

PRÀCTICA DES:

Recapte d'aliments

SIM

Laia Ondoño laia.ondono@est.fib.upc.edu
Anna Llanza anna.llanza@est.fib.upc.edu

Definició del sistema

En aquesta pràctica, com a les anteriors, hem decidit escollir el tema de la recaptació d'aliments. Tot i que la idea inicial ha estat com la de la modelització d'objectes 3D amb Flexsim, per a aquest sistema hem reduït considerablement el seu volum. És per això que només abarquem una part del procés que té lloc al supermercat, en el qual s'obtenen aliments que, posteriorment, es van agrupant en capsas en funció del seu tipus fins que aquestes s'omplen. El nostre simulador inclou 4 objectes: una source (generador), 3 cues (àrees d'espera, una per cada entitat diferent), 3 agrupadors (un per cada entitat diferent) i una sink. Les entitats del sistema són els aliments que es recapten al supermercat. Aquests tenen 2 atributs. D'una banda està el tipus, que pot ser pasta, arròs o llegums. De l'altra banda, tenim l'estat en què aquest es troba, que pot ser recaptat, processat i finalitzat.

El que volem analitzar és la quantitat de capsas de cada tipus que es generaran, a més del temps mitjà que es triga en omplir una capsa de cadascun d'aquests tipus. El temps en omplir una capsa està condicionat pel temps entre arribades, que hem configurat per a que segueixi una distribució exponencial (fent ús de la llibreria Scipy de Python i la crida al mètode `expon.rvs(size, loc, scale)`) de la qual es poden modificar els paràmetres `loc` i `scale`.

A més, la segona distribució considerada al nostre sistema l'hem utilitzat per a generar els valors del tipus de cada aliment. Això ho hem fet mitjançant una distribució uniforme que només generi valors enters entre 1 i 3. Hem utilitzat la llibreria de Python Numpy i la crida a la funció `random.randint(min, max, size)` que aquesta ofereix.

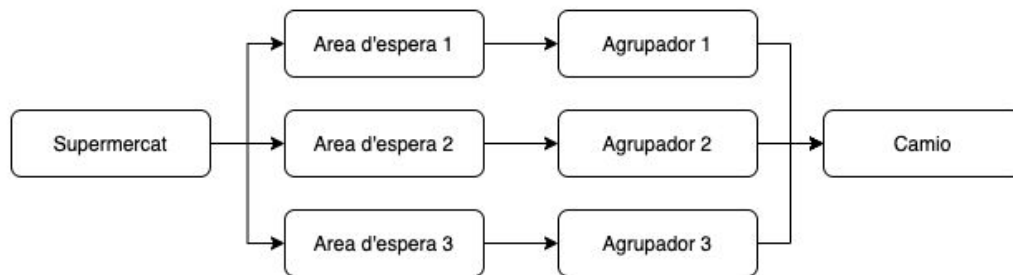
Pel que fa a les hipòtesis considerades al nostre sistema, hem trobat necessari incloure les següents:

- En un recapte d'aliments només s'accepten 3 tipus d'aliments: pasta, arròs, llegums.
- El nombre d'aliments a processar ha de ser divisible per la capacitat dels agrupadors.
- Una unitat de temps representa 1 minut.
- Un aliment només pot anar a l'agrupador que s'encarrega de classificar els aliments del seu mateix tipus.
- Un aliment només pot anar a la cua que ha estat assignada per a contenir aliments del seu mateix tipus.
- L'agrupador omple una capsa quan processa tantes entitats com la capacitat de l'agrupador indicada al fitxer *configuracio.py*.
- No es finalitza un aliment fins que s'omple una capsa.
- El fet de que l'últim aliment afegit a la capsa passi d'estat processat a estat finalitzat comporta que la resta dels aliments de la mateixa capsa es considerin també com a finalitzats.
- L'agrupador no s'espalla en cap moment.

