**NOTA: 8**

**import** Text.Show.Functions()

**import** Data.List ()

*--PARTEA*

**data** **Persona** **=** **Persona**{

nombre**::** **String**,

calorias**::** **Int**,

indiceHidratacion**::** **Int**,

tiempoDisponible**::** **Float**,

equipamiento**::** [**Equipo**]

}

**type** **Equipo** **=** **String**

*--ACCESSORS--------------------------------------------------------------------------------------------------*

*-------------------------------------------------------------------------------------------------------------*

mapNombre**::**(**String** **->** **String**) **->** **Persona** **->** **Persona**

mapNombre f unaPersona **=** unaPersona{nombre **=** f**.**nombre **$** unaPersona}

mapCalorias**::** (**Int** **->** **Int**) **->** **Persona** **->** **Persona**

mapCalorias f unaPersona **=** unaPersona{calorias **=** f**.**calorias **$** unaPersona}

pierdeCalorias**::** **Int** **->** **Persona** **->** **Persona**

pierdeCalorias unValor unaPersona **=** (mapCalorias**.**subtract) unValor **$** unaPersona

mapIndiceHidratacion**::** (**Int** **->** **Int**) **->** **Persona** **->** **Persona**

mapIndiceHidratacion f unaPersona**=** unaPersona{indiceHidratacion **=** f**.**indiceHidratacion **$** unaPersona}

Según el enunciado, la hidratación puede estar entre 0 y 100; esa lógica se podría agregar acá

pierdeHidratacion**::** **Int** **->** **Persona** **->** **Persona**

pierdeHidratacion unValor unaPersona **=** (mapIndiceHidratacion**.**subtract)unValor**$** unaPersona

mapTiempoDisponible**::** (**Float->** **Float**) **->** **Persona** **->** **Persona**

mapTiempoDisponible f unaPersona **=** unaPersona{tiempoDisponible **=** f**.**tiempoDisponible **$** unaPersona}

mapEquipamiento**::** ([**String**] **->** [**String**]) **->** **Persona** **->** **Persona**

mapEquipamiento f unaPersona **=** unaPersona{equipamiento **=** f**.**equipamiento **$** unaPersona}

*--The end Accessors--------------------------------------------------------------------------------------------*

*---------------------------------------------------------------------------------------------------------------*

*--A.1*

**type** **Ejercicio** **=** **Persona** **->** **Persona**

abdominales**::** **Int** **->** **Ejercicio**

abdominales unasRepeticiones unaPersona **=** pierdeCaloriasPorCantidadDeRepeticiones unasRepeticiones 8 unaPersona

pierdeCaloriasPorCantidadDeRepeticiones**::** **Int** **->Int** **->** **Persona** **->** **Persona**

pierdeCaloriasPorCantidadDeRepeticiones unasRepeticiones unValor unaPersona **=** pierdeCalorias (unasRepeticiones **\*** unValor) unaPersona

*--A.2*

flexiones**::** **Int** **->** **Ejercicio**

flexiones unasRepeticiones unaPersona **=**

pierdeCaloriasPorCantidadDeRepeticiones unasRepeticiones 16**.**pierdeHidratacionPorCantidadDeRepeticiones unasRepeticiones 2 **$** unaPersona

pierdeHidratacionPorCantidadDeRepeticiones**::** **Int** **->** **Int** **->** **Persona** **->** **Persona**

pierdeHidratacionPorCantidadDeRepeticiones unasRepeticiones unValor unaPersona **=** pierdeHidratacion (div (unasRepeticiones **\*** unValor) 10 ) unaPersona

No está bueno que uses dos nombres tan parecidos para dos funciones que tienen comportamiento distinto

*--A.3*

levantarPesas**::** **Int** **->** **Int** **->** **Ejercicio**

levantarPesas unasRepeticiones unPesoPesa unaPersona

**|**tieneEntreSuEquipamiento "pesa" unaPersona **=** pierdeCaloriasPorCantidadDeRepeticiones unasRepeticiones 32**.**pierdeHidratacionPorCantidadDeRepeticiones unasRepeticiones unPesoPesa **$** unaPersona

**|** otherwise **=** unaPersona

tieneEntreSuEquipamiento**::** **String** **->** **Persona** **->** **Bool**

tieneEntreSuEquipamiento unEquipo unaPersona **=** elem unEquipo (equipamiento unaPersona)

