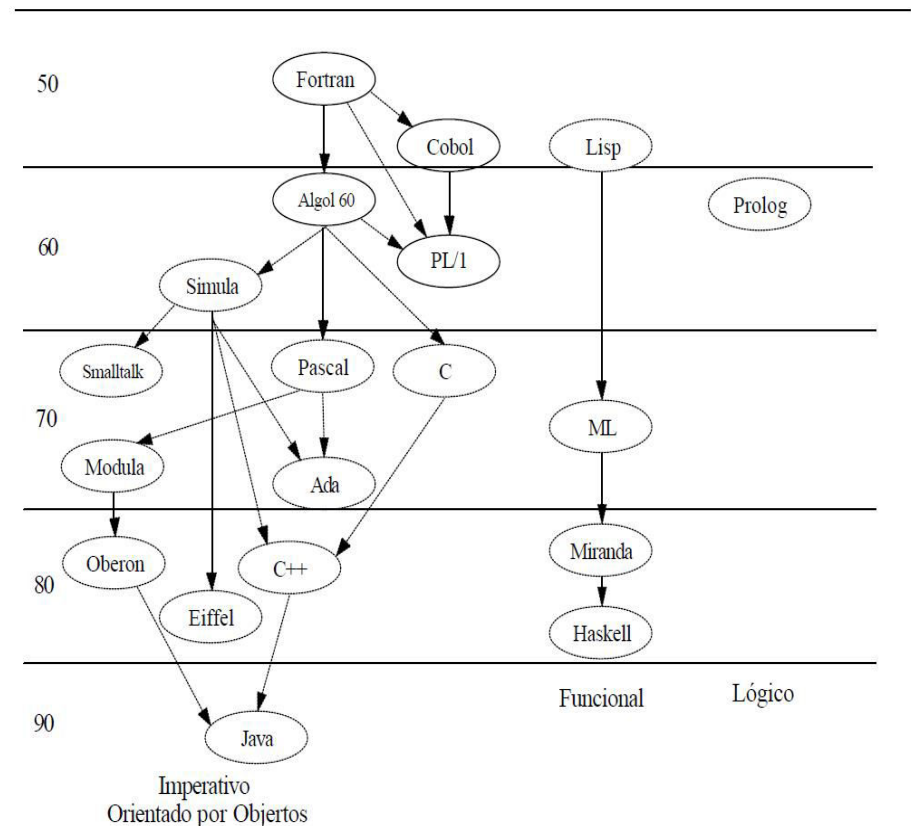


Linguagem de
Programação



Origem

- ▶ Final dos anos 80
- ▶ Linguagem Miranda



Classificação

- ▶ Funcional
- ▶ Estática

Writability

► Haskell vs Java

Haskell

```
let shoppingList = ["Eggs", "Milk"]
```

Java

```
ArrayList<String> shoppingList = new ArrayList<>();  
shoppingList.add("Eggs");  
shoppingList.add("Milk");
```

Expressividade

► Haskell vs C#

Haskell

```
[x * 2 | x <- [1..10]]  
doublesList xs = [x*2 | x <- xs]  
doublesList [1..10]
```

C#

```
public List<int> DoublesList(List<int> list) {  
    var list2 = new List<int>();  
    list.ForEach(x => { list2.Add(x * 2); });  
    return list2;  
}
```

```
var list = new List<int>();  
list.AddRange(new int[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 });  
var result = DoublesList(list);
```

Expressividade

- ▶ Criação de listas infinitas com Lazy Evaluation

```
Positives = [0..]
```

Pattern Matching

```
numDaSorte :: (Integral a) => a -> String
numDaSorte 13 = "Você está com sorte !!"
numDaSorte x = "Opa, parece que você está com azar..."
```

Haskell

```
public string NumeroDaSorte(int a){
    if(a == 7){
        return "Você está com sorte !!"
    } else {
        return "Opa, parece que você está com azar..."
    }
}
```

Java

Pattern Matching

```
factorial :: (Integral a) => a -> a
factorial 0 = 1
factorial n = n * factorial (n - 1)
```

Haskell

```
#include<stdio.h>

int fat, n;

int main()
{
    scanf("%d", &n) ;
    for(fat = 1; n > 1; n = n - 1)
    {
        fat = fat * n;
    }
    printf("\n%d", fat);
    return 0;
}
```

C

► Obrigado !