Especificació del sistema

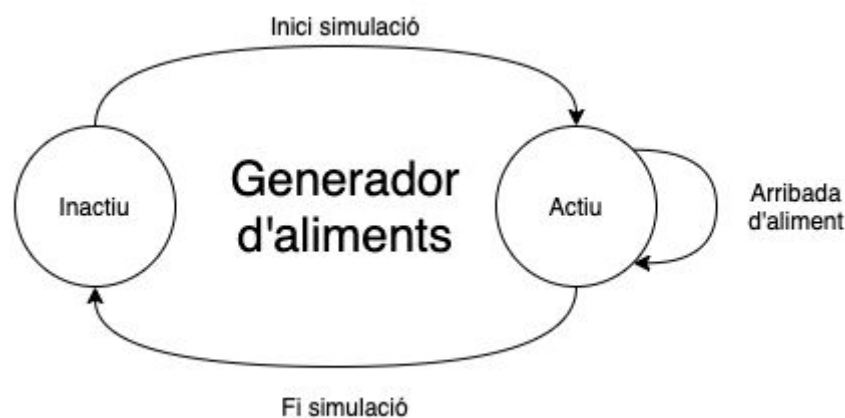
Component estàtic

El següent diagrama mostra l'anàlisi estàtic del sistema. Podem observar els components que formen part del sistema i el flux que seguiran les entitats des que es generen (al supermercat) fins que desapareixen (camió). Quan una entitat (aliment) es recapta al supermercat, es comprovarà si l'agrupador està processant un altre aliment. Si és així, aquest aliment es portarà a la cua corresponent i serà processat una vegada l'agrupador quedi lliure. En cas contrari, l'aliment passarà directament a ser processat per l'agrupador. Per a que un aliment passi al camió, l'agrupador que l'ha processat haurà d'omplir tota la capsa.

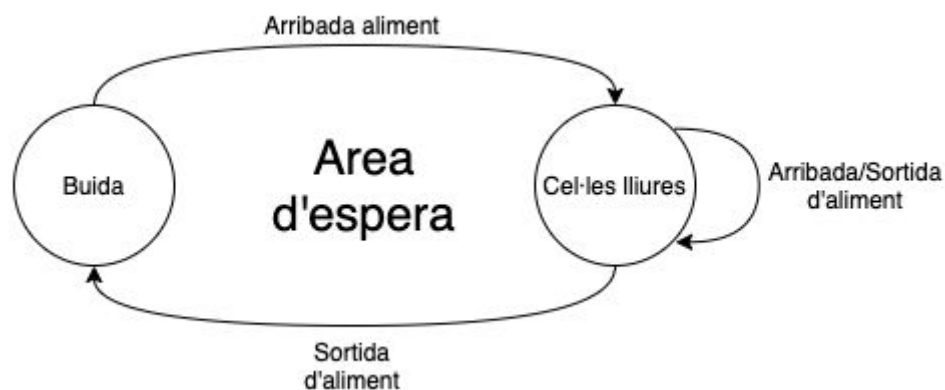


Components dinàmics

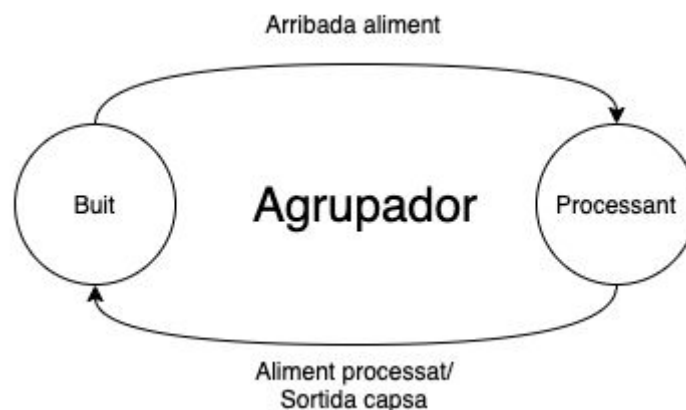
Abans d'iniciar la simulació, el nostre generador d'aliments es troba en estat inactiu. Un cop la simulació és iniciada, el generador passarà a estar en estat actiu, en el qual romandrà sempre que li arribin aliments. Quan la simulació finalitzi, és a dir, quan ja no hi hagi més esdeveniment pendants, aquest tornarà a estar en estat inactiu.