*--A.4*

laGranHomeroSimpson**::** **Ejercicio**

laGranHomeroSimpson **=** id

*--PARTEB*

*--TUPLA*

**type** **Rutina=** (**Float**, [**Ejercicio**])

duracion**::** **Rutina** **->** **Float**

duracion unaRutina **=** fst unaRutina

rutina **::** **Rutina** **->** [**Ejercicio**]

rutina unaRutina **=** snd unaRutina

Es un poco raro que al segundo miembro de la rutina lo estés llamando *rutina.* Para distinguir entre la rutina data y su segundo miembro podrías llamarlo ejercicios.

puedePersonaHacerRutina**::** **Persona** **->** **Rutina** **->** **Bool**

puedePersonaHacerRutina unaPersona unaRutina **=** tiempoDisponible unaPersona **>=** duracion unaRutina

*--B.1*

esPeligrosa**::** **Rutina** **->Persona** **->** **Bool**

esPeligrosa unaRutina unaPersona **=** verificiSiQuedaAgotada unaRutina unaPersona

verificiSiQuedaAgotada**::** **Rutina** **->** **Persona** **->** **Bool**

verificiSiQuedaAgotada unaRutina unaPersona **=** verificaHidratacionCalorias (**<**50) (**<**10)**.**personaHaceRutina unaRutina **$** unaPersona

personaHaceRutina**::Rutina** **->** **Persona** **->** **Persona**

personaHaceRutina unaRutina unaPersona

**|** puedePersonaHacerRutina unaPersona unaRutina **=** foldr (**$**) unaPersona (rutina unaRutina )

**|** otherwise **=** unaPersona

verificaHidratacionCalorias**::**(**Int** **->** **Bool**) **->** (**Int** **->** **Bool** )**->** **Persona** **->** **Bool**

verificaHidratacionCalorias unCriterio otroCriterio unaPersona **=** (unCriterio**.**calorias) unaPersona **&&** (otroCriterio**.**indiceHidratacion) unaPersona

Esta abstraccion no está muy buena porque:

es dicifil encontrar un buen nombre para ella (de hecho, fijate que los criterios los está recibiendo al revés de como sugeriría el nombre)

no aporta mucho en terminos de declaratividad

*--B.2*

esBalanceada**::** **Rutina** **->** **Persona** **->** **Bool**

esBalanceada unaRutina unaPersona **=** verificaHidratacionCalorias (**>**80) (**<** mitadaCaloriasInicial unaPersona)**.**personaHaceRutina unaRutina **$** unaPersona

mitadaCaloriasInicial**::** **Persona** **->** **Int**

mitadaCaloriasInicial unaPersona **=** div (calorias unaPersona) 2

“Inicial” tiene sólo sentido dentro de esBalanceada; quizás sería mejor sólo llamarla mitadCalorias

*--B.3*

*--elAbominableAbdominal:: Rutina -> Rutina*

*--elAbominableAbdominal unaRutina = (55.5 , ejerciciosInfinitos (abdominales 3))*

*--ejerciciosInfinitos:: Ejercicio -> [Ejercicio]*

*--ejerciciosInfinitos unEjercicio = take unEjercicio (repeat unEjercicio)*

*--PARTEC*

*--C.1*

seleccionarGrupoDeEjercicio**::** **Persona** **->** [**Persona**] **->** [**Persona**]

seleccionarGrupoDeEjercicio unaPersona unasPersonas **=** filter( mismoTiempoDisponible unaPersona) unasPersonas

mismoTiempoDisponible**::** **Persona** **->** **Persona** **->** **Bool**

mismoTiempoDisponible unaPersona otraPersona **=** tiempoDisponible unaPersona **==** tiempoDisponible otraPersona

*--C.2*

promedioDeRutina**::** **Rutina** **->** [**Persona**] **->** **Int**

promedioDeRutina unaRutina unasPersonas **=** (promedioCalorias**.**map (personaHaceRutina unaRutina)) unasPersonas **+** (promedioHidratacion**.**map (personaHaceRutina unaRutina)) unasPersonas

No devuelve lo que se pide; la idea era devolver los dos promedios por separado (haciendo uso de una tupla, por ejemplo)

Repetición de lógica en los cálculos de promedios y en hacer que un conjunto de personas haga una rutina

promedioCalorias**::** [**Persona**] **->** **Int**

promedioCalorias unasPersonas **=** div (sum**.**map calorias **$** unasPersonas) (length unasPersonas)

promedioHidratacion**::** [**Persona**] **->** **Int**

promedioHidratacion unasPersonas **=** div (sum**.**map indiceHidratacion **$** unasPersonas) (length unasPersonas)