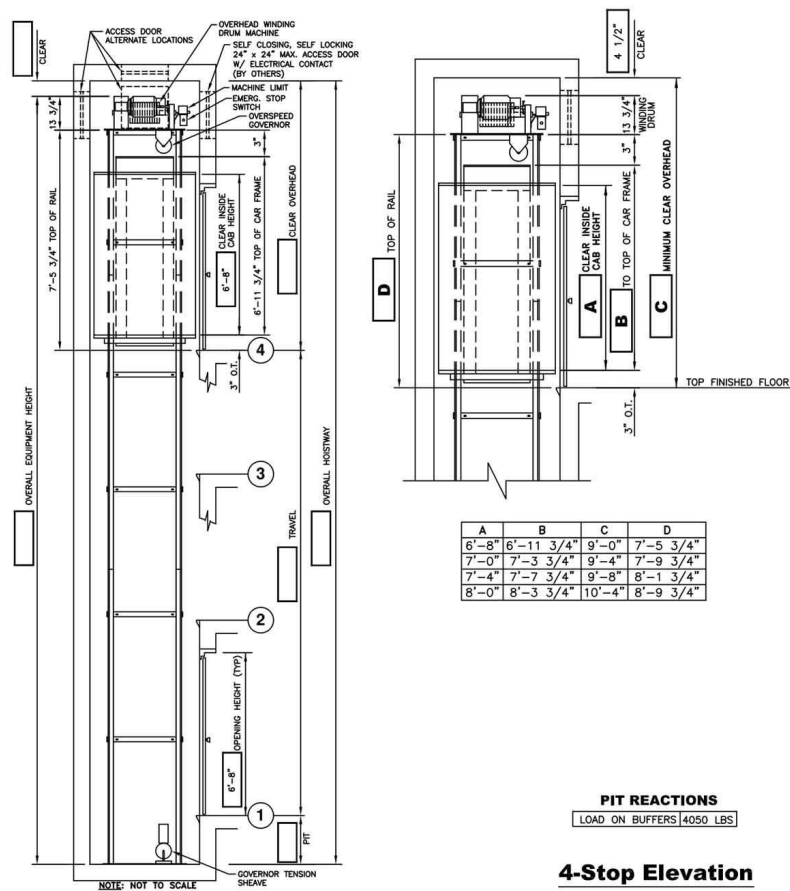


Simulació DES

Estrategia d'Ascensors



Joan Pont Martoris
Hèctor Morales Carnicé

Grup 13
Divendres de 14:00 a 16:00

Simulació DES	0
Descripció del sistema	2
En aquesta entrega hem decidit enfocar-nos en la proposta d'analitzar l'impacte de dues polítiques d'ús en un sistema de quatre ascensors.	2
Suposicions:	2
Especificació del sistema	3
Ascensor	3
Generador	4
Planta	5
Oficina	5
Escala	5
Factory	5
Token (Persona)	6
Entrada del simulador	7
values.yaml	7
Sortida del simulador	8
Execució	9
Sense especificar el fitxer d'entrada	9
Especificant un fitxer d'entrada	9
Dependències i entorn	9

1. Descripció del sistema

En aquesta entrega hem decidit enfocar-nos en la proposta d'analitzar l'impacte de dues polítiques d'ús en un sistema de quatre ascensors.

L'escenari que es planteja, doncs, consta de quatre ascensors que pugen i baixen a través d'un sistema de dotze plantes en total. D'aquesta manera, es vol saber com afecten les següents estratègies en la coordinació dels ascensors i en l'exercici de les múltiples tasques (transportar persones):

- **Alternant:** Dos ascensors es desplacen únicament entre plantes parelles, mentre els altres dos ho fan únicament a plantes senars. Òbviament la planta baixa és comuna en ambdós casos.
- **Express:** Tots els ascensors arriben a totes les plantes, no obstant un dels ascensors és de capacitat reduïda (a la meitat dels demés) però triga menys entre trajectes.