Les 3 àrees d'espera tenen el mateix comportament dinàmic. L'estat inicial de cadascuna d'aquestes és buida, però, si l'agrupador corresponent al tipus està processant i li arriba un altre aliment a processar, aleshores, la cua passarà a estar en estat "cel·les lliures". L'àrea d'espera estarà en aquest estat quan tingui un o més aliments pendents de ser processats per l'agrupador corresponent. Només passarà a estar buida quan l'agrupador comenci a processar l'últim esdeveniment d'aquesta cua.

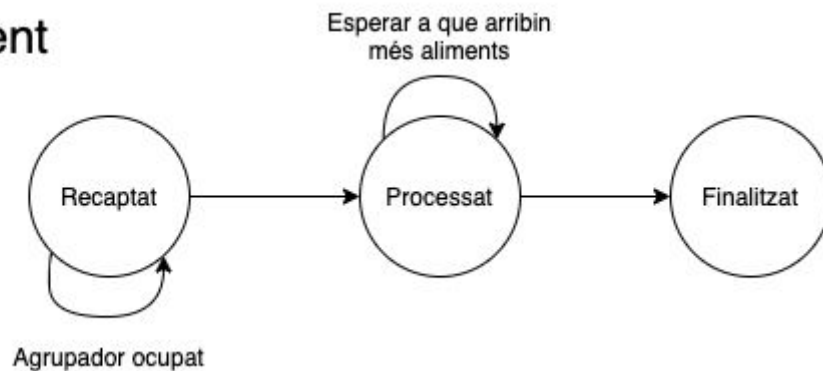


Igual que les àrees d'espera, els 3 agrupadors tenen el mateix comportament dinàmic. El seu estat inicial és buit, però, si li arriba un aliment, aquest passarà a estat processant. Quan l'acaba de processar tornarà a estar en estat buit. Quan s'acaba de processar l'últim aliment d'una capsula, també passarà a estat buit i, consegüentment, farà que tots els aliments de la capsula passin a estar en estat finalitzat.

