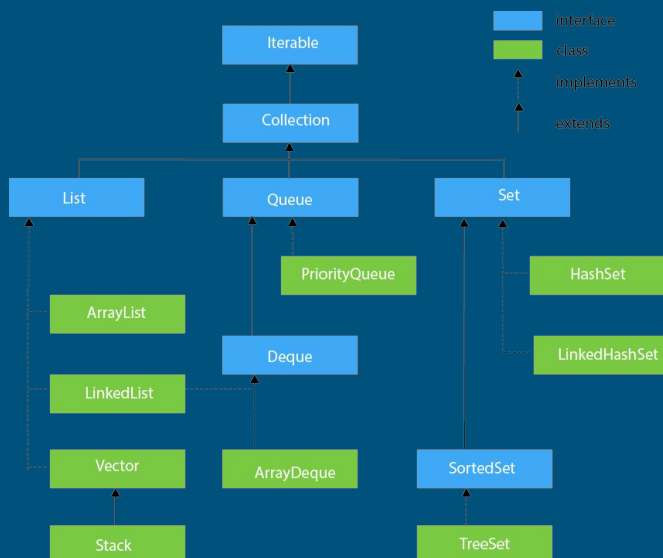


# Coleções



emerson@paduan.pro.br

## Classes para ED em Java



emerson@paduan.pro.br

# ArrayList

`ArrayList<T>` (pacote `java.util`) pode alterar dinamicamente seu tamanho para acomodar mais elementos.

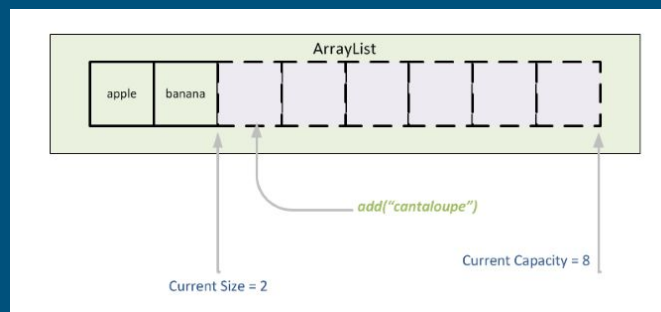
- `T` indica o tipo de elemento armazenado na coleção
- Isso é semelhante a especificar o tipo ao declarar um array, exceto que apenas tipos não-primitivos podem ser utilizados com essas classes de coleção.

Classes com essa espécie de marcador de lugar são chamadas classes genéricas.

emerson@paduan.pro.br

# ArrayList

Coleção em Java que permite armazenar elementos de forma DINÂMICA.



emerson@paduan.pro.br

# ArrayList

Exemplo:

```
import java.util.ArrayList;

public class Exemplo {
    public static void main(String args[])
    {
        ArrayList<String> nomes = new ArrayList<>();

        nomes.add("Huguinho");
        nomes.add("Zezinho");
        nomes.add("Luizinho");

        System.out.println(nomes);
    }
}
```

emerson@paduan.pro.br

## Principais métodos

Método	Descrição
add(Object o)	Adiciona um elemento ao fim do ArrayList
add(int index, Object o)	Adiciona um elemento no índice especificado do ArrayList
clear()	Remove todos os elementos do ArrayList
get(int index)	Retorna o elemento do índice especificado
indexOf(Object o)	Retorna o índice da primeira ocorrência do elemento especificado no ArrayList
remove(Object o)	Remove a primeira ocorrência do valor especificado
remove(int index)	Remove o elemento do índice especificado
size()	Retorna o número de elementos armazenados no ArrayList
isEmpty()	Retorna true se não existem elementos no ArrayList

emerson@paduan.pro.br

# foreach

foreach – (para cada)

iterar sobre coleções de maneira simples e direta

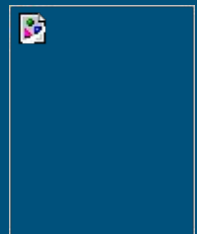
Sintaxe:

```
for( tipo variavel : nomeArray ){  
    //corpo do for  
}
```

