

WUOLAH



Phantone

www.wuolah.com/student/Phantone



716

ED-T3-4.pdf

Apuntes Teoría



2º Estructuras de Datos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



TEMA 3.4: COLAS CON PRIORIDAD

Una cola con prioridad es una ED lineal diseñada para realizar accesos y borrados en uno de sus extremos (frente). Las inserciones se realizan en cualquier posición, de acuerdo a un valor de prioridad.



Operaciones básicas

- Frente: devuelve el elemento del frente.
- Prioridad_Frente: devuelve la prioridad del elemento del frente.
- Poner: añade un elemento con una prioridad asociada.
- Quitar: elimina el elemento del frente.
- Vacía: indica si la cola está vacía.

Esquema de la interfaz

```
#ifndef __COLA_PRI__
#define __COLA_PRI__

class ColaPri{
private:
    ... //La implementación que se elija
public:
    ColaPri();
    ColaPri(const ColaPri& c);
    ~ColaPri();
    ColaPri& operator=(const ColaPri& c);

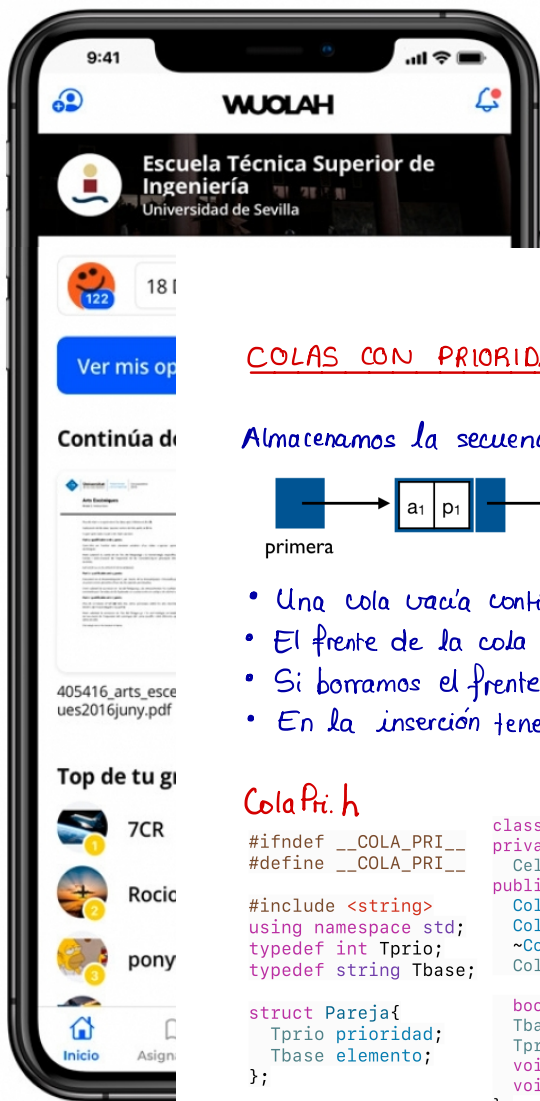
    bool vacia() const;
    Tbase frente() const;
    Tprio prioridad_frente() const;
    void poner(Tbase e, Tprio prio);
    void quitar();
};

#endif /* ColaPri_hpp */
```

Uso de una cola

```
#include <iostream>
#include "ColaPri.hpp"
using namespace std;
int main(){
    ColaPri c;
    int nota;
    string dni;

    cout << "Escriba una nota: ";
    cin >> nota;
    while(nota >=0 && nota <=10){
        cout << "Escriba un dni: ";
        cin >> dni;
        c.poner(dni, nota);
        cout << "Escriba una nota: ";
        cin >> nota;
    }
    cout << "DNIs ordenados por nota:" << endl;
    while(!c.vacia()){
        cout << "DNI: " << c.frente() << " Nota: "
            << c.prioridad_frente() << endl;
        c.quitar();
    }
    return 0;
}
```

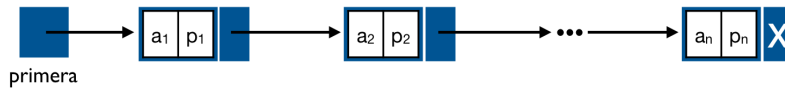


Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



COLAS CON PRIORIDAD: IMPL. CELDAS ENLAZADAS

Almacenamos la secuencia de parejas en celdas enlazadas.



- Una cola vacía contiene un puntero nulo.
- El frente de la cola está en la primera celda (eficiente).
- Si borramos el frente, eliminamos la primera celda.
- En la inserción tenemos que buscar la posición según su prioridad.

ColaPri.h

```
#ifndef __COLA_PRI__
#define __COLA_PRI__

#include <string>
using namespace std;
typedef int Tprio;
typedef string Tbase;

struct Pareja{
    Tprio prioridad;
    Tbase elemento;
};

struct CeldaColaPri{
    Pareja dato;
    CeldaColaPri* sig;
};

class ColaPri{
private:
    CeldaColaPri* primera;
public:
    ColaPri();
    ColaPri(const ColaPri& c);
    ~ColaPri();
    ColaPri& operator=(const ColaPri& c);

    bool vacia() const;
    Tbase frente() const;
    Tprio prioridad_frente() const;
    void poner(Tbase e, Tprio prio);
    void quitar();
};

#endif /* ColaPri.hpp */
```

ColaPri.cpp

```
#include <cassert>
#include <utility>
#include "ColaPri.hpp"

ColaPri::ColaPri(){
    primera = 0;
}

ColaPri::ColaPri(const ColaPri& c){
    if(c.primera==0)
        primera = 0;
    else{
        primera = new CeldaColaPri;
        primera->dato = c.primera->dato;
        CeldaColaPri* src = c.primera;
        CeldaColaPri* dest = primera;
        while(src->sig!=0){
            dest->sig = new CeldaColaPri;
            src = src->sig;
            dest = dest->sig;
            dest->dato = src->dato;
        }
        dest->sig = 0;
    }
}

ColaPri::~ColaPri(){
    CeldaColaPri* aux;
    while(primera != 0){
        aux = primera;
        primera = primera->sig;
        delete aux;
    }
}

ColaPri& ColaPri::operator=(const ColaPri& c){
    ColaPri colatemp(c);
    swap(this->primera, colatemp.primera);
    return *this;
}

bool ColaPri::vacia() const{
    return (primera==0);
}

Tbase ColaPri::frente()const{
    assert(primera!=0);
    return (primera->dato.elemento);
}

Tprio ColaPri::prioridad_frente() const{
    assert(primera!=0);
    return(primera->dato.prioridad);
}

void ColaPri::quitar(){
    assert(primera!=0);
    CeldaColaPri* aux = primera;
    primera = primera->sig;
    delete aux;
}

void ColaPri::poner(Tbase e, Tprio prio){
    CeldaColaPri* aux = new CeldaColaPri;
    aux->dato.elemento = e;
    aux->dato.prioridad = prio;
    aux->sig = 0;
    if (primera==0)
        primera = aux;
    else if(primera->dato.prioridad<prio){
        aux->sig = primera;
        primera = aux;
    }
    else{
        CeldaColaPri* p = primera;
        while(p->sig!=0){
            if(p->sig->dato.prioridad<prio){
                aux->sig = p->sig;
                p->sig = aux;
                return;
            }
            else p = p->sig;
        }
        p->sig = aux;
    }
}
```

WUOLAH