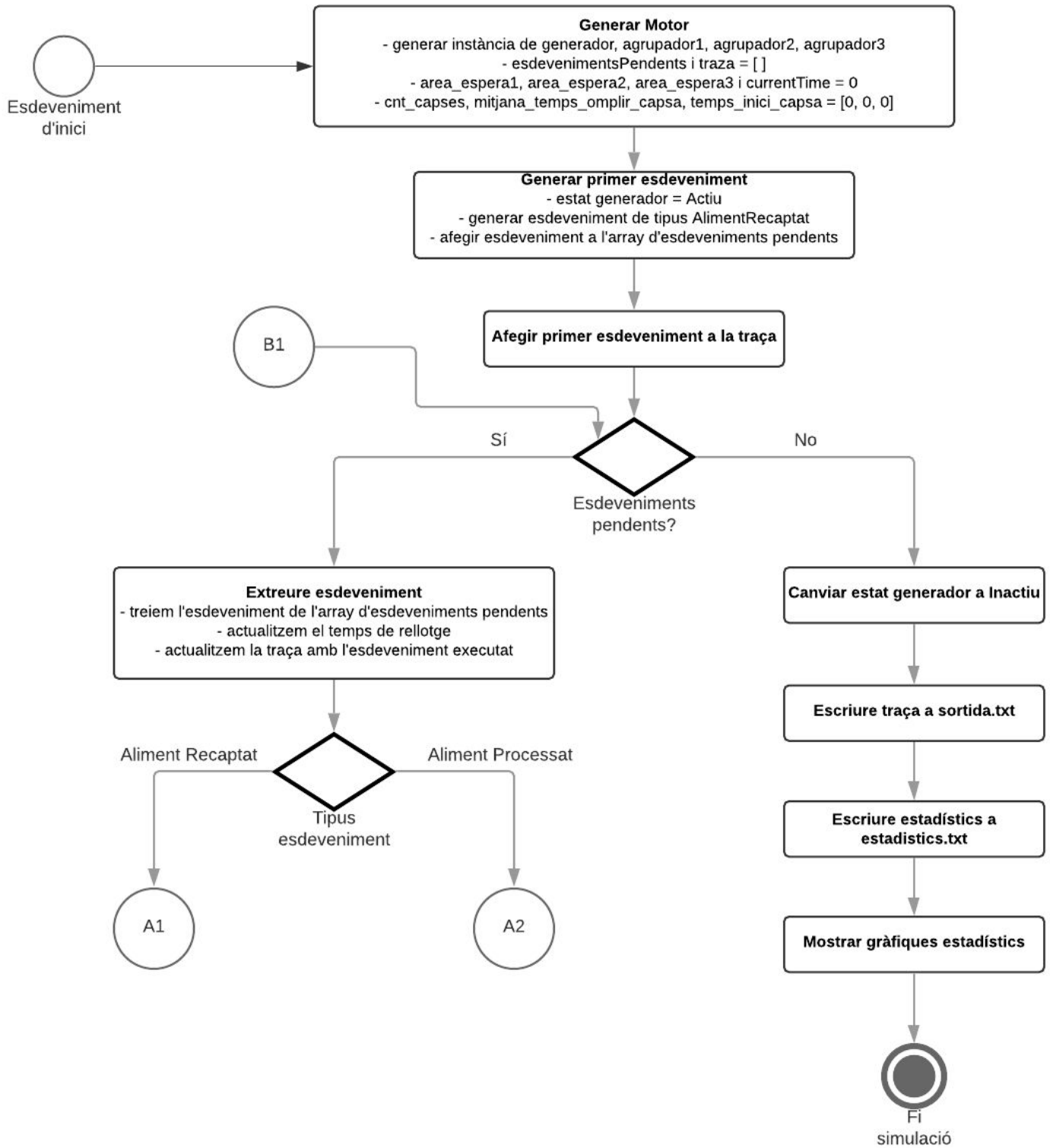


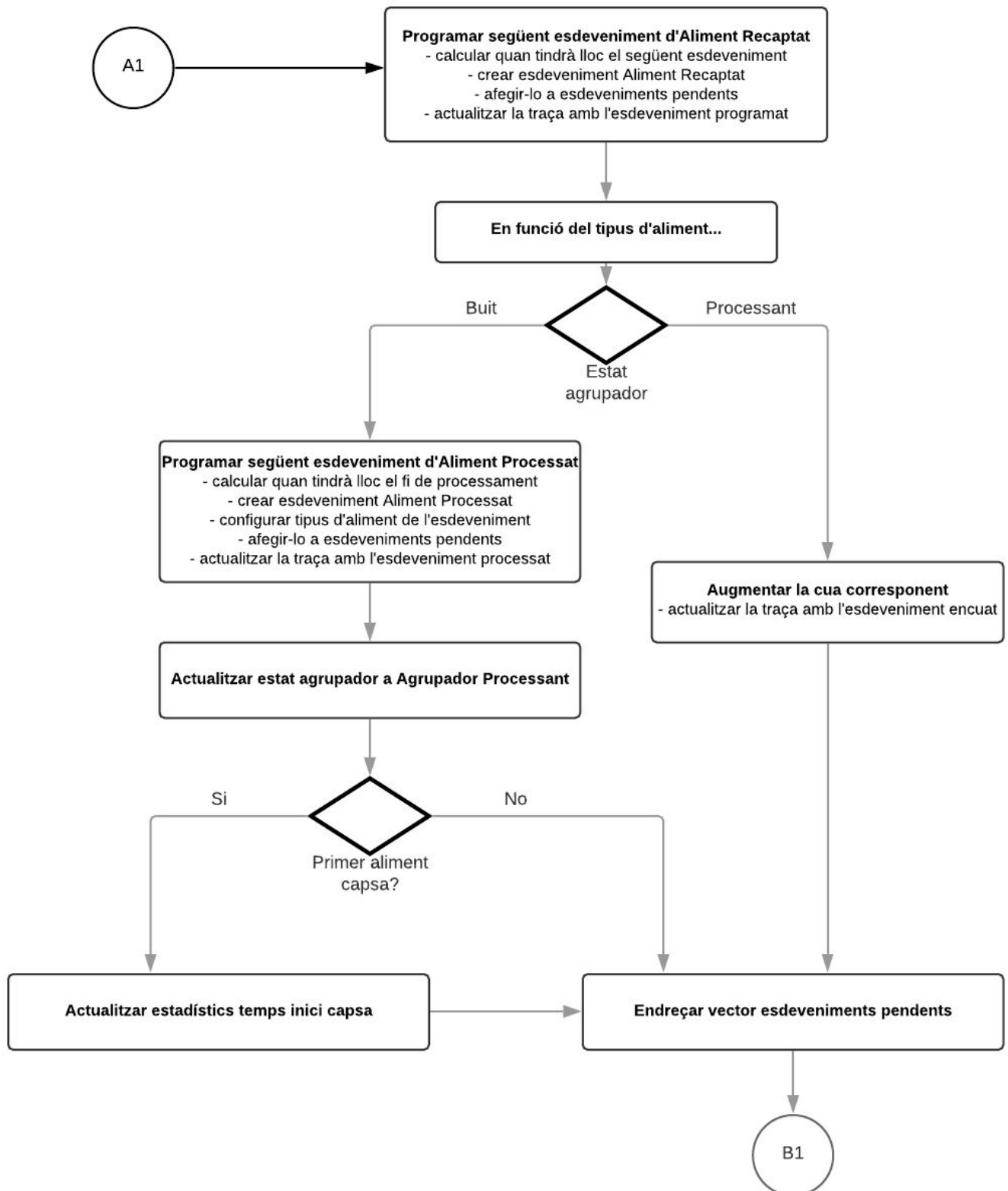
L'aliment és l'entitat que s'utilitza en el nostre sistema, el qual pot estar en 3 estats diferents. Quan és creat pel generador, l'aliment es troba en estat recaptat i està pendent de ser processat. Si l'agrupador corresponent al tipus de l'aliment està processant, l'aliment seguirà en estat recaptat i s'afegirà a la cua. Altrament, l'agrupador el comença a processar i, quan acabi aquest procés, l'aliment passarà a estar en estat processat. Mentre l'agrupador vagi processant aliments i la capsa no estigui plena, els aliments que aquesta conté, continuaran estant en estat processat. Una vegada es completi la capsa, tots ells passaran a estat finalitzat.

