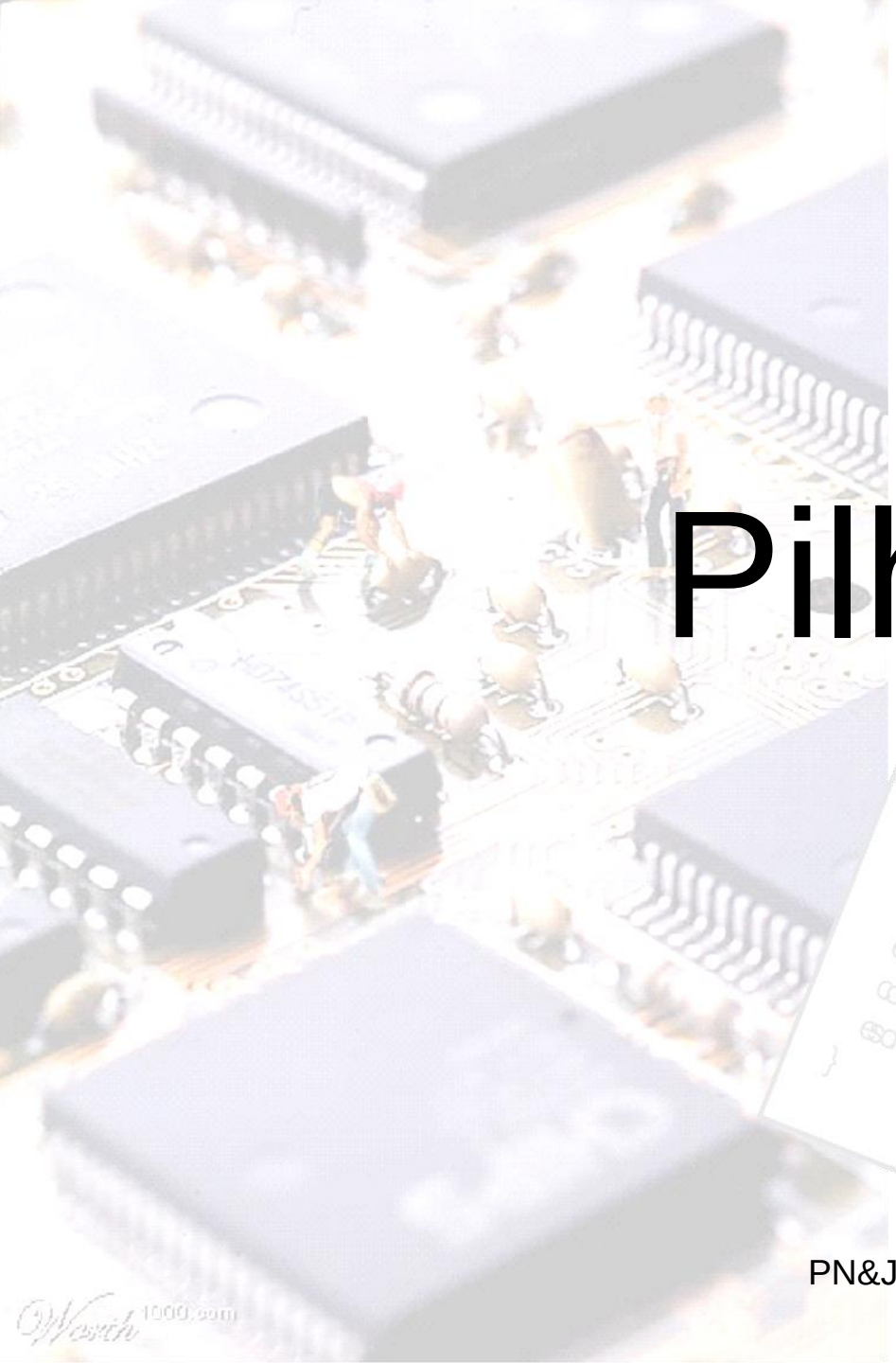
Pilhas e Filas

PN & JFig

2004



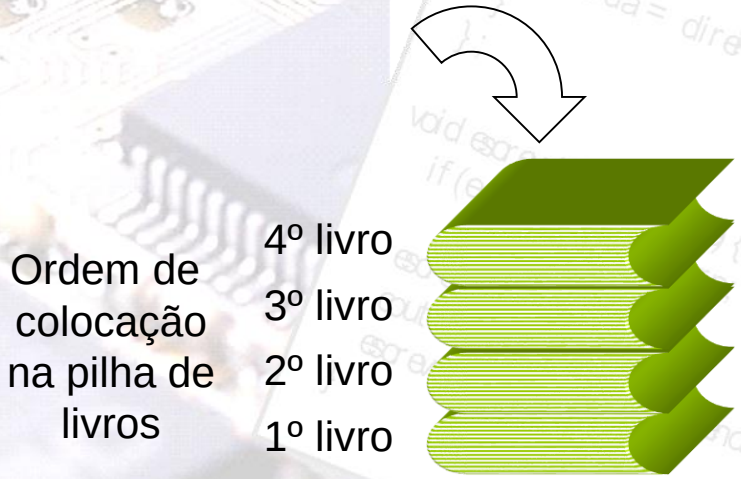
Pilhas

Pilhas - definição

- Uma pilha é uma estrutura linear de dados que pode ser acedida unicamente de uma das extremidades.

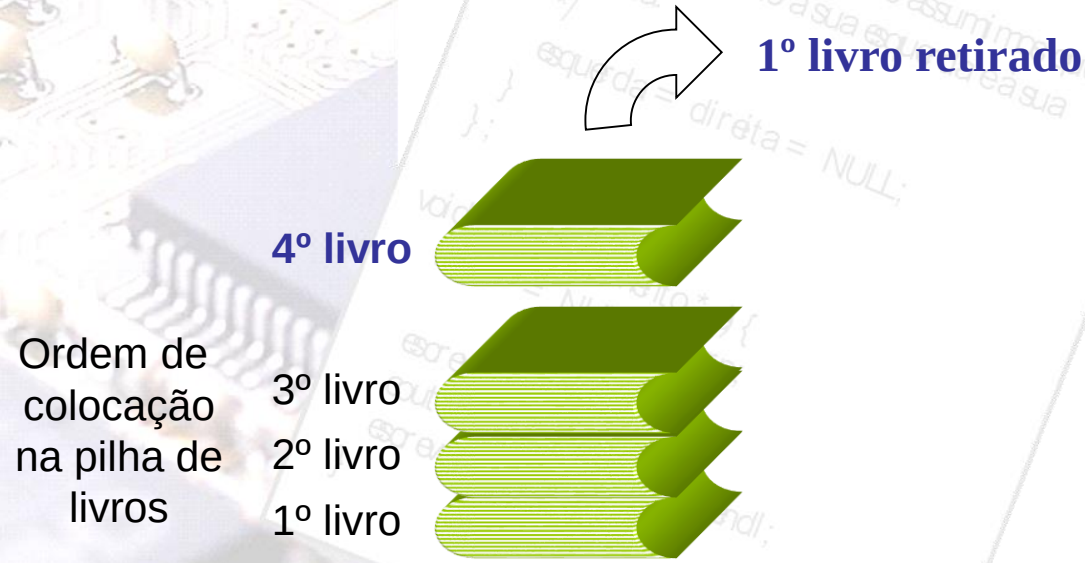
Pilhas

- Podemos ver uma pilha como um conjunto de livros, colocados uns em cima dos outros.



Pilhas

- Para retirar os livros começamos pelo último livro colocado na pilha.



Pilhas

- Por este motivo, a pilha é designada por uma estrutura do tipo LIFO (**L**ast-**I**n-**F**irst-**O**ut).

