).
- REUTILIZAR las funciones de los puntos anteriores!

Nota:

Se puede usar la función qfsort y la función concat. La primera ordena en base a un criterio, la segunda aplana una lista de listas:

```
qfsort f [ ] = [ ]
qfsort f (x:xs) = (qfsort f (filter ((> f x).f) xs))
      ++ [x]
      ++ (qfsort f (filter ((< f x).f) xs))
concat xs = foldl (++) [ ] xs
```

1a) **nombresTorneos/0**, retorna una lista con los nombres de todos los torneos.

```
> nombresTorneos
["Apertura 2008","Clausura 2008"]
```

1b) **segundoPara/2**, que recibe un valor y una lista de duplas, y la segunda componente de la dupla cuyo primer elemento es igual al valor enviado. Se puede asumir que siempre va a haber una y sólo una dupla con lo pedido

```
> segundoPara "Lanus" [("Boca",1),("River",2),("Lanus",3),("Velez",4)]
3
```

1c) **fechasDeTorneo/1,** dado un nombre de un torneo, devuelve su lista de fechas:

```
> fechasDeTorneo "Apertura 2008"
```

```
[(1,[("Boca",2,"Colon",1),("Lanus",2,"Tigre",0),("Velez",0,"River",1),("Racing",2,"Ban
field",2)]),(2,[("Boca",2,"Velez",1),("Lanus",2,"River",0),("Racing",0,"Tigre",2),("Co
lon",2,"Banfield",2)])]
```

1d) **partidosDeLaFecha/2,** dado el nombre de un torneo y un número de fecha, devuelve los partidos de esa fecha.

```
> partidosDeLaFecha "Clausura 2008" 2
[("Boca",2,"Racing",1),("Colon",2,"River",0), . . . ]
```

Punto 2) Programar una función de orden superior llamada **estadisticasDePartidos**/2, que recibe una función que calcula una estadística sobre un partido, y una fecha; y retorna la lista de estadísticas para los partidos de la fecha. A continuación, programe las siguientes funciones, que utilizan **estadisticasDePartidos**, de manera de obtener el resultado deseado (en todo caso, las funciones se definen invocando en algún momento a **estadisticasDePartidos**).

2a) Obtener la **lista** de equipos locales y visitantes para una fecha

```
> losLocales fecha1Apertura
["Boca","Lanus","Velez","Racing"]
> losVisitantes fecha1Apertura
["Colon","Tigre","River","Banfield"]
```

2b) Obtener la lista de diferencia de goles de cada partido.

```
>diferenciasDeGoles fecha1Apertura
[1,2,1,0]
```

2c) Obtener la **cantidad** total de goles convertidos en una fecha

```
>todosLosGoles fecha1Apertura
10
```

2d) Obtener la **cantidad** de victorias (o sea partidos que terminaron con un equipo haciendo más goles que el otro).

```
>contarVictorias fecha1Apertura
3
```

2e) Obtener la **lista de puntos** obtenidos por el equipo local. Las reglas son: se otorgan 3 puntos para una victoria del local, 1 punto si hay empate, y 0 puntos si hay derrota del local; e igual con el equipo visitante (ayuda: programar auxiliares para cada caso)

```
>puntosDeLocal fecha1Apertura (para el equipo local)
[3,3,0,1]
>puntosDeVisitante fecha1Apertura (para el equipo visitante)
[0,0,3,1]
```

3a) **puntosEnFecha/2**, que recibe un equipo y una fecha, y retorna la cantidad de puntos que hizo el equipo en esa fecha (se aplican las mismas reglas descriptas arriba). Ojo! El equipo pudo haber jugado como local o como visitante.

```
> puntosEnFecha "Boca" fecha1Apertura
3
```

3b) **puntosTorneo/2**, que recibe un equipo y un nombre de torneo, y retorna los puntos que hizo el equipo en el torneo.

```
> puntosTorneo "Boca" "Apertura 2008"
6
```

3c) **participantesTorneo,** que recibe un nombre de torneo y retorna la lista de equipos que participaron en el torneo (con analizar 1 sola fecha alcanza, los equipos son los mismos entre fechas).

```
> participantesTorneo "Apertura 2008"
["Boca","Colon","Lanus","Tigre","Velez","River","Racing","Banfield"]
```

3d) **ganadorTorneo/2,** que recibe un nombre de torneo y retorna el nombre del equipo ganador.

```
> ganadorTorneo "Apertura 2008"
"Boca"
```