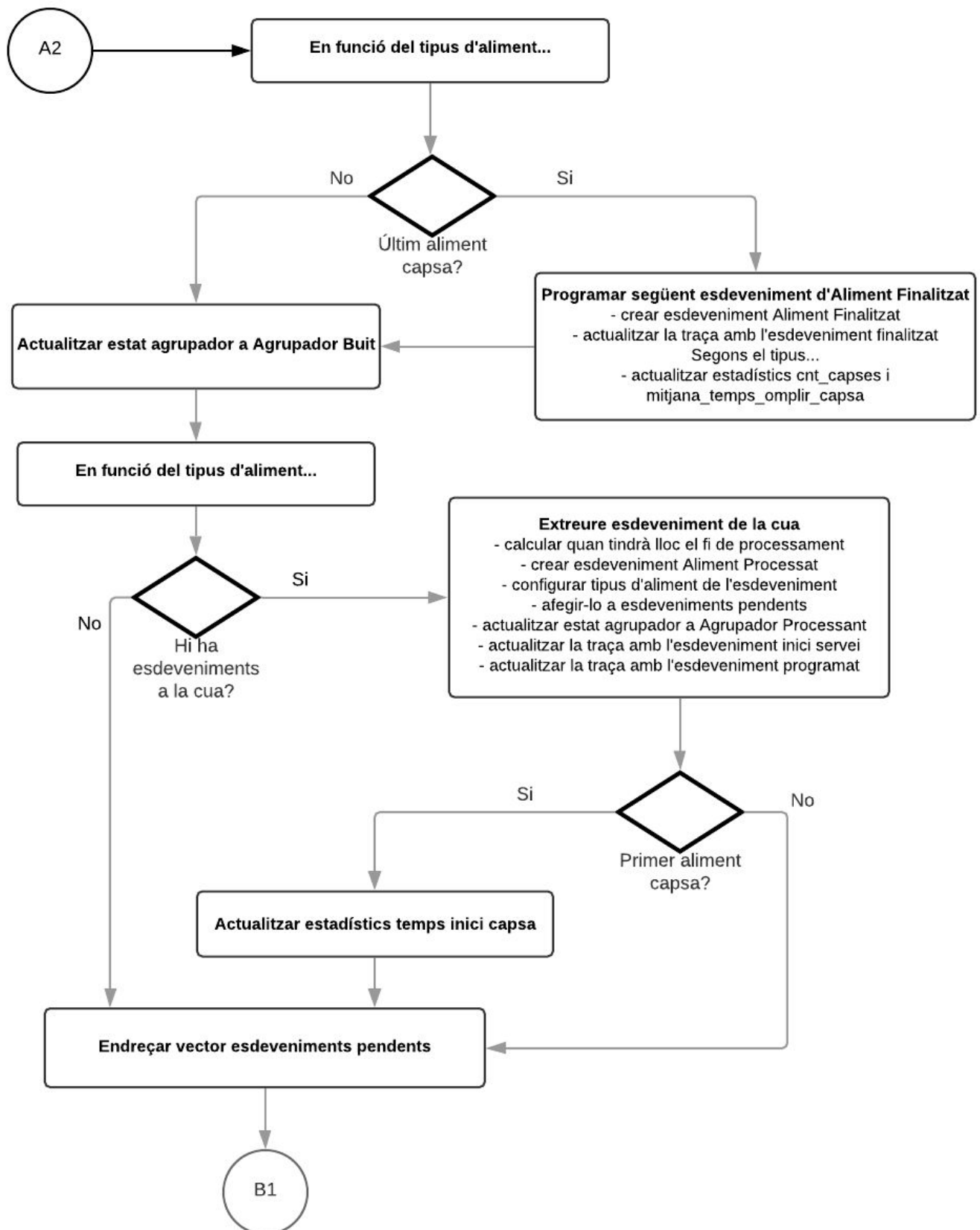
Aliment



Lògica associada al canvi d'estats (BPMN)







Entrada i sortida

Entrada

En el nostre sistema, hem utilitzat el fitxer *configuracio.py* com a fitxer d'entrada. Aquest ens serveix per a definir tots els paràmetres d'entrada del nostre simulador que es poden modificar per a veure canviar els resultats i la traça de la simulació.

