**Ejercicio 1.1:**

local L L2 Length in

fun {Length Xs}

case Xs of nil then

0

[] H|T then

1 + {Length T}

end

end

L = 1|2|3|4|5|6|nil

L2 = nil

{Browse {Length L}}

{Browse {Length L2}}

end

**Ejercicio 1.2:**

local L Take in

fun {Take Xs N}

if N > 0 then

case Xs of nil then

nil

[] H|T then

H|{Take T N-1}

end

else

nil

end

end

L = [1 2 3 4 5 6]

{Browse {Take L 4}}

end

**Ejercicio 1.3:**

local L Drop in

fun {Drop Xs N}

case Xs of nil then

nil

[] H|T then

if N > 0 then

{Drop T N-1}

else

Xs

end

end

end

L = [1 2 3 4 5 6]

{Browse {Drop L 0}}

end

**Ejercicio 1.4:**

local L L2 Append in

fun {Append Xs Ys}

case Xs of nil then

Ys

[] H|nil then

H|Ys

[] H|T then

H|{Append T Ys}

end

end

L = [1 2 3 4 5 6]

L2 = [7 8]

{Browse {Append L L2}}

end

**Ejercicio 1.5:**

local L X Member in

fun {Member Xs Y}

case Xs of nil then

false

[] H|T then

if (H == X) then

true

else

{Member T X}

end

end

end

L = [1 2 3 4 5 6]

X = 6

{Browse {Member L X}}

end

**Ejercicio 1.6:**

local L X Position in

fun {Position Xs X}

local I in

Pos = 1

{PositionAux Xs X Pos}

end

end

fun {PositionInt Xs X Pos}

case Xs of H|T then

if (H == X) then

Pos

else

{PositionInt T X Pos + 1}

end

end

end

L = [1 2 3 9 5 6]

X = 9

{Browse {Position L X}}

end

**Ejercicio 2**

1. proc {P X Y} local Z in {Q Z U} end end

2. proc {P X Y} local Z in {Q Z Y} end end

3. proc {P X Y} local Z in {P Z Y} end end

1. Referencias externas: Q y U

2. Referencia externa: Q

3. No hay referencias externas

**Ejercicio 5.1:**

proc {Length Xs N}

case Xs of nil then

N = 0

[] X|Xr then

local N int in

{Length Xr Naux}

N = Naux + 1

end

end

end

**Ejercicio 5.2:**

proc {LengthTail Xs N NTail}

case Xs of nil then

N = NTail

[] X|Xr then

{LengthTail Xs N NTail+1}

end

end

proc {Length Xs N}

{LengthTail Xs N 0}

end

La ventaja de usar "tail recursive" es que no se apila un nuevo frame en el stack al efectuar la llamada, sino que se utiliza el frame actual. De ésta manera el procesador puede efectuar el cálculo de manera más fácil y rápida.