# Implementando una lista de caracteres

Carlos E. Alvarez<sup>1</sup>.

<sup>1</sup>Dep. de Matemáticas aplicadas y Ciencias de la Computación, Universidad del Rosario

2019-II







#### Estructura Cell

Estructura que agrupa un elemento con una referencia a otra estructura del mismo tipo.

```
struct Cell {
  char ch;
  Cell *next;
};
```

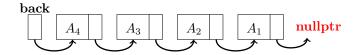
## Operador ->

Permite el acceso a los miembros de una estructura o clase desde un apuntador.

```
Ejemplo
 int main() {
   Cell *mycell; //pointer to cell
   mycell = new Cell; //create cell and point to it
   //initialize cell
   mycell->ch = 'A';
   mycell->next = nullptr;
   cout << nycell->ch << endl;
   delete mycell; //free memory
   return 0;
```

I gos ososa Ciencias de la Computació

### Lista enlazada





```
Interfase (charlist.hpp)
 class CharList. {
 private:
   struct Cell {
    char ch;
     Cell *next;
   };
   Cell *back; //pointer to end of list
 public:
   CharList() : back(nullptr) {}
   ~CharList();
   bool empty();
   void push_back(char c); //create element at end of list
   void pop_back(); //delete element at end of list
 };
```

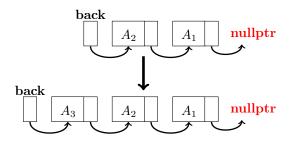
Método empty: Retorna true si la lista está vacía o false en caso contrario.

- 1. Implemente el método empty
- 2. Comente el destructor en la interfase. Cree una CharList y pruebe que el método empty funcione





Método push\_back: Añade un elemento Cell al final de la lista.



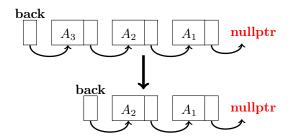


- 1. Implemente el método push\_back. No olvide el caso especial cuando la lista está vacía
- 2. Añada a la implementación una línea que imprima: "pushing" + el elemento que está añadiendo
- 3. Comente el destructor en la interfase. Cree una CharList y pruebe que el método push back funcione





Método pop\_back: Elimina el elemento al final de la lista.







- 1. Implemente el método pop back
- 2. Añada a la implementación una línea que imprima: "popping" + el elemento que está borrando
- 3. Comente el destructor en la interfase. Cree una CharList, añada varios elementos y pruebe que el método pop\_back funcione. Úselo en un ciclo while hasta que la lista quede vacía

Método destructor: Debe asegurarse de liberar toda la memoria que la estructura tenga reservada.

- 1. Implemente el método destructor de la clase CharList
- 2. Cree una CharList y asegúrese que el destructor cumple su labor al salir de la función main