Hem considerat necessari incloure els següents:

- *capacitat_capsa*: aquest paràmetre representa els aliments que caben a cada capsa que omple cada agrupador.
- *aliments_a_processar*: aquest paràmetre representa el nombre d'aliments totals a processar a la simulació. Aquest paràmetre, a més, representa la condició de fi de simulació, donat que una vegada s'hagin acabat de processar tots els aliments i no hi hagi cap esdeveniment pendent de processar, la simulació finalitzarà.
- *temps_processament_agrupador*: aquest paràmetre representa el temps que triga un agrupador en processar un aliment per tal d'agrupar-lo a la capsa.
- *loc_temps_entre_arribades*: aquest és un dels paràmetres que utilitzarà el generador per a generar el temps entre arribades seguint una distribució exponencial.
- *scale_temps_entre_arribades*: aquest és un dels paràmetres que utilitzarà el generador per a generar el temps entre arribades seguint una distribució exponencial.

Sortida

Com a resultat de la simulació, es generen dos fitxers amb resultats a més de dues gràfiques per tal de visualitzar i analitzar els estadístics. Un dels fitxers, anomenat *sortida.txt*, conté la traça de cada esdeveniment que s'ha executat durant la simulació, incloent el transcurs dels esdeveniments cronològicament. A l'altre fitxer de text pla, *estadistics.txt*, es mostren les dades obtingudes a partir de les quals es generen les gràfiques de la simulació. Aquests estadístics són les capses generades per cadascun dels 3 agrupadors i la mitjana del temps que cada agrupador triga en omplir una capsa.

Hem considerat que la millor opció per a generar unes gràfiques amb les que després analitzarem els estadístics ha sigut utilitzant la llibreria Matplotlib de Python.

Execució

Manual d'ús

A continuació, us detallarem els passos a seguir per tal de poder executar el nostre simulador.

1. *Configurar els paràmetres d'entrada*: Aquesta configuració es troba al fitxer *configuracio.py*. En aquest, es permet modificar la capacitat de la capsa, els aliments a processar, els temps de processament de l'agrupador i els paràmetres (loc i scale) que limiten els valors de la distribució exponencial que defineix el temps entre arribades.
2. *Executar el simulador*: Per a fer això, només caldrà que s'executi el mètode main del fitxer *main.py*, el qual genera el motor del simulador i el fa córrer.

```
>>> python main.py
```
3. *Visualitzar els resultats*: Aquest pas es farà automàticament un cop es finalitzi la simulació, ja que el propi simulador genera els estadístics i els mostra per pantalla. A la gràfica superior, es podrà observar la quantitat de capsos que cada agrupador ha processat. A la inferior, el temps mitjà que ha trigat cada agrupador en omplir una capsa. Per a veure els estadístics exactes, obre el fitxer *estadistics.txt*. Si es desitja mirar la traça generada per la simulació, només cal obrir el fitxer *sortida.txt*.

Dependències

Per tal de poder executar el simulador correctament, és necessari que es tinguin instal·lats un seguit de paquets.

- Python 3
- Pip
- Numpy
- Matplotlib
- Scipy

Resultats de la simulació

Per tal de realitzar un estudi de la simulació que hem creat i veure i analitzar els seus resultats, hem hagut de configurar els paràmetres d'entrada i ho hem fet amb els següents valors:

- *capacitat_capsa* = 4
- *aliments_a_processar* = 800
- *temps_processament_agrupador* = 3
- *loc_temps_entre_arribades* = 1
- *scale_temps_entre_arribades* = 5

Un cop ja hem definit l'escenari, hem executat la simulació de la qual hem obtingut els següents resultats dels fitxers de *sortida.txt*, *estadistics.txt* i de les gràfiques generades:

