



Trabajo Práctico 1

Programación Declarativa

BOZZANO, Felipe – Legajo: 102050

OLMOS, Francisco – Legajo: 102369

Profesor: Maximiliano Andrés Eschoyez

Problemática:

El objetivo de este proyecto es implementar un programa cuyo input es un archivo de texto con la información de horarios de entrada y salida de empleados de una empresa y el output es el siguiente:

- Por cada registro, imprimir día de la semana, fecha y hora trabajadas.
- Cantidad de empleados.
- Cantidad de horas acumuladas en el mes por todos los empleados.
- Cantidad de horas/hombre en el mes por todos los empleados.
- Promedio diario de horas/hombre.

El lenguaje a utilizar es Haskell, para ser implementado en un paradigma funcional, y se deberá implementar un módulo para el dato de tipo Hora conformado por dos valores de tipo Int, re definición de la función Show para imprimir las horas y minutos (HH:MM), su constructor, getters de hora y minuto, funciones de normalización y des normalización a minutos desde la hora cero y función para calcular la diferencia horaria entre dos Horas.

El programa cuenta con una función Main que hace uso del módulo Hora y es donde se procesa este archivo de texto.

Solución:

Para la implementación del programa en primer medida creamos un módulo Hora con todos requerimientos descritos en la problemática.

Por otro lado leímos el archivo registro por registro y filtramos según lo que se nos pedía en cada consigna.

Para el caso de los registros de entrada y salida generamos tuplas del estilo (entrada, salida) y verificamos que para cada una de ellas existiesen ambos valores, en caso de que la tupla no estuviera completa la tomamos como 0. Utilizamos el módulo Hora para trabajar estas tuplas Y convertirlas en datos que luego mostramos como resultado.

Para el caso de la cantidad de empleados, si bien el archivo contenía Información sobre la misma, filtramos y contamos la cantidad total y verificamos que no era la misma.

El código fuente posee comentarios donde se informa al usuario sobre los detalles de cada función y variables creadas.

Como compilar el programa:

- Ubicarse en la raíz del proyecto.
- Ejecutar: `ghc main.hs && ./main`

Conclusión:

Al ser un requerimiento el utilizar Haskell para el desarrollo del programa, aprendimos que el paradigma funcional deja de lado la escuela de trabajar con objetos que conlleva un tipo de escritura que a veces es difícil de leer, a cambio de una forma de programar de manera declarativa, en donde escribir código es más similar a escribir la solución del problema volviéndolo más intuitivo y semántico.

Al ser un paradigma nuevo para nosotros, al principio nos costó adaptarnos a la forma de trabajo funcional donde básicamente todo es una sucesión de funciones y es fácil perder el hilo, pero comprendemos la potencia de este paradigma en donde con pocas líneas de código se puede hacer un gran trabajo computacional que quizás en otro conlleva más desarrollo para obtener el mismo resultado.

Nos pareció interesante el concepto de evaluación perezosa que retrasa el cálculo de una expresión hasta que su valor es necesario, lo cual vimos reflejado cuando debuguemos el programa ya que no nos hizo falta comentar expresiones que no utilizábamos para poder imprimir resultados Parciales al momento de desarrollar la solución del programa.