Suposicions:

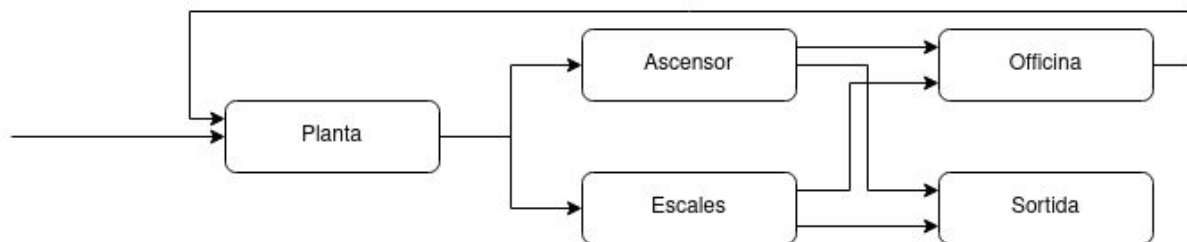
A partir de la definició anterior, nosaltres hem volgut suposar en quina situació podria donar-se el cas d'un sistema així. La raó d'això és conèixer ben bé l'entorn de la simulació per tal de que la implementació final s'ajusti el màxim possible a un cas real.

Així doncs, el nostre model serà el d'un edifici d'oficines on els treballadors hi van a fer la seva jornada laboral, i després se n'en tornem a casa.

Un cop això està clar, podem procedir amb els components que formen aquest model.

2. Especificació del sistema

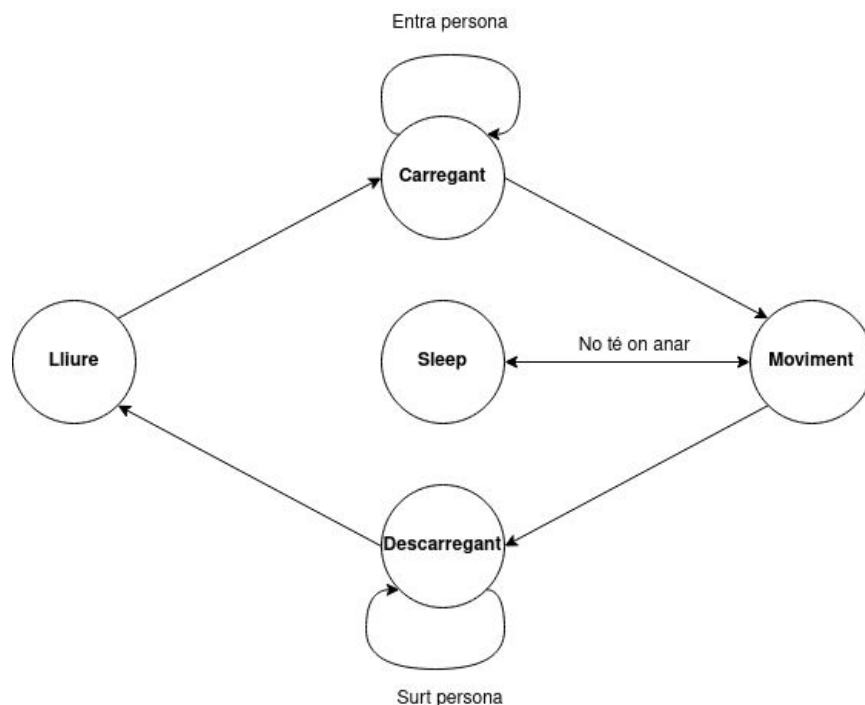
Degut que el nostre sistema simula un entorn laboral que farà ús dels ascensors (l'objecte clau d'aquest model) podem dissenyar de manera molt esquematitzada el següent flux de conducta per part d'un "treballador":



A partir d'aquest diagrama en podem començar a extreure els objectes de la simulació i com s'han de comportar en cada cas:

1. Ascensor

En el nostre model els ascensors, indiferentment de quina estratègia estiguin seguint, satisfan els següents estats:



On **Lliure** és l'estat inicial i serveix per indicar que l'ascensor té places lliures, és a dir, hi pot entrar gent. Si és així, passa a l'estat de **Carregant**, el qual ha de tenir un temps màxim de duració, o bé, finalitzar un cop l'ascensor estigui ple (allò que

succeeixi abans). Un cop acabada la càrrega, el comportament de l'estat **Moviment** estableix dos casos:

- **L'ascensor està buit, i no hi ha cap persona a cap altre planta que esperi per a l'ascensor:** llavors l'ascensor passa a estat **Sleep** on s'hi quedarà per un temps finit de temps abans de saltar novament a **Moviment** esperant tenir on anar.
- **L'ascensor té gent o bé l'esperen a alguna altre planta:** En aquest cas l'ascensor trigarà un temps (segons la distància que hagi de recórrer) fins saltar a l'estat **Descarregant**.

Cal destacar que l'estat **Moviment** és, per tant, on es decideix quin serà el pròxim pis a visitar. L'algorisme que hem definit estableix que: és prioritari conservar el sentit al qual anava inicialment l'ascensor. És a dir, si l'ascensor estava pujant, aquest considerarà preferible anar a deixar o recollir persones dels pisos superiors, que no pas a baixar. Òbviament, si cap dels pisos superiors tenen persones esperant, ni cal dels individus a l'interior de l'ascensor volen anar a cap d'aquells, l'ascensor buscarà el pis en sentit oposat més proper al qual arribar.

Finalment, l'estat **Descarregant** simplement allibera al pis les persones que hi volen arribar, si es que n'hi ha cap. I repeteix el cicle començant novament des de l'estat **Lliure**.

2. Generador

Aprofitant que els ascensors requereixen d'energia per funcionar, i aquest consum es calculable, poden afegir un nou objecte senzill, que no es contemplava a priori, que pot ser el Generador o motor de l'ascensor.

Aquest objecte té per labor computar quanta energia consumeix un ascensor, o el que és el mateix: ens permet saber en quins moments ha funcionat un ascensor en concret. Per fer-ho, el generador té els següents estats:



De tal manera que l'estat **ON** s'activa en l'inici del moviment de l'ascensor, i captura l'hora d'engegada. Un cop l'ascensor para, és a dir, ha acabat el moviment: l'estat **OFF** calcula la diferència entre l'hora d'apagada i la d'engegada per tal de saber el període de funcionament. Afegint-hi altres variables de l'objecte ascensor, com ara: la quantitat de gent que transporta; ens permet ajustar la formula del consum de manera més precisa. No és el mateix aixecar 5 plantes a 2 persones, que a 12. Doncs el temps no s'ha de veure reduït segons el pes.

3. Planta

El comportament de la planta consisteix en distribuir les persones que arriben entre:

- **Cua ascensors:** Si la persona que arriba a la planta no té l'opció de caminar (walker) habilitada: es quedarà esperant fins que arribi un ascensor al qual pugui pujar i que passi per la planta a la qual vol arribar.
- **Escale:** Si la persona que arriba a la planta té l'opció d'anar caminant (walker) habilitada: s'espera a que l'escala tingui espai i un cop la persona te via lliure per usar les escales se li assigna.
- **Sortida:** Si la persona està a la planta destí, i aquesta és la planta 0, només pot ser el cas de que hagi acabat de treballar. Així doncs s'elimina a la persona del sistema a través de l'objecte Factory.
- **Oficina:** Altrament, si la persona està a la planta destí, i aquesta és superior a la planta 0, llavors la persona està anant a treballar. Se l'assigna a l'oficina associada a la planta.

4. Oficina

L'offina correspon a un espai que la persona ocupa en una planta durant la seva jornada laboral. Un cop la persona ha completat aquesta jornada l'oficina expulsa a la persona marcant-li com a destinació la planta baixa. D'aquesta manera l'oficina pot lliurar a la persona a la seva mateixa planta de tal manera que, com ara la destinació ja no coincideix amb la planta actual, el sistema mourà a la persona fins la planta 0.