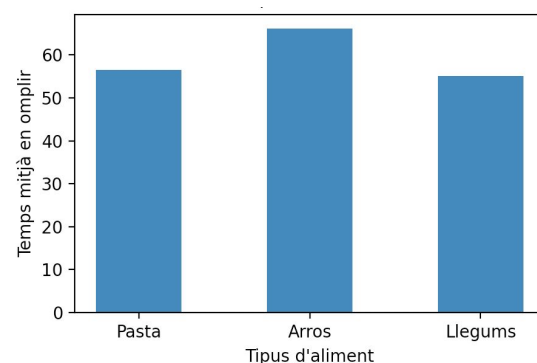
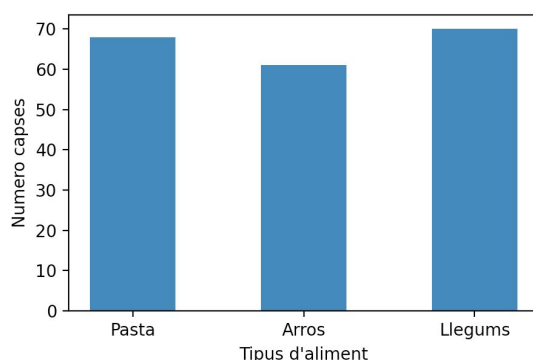
Al fitxer *sortida.txt* és on es van registrant tots els esdeveniments que succeeixen al llarg de l'execució. Com es pot veure, la traça està ordenada cronològicament, indicant, per a cada temps de rellotge, els esdeveniments que o bé tenen lloc o bé són programats.

```

1  L'Agrupador1 agrupa aliments de tipus Pasta
2  L'Agrupador2 agrupa aliments de tipus Arros
3  L'Agrupador3 agrupa aliments de tipus Llegums
4
5  0
6  El Generador programa un esdeveniment AlimentDonat per al temps 4.37
7  4.37
8  El Generador executa un esdeveniment AlimentDonat
9  El Generador programa un esdeveniment AlimentDonat per al temps 5.47
10 Agrupador3 inicia processament
11 L'Agrupador3 programa un esdeveniment AlimentProcessat per al temps 7.37
12 5.47
13 El Generador executa un esdeveniment AlimentDonat
14 El Generador programa un esdeveniment AlimentDonat per al temps 6.73
15 Agrupador2 inicia processament
16 L'Agrupador2 programa un esdeveniment AlimentProcessat per al temps 8.47
17 6.73
18 El Generador executa un esdeveniment AlimentDonat
19 El Generador programa un esdeveniment AlimentDonat per al temps 10.3
20 S'afegeix un esdeveniment a la cua de l'Generador. La cua passa a tenir mida 1
21 7.37
22 L'Agrupador3 executa un esdeveniment AlimentProcessat
23 8.47
24 L'Agrupador2 executa un esdeveniment AlimentProcessat
25 Agrupador2 inicia processament
26 L'Agrupador2 programa un esdeveniment AlimentProcessat per al temps 11.47

```

Finalment, s'obtenen els estadístics de la simulació en dues gràfiques. Aquestes imatges representen el número de capsas que cada agrupador ha omplert per al seu tipus d'aliment i el temps mitjà que ha trigat en omplir una capsa, condicionat únicament pel temps entre arribades. Hem afegit el fitxer *estadistics.txt* per a poder conèixer els valors exactes d'aquests estadístics.



```

1  Capses generades per l'Agrupador1: 68
2  Capses generades per l'Agrupador2: 61
3  Capses generades per l'Agrupador3: 70
4
5  Mitjana del temps en omplir una capsa de l'Agrupador1: 56.51
6  Mitjana del temps en omplir una capsa de l'Agrupador2: 66.11
7  Mitjana del temps en omplir una capsa de l'Agrupador3: 55.04
8

```

Per a confirmar que els estadístics són correctes, només cal veure la correlació entre les dues gràfiques. Com es pot apreciar a les gràfiques, els agrupadors 1 i 3 (pasta i llegums respectivament) han generat més quantitat de capsos d'aliments del seu tipus, cosa que és coherent tenint en compte que, a la segona gràfica, es pot veure com aquests agrupadors tenen un temps mitjà per a omplir una capsa menor a l'agrupador de tipus arròs.