

# Llistes: especificació

3 d'abril de 2015

```
template <class T> class list {

// Tipus de mòdul: dades

// Descripció del tipus: Estructura lineal que conté elements de tipus T, que es
// pot començar a consultar pels extrems, on des de cada element es pot accedir
// a l'element anterior i posterior (si existeixen), i que admet afegir-hi
// i esborrar-hi elements a qualsevol punt

private:

public:

// Constructores

list();
/* Pre: cert */
/* Post: El resultat és una llista sense cap element */

list(const list & original);
/* Pre: cert */
/* Post: El resultat és una llista còpia d'original */

// Destructora: Esborra automàticament els objectes locals en sortir d'un àmbit
// de visibilitat

~list();

// Modificadores
```

```

void clear();
/* Pre: cert */
/* Post: El paràmetre implícit és una llista buida */

void insert(iterator it, const T& x);
/* Pre: it referencia algun element existent al paràmetre implícit o
és igual al end() d'aquest */
/* Post: El paràmetre implícit és com el paràmetre implícit original amb x
davant de l'element referenciat per it al paràmetre implícit original */

iterator erase(iterator it);
/* Pre: it referencia algun element existent al paràmetre implícit,
que no és buit */
/* Post: El paràmetre implícit és com el paràmetre implícit original sense
l'element referenciat per l'it original; el resultat referencia l'element
següent al que referenciava it al paràmetre implícit original */

void splice(iterator it, list& l);
/* Pre: l=L, it referencia algun element del paràmetre implícit o
és igual a l'end() d'aquest; el p.i. i l no són el mateix objecte */
/* Post: El paràmetre implícit té els seus elements originals i els
d'l inserits davant l'element referenciar per it; l està buida */

// Consultores

bool empty() const;
/* Pre: cert */
/* Post: El resultat indica si el paràmetre implícit te elements o no */

int size() const;
/* Pre: cert */
/* Post: El resultat és el nombre d'elements del paràmetre implícit */

};

```