

Taller de Programación Relacional
Modelos y Paradigmas de Programación
Escuela de Ingeniería de Sistemas y Computación
Universidad del Valle



Profesor Juan Francisco DIAZ FRIAS*

A continuación Usted encontrará una serie de ejercicios que debe resolver **individualmente**. No se admite ninguna consulta con sus compañeros de otros grupos de trabajo, pero sí con el profesor. Esa es la hipótesis básica sobre la cual se fundamenta esta tarea. Cualquier violación a esta regla será considerada un intento de copia y causará la anulación del taller.

La solución a todos y cada uno de los puntos del taller se debe almacenar en un archivo `.oz` tal como se describe en cada punto. Usted debe entregar como solución al examen, el conjunto de esos archivos empaquetado y comprimido en un solo archivo cuyo nombre debe ser análogo a `taller6JFDiaz.zip` o `taller6JFDiaz.tgz` o `taller6JFDiaz.tar.gz` o `taller6JFDiaz.rar`. Al desempaquetar este archivo, debo encontrar solamente los archivos siguientes: `MinimoRelacional.oz` con el contenido descrito abajo. Adicionalmente debe entregar un archivo `Practica6.pdf`, con el desarrollo del taller que no corresponde a código Oz.

No envíe ningún archivo adicional, así lo haya utilizado para sus pruebas. Su solución debe ser enviada a través del Campus Virtual en el enlace correspondiente al Taller de Programación Relacional según las fechas indicadas allí. El sistema no permitirá que se envíen soluciones después de dicha fecha.

*juanfco.diaz@correounivalle.edu.co

Un problema típico de programación, es el problema de ordenar los elementos de una lista de acuerdo a una llave. En este ejercicio abordaremos este problema desde el modelo relacional. Supongamos que tenemos una lista de tuplas $[t(e_1 \text{ ll}_1) \ t(e_2 \text{ ll}_2) \ \dots \ t(e_n \text{ ll}_n)]$ y que queremos ordenarla de acuerdo a la llave que acompaña cada elemento de la tupla. Si no hay llaves repetidas, sólo hay una solución. Pero si hay llaves repetidas, hay varias soluciones. El objetivo de este ejercicio es calcular todas las soluciones, usando el modelo relacional.

1. Calculando el menor de dos tuplas de forma relacional. Escriba un procedimiento `Menor` que relacione dos tuplas `T1` y `T2` con una variable sin ligar `Min` de manera que `Min` represente la tupía con la menor llave. Si la llave es igual, `Min` puede representar cualquiera de las dos tuplas.

Su procedimiento será probado con ejemplos como los siguientes:

```
proc {Menor T1 T2 Minimo}
    % su código
end

{Browse {SolveAll proc {$ Min} {Menor t(a 3) t(b 5) Min} end }}
{Browse {SolveAll proc {$ Min} {Menor t(a 5) t(b 3) Min} end }}
{Browse {SolveAll proc {$ Min} {Menor t(a 3) t(b 3) Min} end }}
```

2. Calculando el mínimo de una lista de forma relacional. Escriba el procedimiento `PMinimo` que recibe una lista `Lista` de tuplas de la forma $t(e \text{ ll})$ y una variable sin ligar `Minimo` y las relaciona de forma que `Minimo` representa una tupla cuya llave es la menor de todas las llaves. Su procedimiento debe usar el procedimiento `Menor` del ejercicio precedente.

Su procedimiento será probado con ejemplos como el siguiente:

```
proc {Menor T1 T2 Minimo}
    % su código
end

proc {PMinimo Lista Minimo}
    % su código
end

fun {MinimoLista L}
    proc {$ M}
        {PMinimo L M}
    end
end

{Browse {SolveAll {MinimoLista [t(d 2) t(b 3) t(c 4) t(a 2) t(e 5)]}}}
```

3. Escriba el procedimiento `LOrd` que recibe una lista `Lista` de tuplas de la forma $t(e \text{ ll})$ y una variable sin ligar `ListaOrd` y las relaciona de forma que `ListaOrd` representa una lista con las mismas tuplas que `Lista` pero ordenadas según sus respectivas llaves. Su procedimiento debe usar el procedimiento `PMinimo` del ejercicio precedente.

Su procedimiento será probado con ejemplos como el siguiente:

```
proc {Menor T1 T2 Minimo}
    % su código
end

proc {PMinimo Lista Minimo}
    % su código
end

proc {LOrd Lista ListaOrd}
    % su código
end

fun {OrdenarLista Lista}
    proc {$ LO}
        {LOrd Lista LO}
    end
end

{Browse {SolveAll {OrdenarLista [t(d 2) t(b 3) t(c 4) t(a 2) t(e 5)]}}}
```

Almacene su implementación de los procedimientos Menor, PMinimo, y LOrd en el archivo `MinimoRelacional.oz`.