5. Escales

Les escales en aquest model funcionen com un recurs capaç de retenir a la persona durant un temps finit (segons la distància entre el pis des del qual ha arribat la persona a les escales i el pis al qual vol arribar); i un cop s'ha completat aquesta espera: el propi objecte lliura a la persona a la planta destí. Deixant un recurs a la instància Escala lliure per a alguna altre persona que hi vulgui accedir.

6. Factory

Aquest element és l'encarregat de fer arribar individus a la planta baixa.

Per fer-ho, segueix dues distribucions:

- **Normal:** descriu la quantitat de persones que arriben en cada instant de temps.
- **Uniforme:** descriu el temps entre arribades.

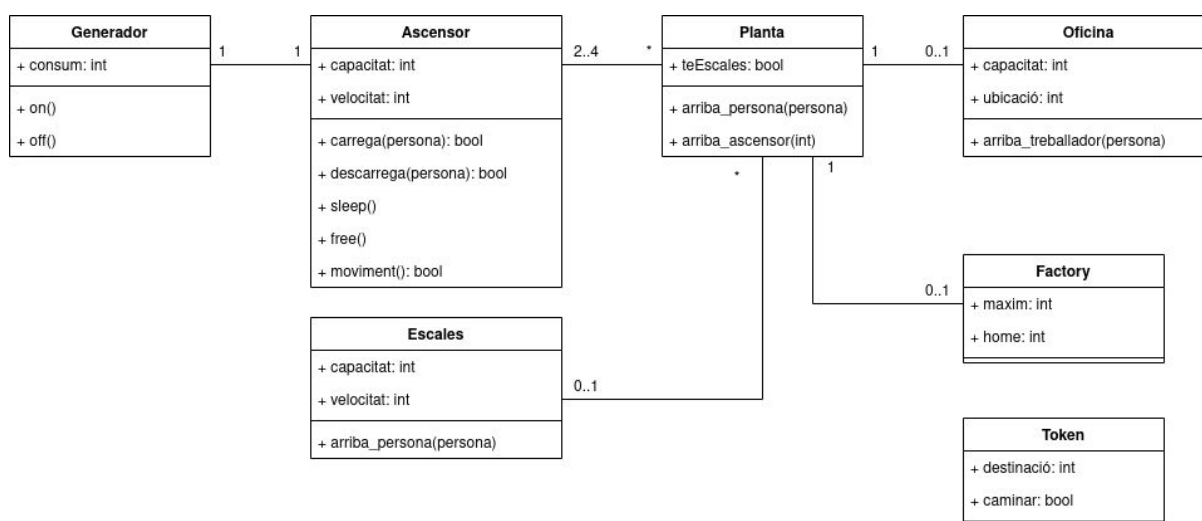
Seguint aquestes dues distribucions l'objecte factoria itera fins exhaurir la capacitat de l'edifici. És a dir, estem parlant de casos en els quals l'assistència a les oficines el total.

7. Token (Persona)

Aquest element és precisament “la persona” a la qual ens hem estat referint durant aquest document. La seva labor es definir tots els comportaments variables de la persona en el moment de la seva creació de tal manera que el sistema sàpiga com adaptar-se a les necessitats d’aquesta. En concret, volem destacar aquells valors que s’assignen utilitzant algun tipus de distribució:

- **oficina:** Defineix en quina planta està l’oficina on treballa la persona. Com aquest valor no hauria de variar gaire entre les plantes, s’utilitza una distribució uniforme per generar-ne els valors de manera pseudoaleatoria.
- **caminar:** Defineix si el treballador prefereix utilitzar les escales que no pas l’ascensor. Degut que aquest comportament és poc habitual (la gent normalment no vol agafar les escales), hem considerat oportú utilitzar un distribució exponencial. Si el valor obtingut arriba a cert llindar, llavors la persona prefereix les escales. En cas contrari només agafa l’ascensor.

Vist tot plegat el model quedaria esquematitzat de la següent manera:



Model conceptual [UML]

3. Entrada del simulador

Per personalitzar les dades d'entrada cal modificar el fitxer **values.yaml** ubicat a la carpeta arrel del projecte.

1. values.yaml

Aquest fitxer, com la seva extensió ja defineix, és de tipus Yaml. En ell s'hi descriuen múltiples paràmetres sobre l'entorn de simulació i com s'han de comportar els seus components.