Pilhas

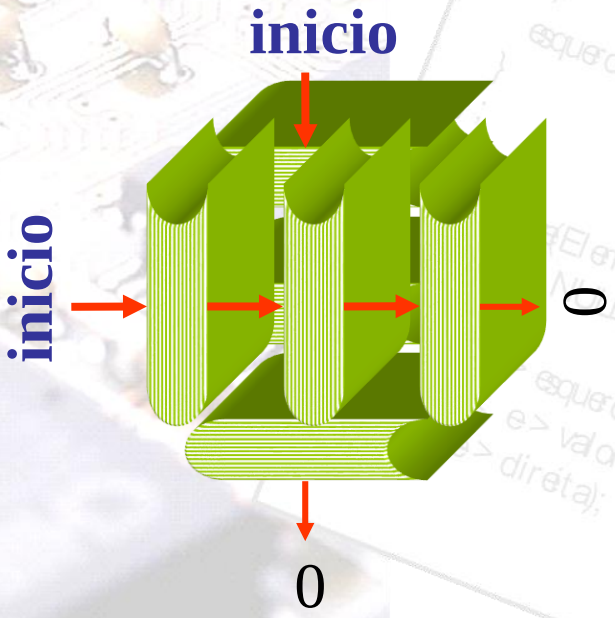
- As operações normalmente definidas para manipulação de uma pilha são:
  - **clear()** – eliminar todos os elementos da pilha;
  - **isEmpty()** – verificar se a pilha está vazia;
  - **push(elemento)** – colocar um elemento no topo da pilha;
  - **pop()** – retirar o elemento mais alto da pilha;
  - **topEl()** – devolve o elemento da mais alto da pilha, sem o eliminar;

Pilhas

- Partindo da definição existente para a lista simples e utilizando as vantagens da utilização de classes, facilmente chegamos à implementação da class Pilha.

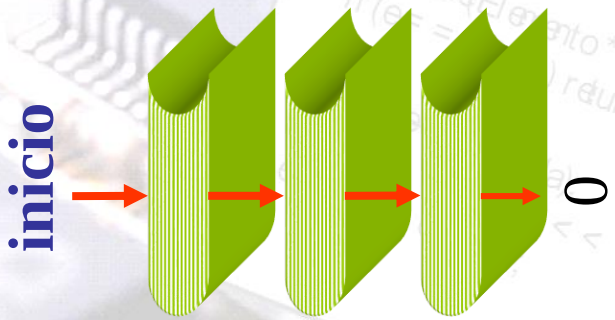
Pilhas

- A pilha não é mais que uma implementação especifica da lista.



Pilhas: implementação

- No caso especifico da pilha, vamos utilizar os métodos *inserirInicio()* e *eliminarInicio()*, implementados na classe Lista, para a manipulação da pilha.



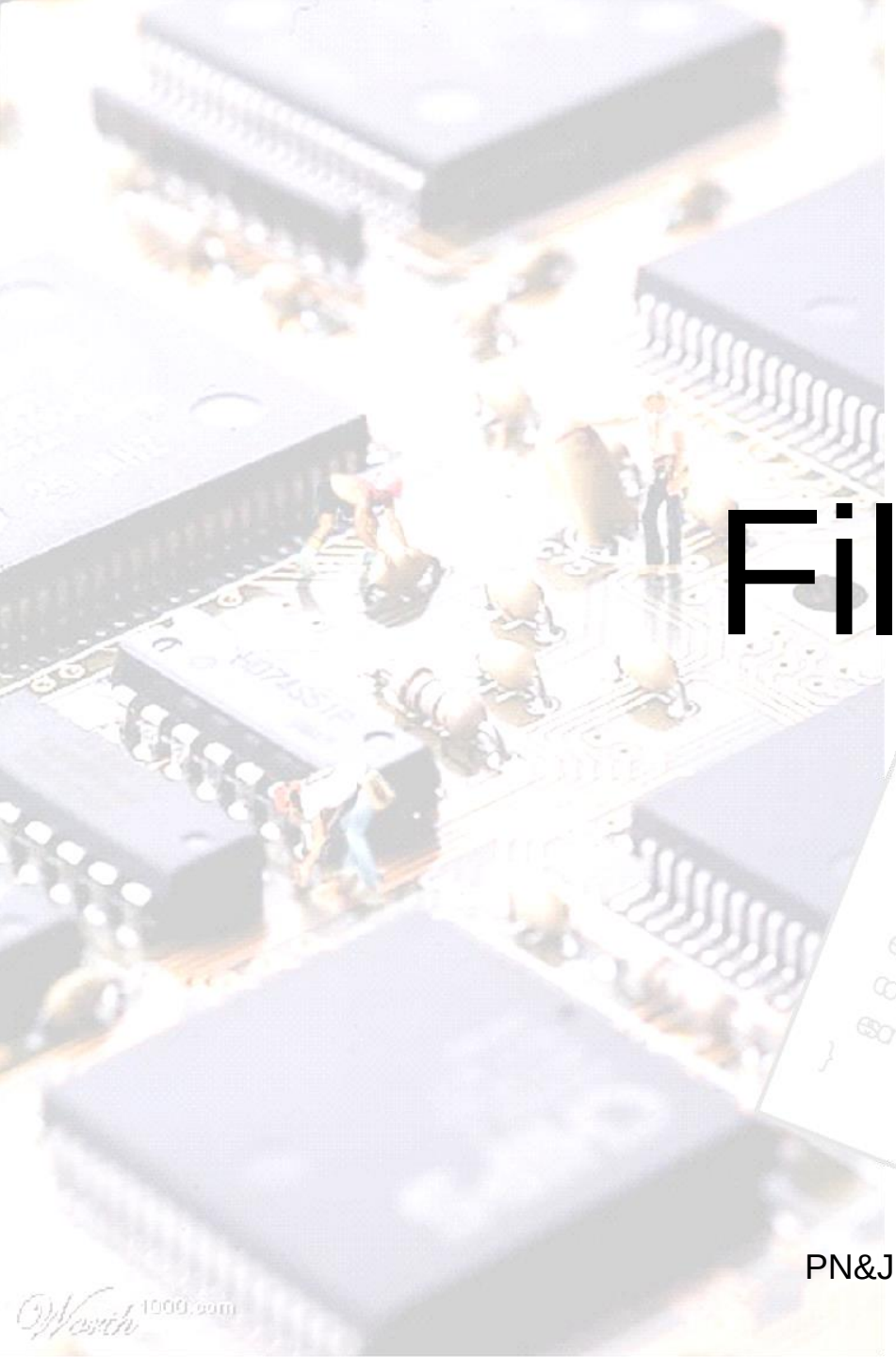
Pilhas: implementação

```
class Lista{
  private:
     Elemento * inicio;
  public:
     Lista(){ inicio = 0; }
     Lista(int valor)
     void inserirInicio(int valor);
     void inserirFim(int valor);
     int eliminaInicio();
     int eliminaFim();
     string verTudo();
     bool vazia();
};
```

```
class TPilha : protected Lista {
   public:
     TPilha() { Lista(); }
     void push(int e);
     int pop();
     string verTudo();
     bool isEmpty();
     void clear();
};
```