emerson@paduan.pro.br

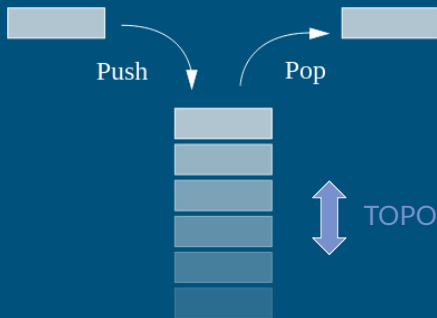
# Pilha

- ✓ Conjunto ordenado de itens no qual somente em uma das extremidades novos itens podem ser inseridos, ou itens podem ser removidos.
- ✓ A extremidade onde os itens são inseridos ou removidos chama-se **Topo** da pilha.
- ✓ **FILO** – First In, Last Out : O primeiro a entrar será o último a sair.
- ✓ **LIFO** – Last In, First Out : O último a entrar será o primeiro a sair



emerson@paduan.pro.br

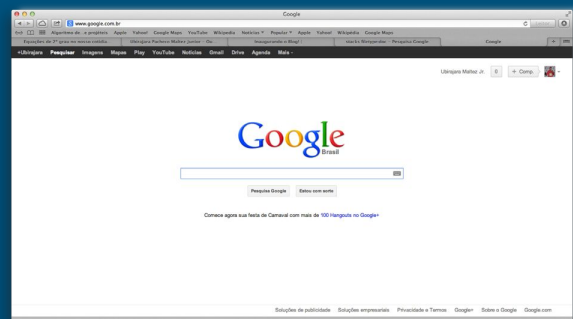
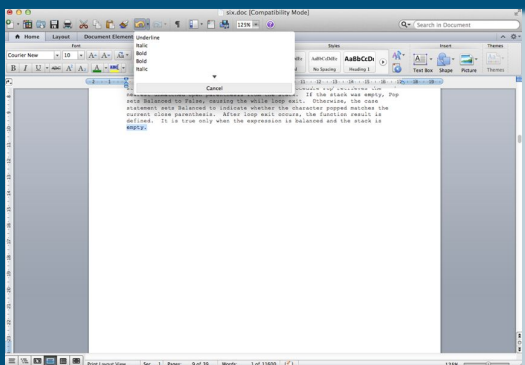
# Operações



- ✓ Pilha vazia → ***isEmpty()***
- ✓ Inserir (empilhar) → ***push()***
- ✓ Remover (desempilhar) → ***pop()***
- ✓ Informa o elemento no topo → ***peek()***
- ✓ Quantidade de elementos → ***size()***

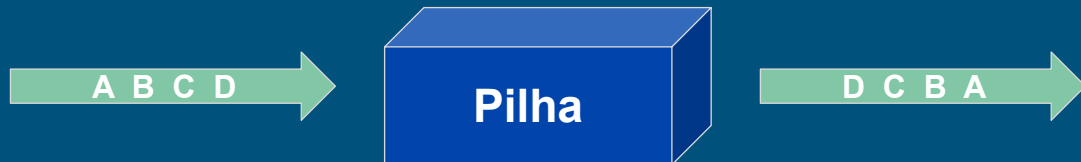
emerson@paduan.pro.br

# Aplicações



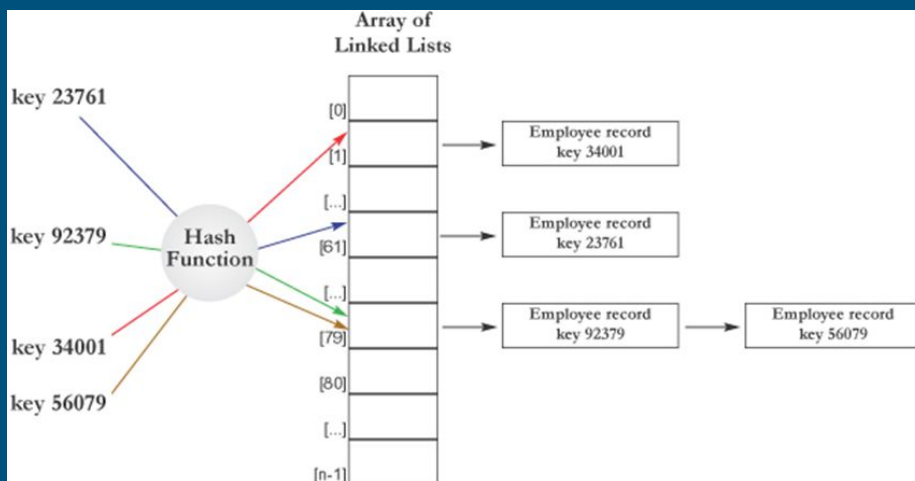
emerson@paduan.pro.br

# Utilizando uma pilha



emerson@paduan.pro.br

# Hash



emerson@paduan.pro.br