Per defecte, el fitxer proporciona els següents valors:

```
simulation:
  # 0: totes a l'hora; 1: només alternant; 2: només l'express
  strategy: 0
  seed: 12345 # random seed

  # Interval ininterromput de simulació [ 0 <= hora < 24]
  time:
    from: 7 # hora entera a la que obre l'edifici
    to: 20 # hora entera a la que tanca l'edifici
    work: 8 # temps en hores d'una jornada laboral

  # Sobre com es l'entorn on simular la/les estratègia/es
  environment:
    n_elevator: 4 # total d'ascensors
    n_floors: 12 # total de plantes (SENSE COMPTAR amb la planta baixa; pl=0)
    cap_floor: 90 # capacitat de persones per planta
    stairs: True # L'edifici té escales

  # Sobre com són els ascensors
  elevators:
    capacity: 12 # capacitat d'un ascensor estàndard
    velocity: 10 # temps en segons que triga un ascensor normal entre dues plantes consecutives
    express: 5 # temps en segons que triga l'ascensor express entre dues plantes consecutives
    x_requ: 200 # energia requerida per l'ascensor express
    waiting: 15 # temps d'espera a cada planta
    required: 150 # energia requerida per l'ascensor a cada unitat de temps per persona a desplaçar

  # Sobre com són les escales
  stairs:
    velocity: 30 # temps en segons que triga qualsevol individu en desplaçar-se entre pisos consecutius
    capacity: 50 # capacitat total de persones que poden pujar les escales simultàniament
    range: 0.75 # essent la probabilitat d'agafar les escales una distribució entre 0. i 1., a partir de quing rang es considera que l'agafa

  # Sobre la distribució amb la que arriben
  arrival:
    loc: 10 # mitja d'arribades en un instant de temps
    scale: 5 # variància en les arribades
    upper: 60 # temps en segons màxim entre arribades
    lower: 10 # temps en segons mínim entre arribades

  # Sobre la captura de mètriques
  metrics: 1800 # temps en segons entre captura de mètriques
```

fitxer: `./values.yaml`

Ja que el format YAML permet afegir comentaris, hem considerat una *best practice* afegir la informació sobre que configura cada paràmetre allí mateix. D'aquesta manera no cal consultar dos fitxers simultàniament per entendre que fa cada cosa.

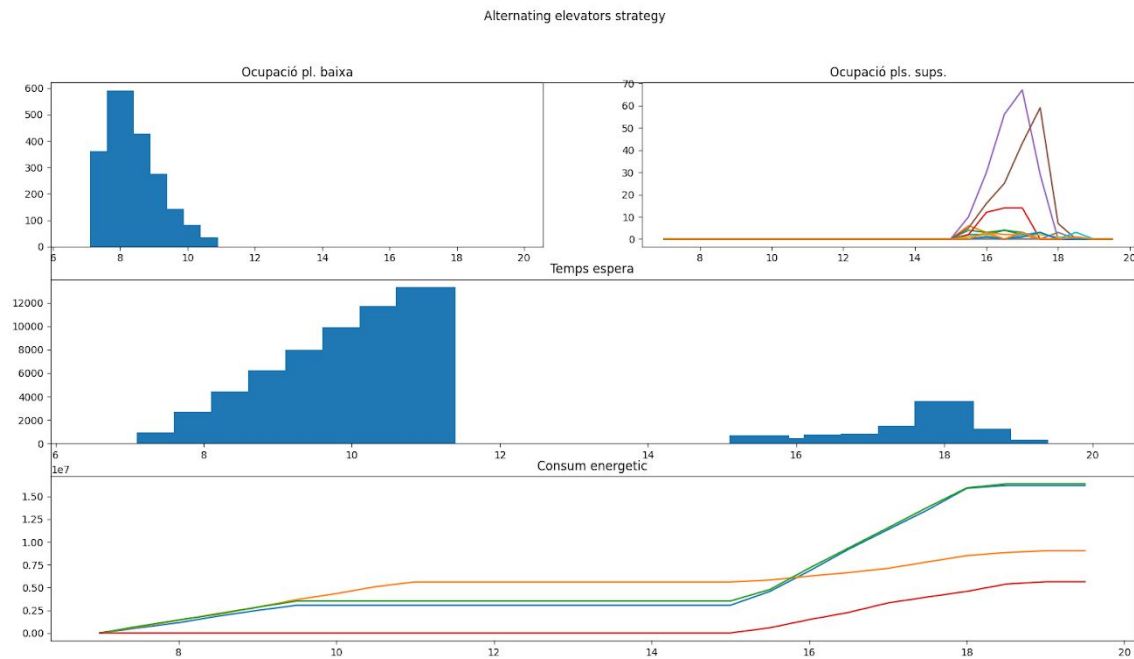
És important que:

- els paràmetres siguin englobats pel label “simulation” a dalt de tot.
- en cap cas es canviï el nom de cap label.
- es conservi la coherència entre els labels i els sagnats de text; ja que aquests sagnats són els que descriuen l'herència entre labels.
- els sagnats es facin tots amb l'ús del tabulador.

4. Sortida del simulador

Com a resultat de l'execució de cada política en el sistema d'ascensors s'obté un conjunt de gràfiques que ens mostres, de manera genèrica, com esdevenen els successos.

Sigui com sigui l'estratègia a executar, un cop acabi s'esperia un plot amb l'estructura següent:

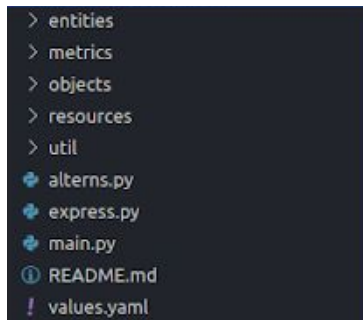


Mètriques obtingudes de l'estratègia d'alternar ascensors

- **Ocupació pl. baixa:** Ens mostra de quina manera s'ha anat ocupant i desocupant la planta baixa a mesura que avançava la simulació.
- **Ocupació pls. sups.:** Ens mostra de quina manera s'han anat ocupant i desocupant la resta de plantes superiors a mesura que avançava la simulació.
- **Temps espera:** Ens mostra la mitja del temps d'espera de totes les persones presents en el sistema a mesura que avança la simulació.
- **Consum energètic:** Ens mostra el consum energètic total fet per cada ascensor al llarg de la simulació.

5. Execució

Tal i com hem implementat el simulador existeixen dues maneres d'executar-lo.



En qualsevol cas, però, es requereix d'un **fitxer values.yaml** que el configuri; així com **ubicar-se a la carpeta arrel** del codi font.

La carpeta arrel té l'estructura que es mostra a la imatge d'aquí a l'esquerra. Com s'hi pot observar, hi ha un values.yaml per defecte que és el que animem a utilitzar.

Les opcions d'execució són les següents:

1. Sense especificar el fitxer d'entrada
`$ python3 ./main.py`
2. Especificant un fitxer d'entrada
`$ python3 ./main.py path/values.yaml`

Un cop executada la comanda s'imprimiran per consola tots els logs d'esdeveniments que vagin sorgint durant la simulació. Un cop finalitzi s'obrirà una nova finestra amb el plot resultant de la simulació.

* El programa no finalitza fins que la finestra on s'hi imprimeixen les mètriques s'hagi tancat.

6. Dependències i entorn

Aquest projecte ha sigut desenvolupat en una **plataforma linux** (Ubuntu 20.04LTS) utilitzant **Python 3.8.2** com a llenguatge de programació. Totes les proves han sigut executades en aquest entorn

Cal tenir instal·lats els següents paquets per al correcte funcionament del simulador:

- [python 3.8.2] sudo apt install python3.8
- [simpy] pip3 install simpy
- [numpy] pip3 install numpy **
- [math] pip3 install math **
- [matplotlib] pip3 install matplotlib
- [yaml] pip3 install yaml

** Dependència associada a la generació de nombres i variables aleatòries.