Pilhas: class TPilha

```
void TPilha::push(int e){
   inserirInicio(e);
}
int TPilha::pop(){
   return eliminaInicio();
}
bool isEmpty(){ return vazia(); }
void clear() { limpar(); }
```



Filas

Filas: definição

- Fila é uma linha de espera que cresce adicionando elementos no seu final e que diminui retirando elementos do seu inicio.

Filas

- Para melhor entendermos as filas, consideremos uma fila de pessoas. Cada vez que chega uma pessoa à fila, esta fica atrás das outras que já lá estavam. A primeira pessoa a sair da fila é sempre a que chegou primeiro, depois a que chegou em segundo lugar e assim sucessivamente.

Filas

- Uma fila é designada por uma estrutura do tipo FIFO (**F**irst-**I**n-**F**irst-**O**ut). Ou seja, o primeiro elemento a entrar na fila é o primeiro a sair.

Filas: operações frequentes

- As operações mais frequentes para manipulação de uma fila são:
  - **clear()** – eliminar todos os elementos da fila;
  - **isEmpty()** – verificar se a fila está vazia;
  - **enqueue(elemento)** – colocar um elemento no final da final;
  - **dequeue()** – retira o primeiro elemento da fila;

Filas: implementação

- Da mesma forma que as pilhas, e tirando partido das características das classes, vamos utilizar a class Lista para implementar a class TFila.
- As filas não são mais que uma implementação especifica das listas.

Pilhas: implementação

```
class Lista{
  private:
     Elemento * inicio;
  public:
     Lista(){ inicio = 0; }
     Lista(int valor)
     void inserirInicio(int valor);
     void inserirFim(int valor);
     int eliminaInicio();
     int eliminaFim();
     string verTudo();
     bool vazia();
};
```

```
class TFila : protected Lista {
   public:
     TFila() { Lista(); }
     void enqueue(int e);
     int dequeue();
     string verTudo();
     bool isEmpty();
     void clear();
};
```

Filas: class TFila

```
void TFila::enqueue(int e){
   inserirFim(e);
}
int TFila::dequeue(){
   return eliminaInicio();
};
bool isEmpty(){ return vazia() };
void clear(){ limpa() };
```