VamoAJuntarno.org

Ante la eterna problemática para organizar reuniones de amigos, que siempre tienen algo para objetar cuando se pone una fecha, queremos hacer una herramienta para que los distintos invitados puedan indicar, de la forma más automática posible, para cuándo pueden sumarse y así poder elegir la mejor opción para distintos días propuestos.

Tenemos modelada la información de la siguiente forma:

```
data Dia = Dia { año :: Int, mes :: Int, dia :: Int } deriving Eq
data Invitado = Invitado { nombre :: String,
                           calendario :: [ RestriccionParaJuntarse ] }
```

Además, ya tenemos una función cantidadDeDiasEnMes, que dado un número de mes y de año nos dice la cantidad de días que tiene ese mes:

```
> cantidadDeDiasEnMes 12 2017
=> 31
```

Resolver los siguientes puntos **aprovechando los conceptos del paradigma funcional**, en particular se espera que se demuestre dominio de orden superior, composición y aplicación parcial. **Para cada función pedida, declarar el tipo de la misma**.
Consejo: Leer todo el enunciado antes de empezar.

Para arrancar, necesitamos poder trabajar con los calendarios de los invitados...

1.
    a. Declarar el tipo de dato RestriccionParaJuntarse usado en el calendario de los invitados, sabiendo que una restricción nos indicará si un día en particular lo tenemos ocupado (es decir, NO disponible).
    b. Dados un conjunto de días, determinar en cuáles puede juntarse cierto invitado. Un invitado puede juntarse en un día únicamente si para ese día todas las restricciones de su calendario se lo permiten. Cada restricción le permitirá ir a una reunión cuando no se cumpla para el día de la misma.
    c. ¿Sería posible determinar en qué días puede juntarse un invitado con una cantidad infinita de restricciones? Justificar conceptualmente.

2. Declarar las siguientes funciones de modo que puedan usarse para generar restricciones para juntarse:
    a. tengoUnaCita, que es verdadero si el día es el mismo que el día que se tiene la cita.
    b. esFeriado, que dada una lista de días feriados nos dice si el día en cuestión es feriado.
    c. podriaIrIndeseable, que retorna verdadero si, a partir de una persona a la que se quiere evitar, esa persona podría juntarse ese día en base a sus propias restricciones para juntarse.
    d. esFinDeMes, es verdadero si el día es uno de los últimos 5 días del mes.
    e. uhJustoTengoTurnoConElDentista, que da verdadero siempre, sea el día que sea (sospechamos que simplemente puede ser una excusa).

    Mostrar un ejemplo de creación de un invitado que incluya en su calendario una restricción de cada tipo de las que se mencionan en este punto.

3. Necesitamos que se le pueda agendar una cita a alguien para cierto día mediante una restricción en su calendario. Por ejemplo, hacer que Carlos tenga una cita el 5 de Agosto de este año. De esa forma ya no podrá comprometerse para otros eventos del mismo día.

Ahora queremos organizar y confirmar reuniones:

4. Una reunión está determinada por los días posibles para llevarla a cabo, y el día oficial de la misma se obtiene a partir de la disponibilidad de los invitados para asistir.
    a. Dada una reunión y una lista de invitados, necesitamos determinar el día que más gente pueda juntarse. En caso de haber más de un día con esa misma cantidad de gente, cualquiera de ellos es una respuesta adecuada.
    b. Queremos poder confirmar una reunión con un conjunto de invitados, que implica elegir como día para reunirse el día que más gente pueda ir y hacer que los invitados que puedan ir ese día se comprometan a ese evento agendándose una cita para ese día (los demás no deberían cambiar su calendario).

5. Finalmente, queremos organizar una maratón de reuniones, esto significa que dadas unas reuniones con sus días disponibles, y un grupo de invitados, hacer que los invitados traten de confirmar todas las reuniones sucesivamente (un invitado que se comprometió a una reunión no debería luego comprometerse a otra el mismo día).