**RECIBIR MENSAJES**

m = nuevo\_mensaje()

tipo = m.info\_mensaje.tipo

if tipo == “MSI”:

cont++ //inicializado a 0

m.info\_mensaje.emisor.ID = cont

if m.info\_mensaje.tipo\_agente == “tienda”:

contT++ //inicializado a 0

else:

contC++ //inicializado a 0

send\_ack(m)

else if tipo == “MFO”:

preguntar en clase: es necesario el ack? Hay que comprobar que no haya clientes en la tienda o eso lo hace la tienda? Hay que informar clientes (que tengan esa tienda en su lista) de que esa tienda ya no está disponible y proporcionar un reemplazo. Si un cliente termina, hay que hacer algo, o ya se supone que ha salido de la tienda.

endif

$sentencia = “INSERT …”

if (!my\_sql\_query($sentencia)) {

            //----------TODO: error al insertar info en bbdd

        }

**ASIGNAR CONFIGURACIÓN INICIAL**

Constantes: n\_prod //número de productos distintos (pan, agua y vino: n\_prod=3)

n\_tiendas = contT

n\_comp = contC

listasP\_t,listasP\_c = generar\_listas\_productos(n\_tiendas,n\_comp,n\_prod)

listasT = generar\_listas\_tiendas(n\_tiendas,n\_comp)

compradores = 0 //contadores

tiendas = 0

for (i=1;i<=cont;i++):

m = BBDD.mensajes(id==i)

if m.info\_mensaje.tipo\_agente == “comprador”:

listaP = listasP\_c[compradores]

listaT = listasT[compradores]

BBDD.compradores[i].listaP = listaP

BBDD.compradores[i].listaT = listaT

lista\_MCI\_c[compradores] = generar\_MCI(listaP,listaT,m)

compradores++

else

listaP = listasP\_t [tiendas]

listaT = []

BBDD.tiendas[i].listaP = listaP

lista\_MCI\_t[tiendas] = generar\_MCI(listaP,listaT,m)

tiendas++

endif

endfor

**generar\_listas\_productos(n\_tiendas,n\_comp,n\_prod)**

listasP\_t

listasP\_c

for i=0;i<n\_comp;i++:

listasP\_c[i] = lista\_aleatoria(tamaño=n\_prod,min=0,max=5) // el 5 me lo he inventao

// Teniendo productos [“Pan”, “Agua”]

// Ejemplo resultado lista aleatoria [0, 2]

// Ejemplo listasP\_c [[0,2],[3,5],[1,0]] si hubiese 3 compradores

// Ejemplo min\_prod [(0+3+1),(2+5+0)]

min\_prod = suma\_por\_columnas(listasP\_c)

for i=0;i<n\_tiendas;i++:

lista = new Lista(n\_prod)

for j=0;j<n\_prod-1;j++:

random = aleatorio(0,min\_prod[j])

min\_prod[j]-=random

lista[j] = random

lista[n\_prod-1] = minprod

listasP\_t[i] = lista

**generar\_listas\_tiendas(n\_tiendas,n\_comp**)

t = 1 //Aqui asumimos que los índices de clientes y tiendas son indepes, si no habría que

//cambiarlo

for i=0;i<n\_conp;i++:

listasT[i] = [t,t+1]

t++

if(t==n\_tiendas):

t=1

endif

**ARRANQUE**

for (i=1;i<=n\_tiendas;i++)

send(lista\_MCI\_t[i])

endfor

for (i=1;i<=n\_comp;i++)

send(lista\_MCI\_c[i])

endfor

**GENERAR ESTADÍSTICAS**

Clientes: productos faltan, productos quedan, tiendas asignadas, tiendas visitadas, estado(ON/OFF)

Tiendas: stock, productos vendidos, nº de clientes que han estado en la tienda, estado(ON/OFF)

Mensajes: tipo mensaje, emisor, receptor, detalle de mensaje

Generales: nº mensajes, nº tiendas activas, nº clientes activos