

un estudio de TAD's implementando los TAD

Pila

Usamos como tipo base el tipo `-class- list` de python. Aporta mucho trabajo hecho. (esto será un wrapper y poco mas)

diseño

datos: la pila es generica. (apila cualquier dato,, tipo: object)
operaciones:

- ☒ `push()`
- ☒ `pop()`
- ☒ `h()` // altura
- ☒ `drop()` // descarte
- ☒ `esVacia()`
- ☒ `esLlena()`

Cola

Para hacer una cola con tamaño máximo Nos inspiramos en este material

List is a Python's built-in data structure that can be used as a queue. Instead of `enqueue()` and `dequeue()`, `append()` and `pop()` function is used. However, lists are **quite slow** for this purpose because inserting or deleting an element at the beginning requires shifting all of the other elements by one: a task in $O(n)$ time.

objetivo:

un cliente (manejador) que use objetos `micola` tal y como usa objetos `mipila`: transparentemente (y aunque los resultados serán distintos)

v0: sin usar `collections.deque`;

- ☐ `enqueue()` // append
- ☐ `dequeue()` //
- ☐ `l()` // length
- ☐ `drop()` // descarte
- ☐ `esVacia()`
- ☐ `esLlena()`