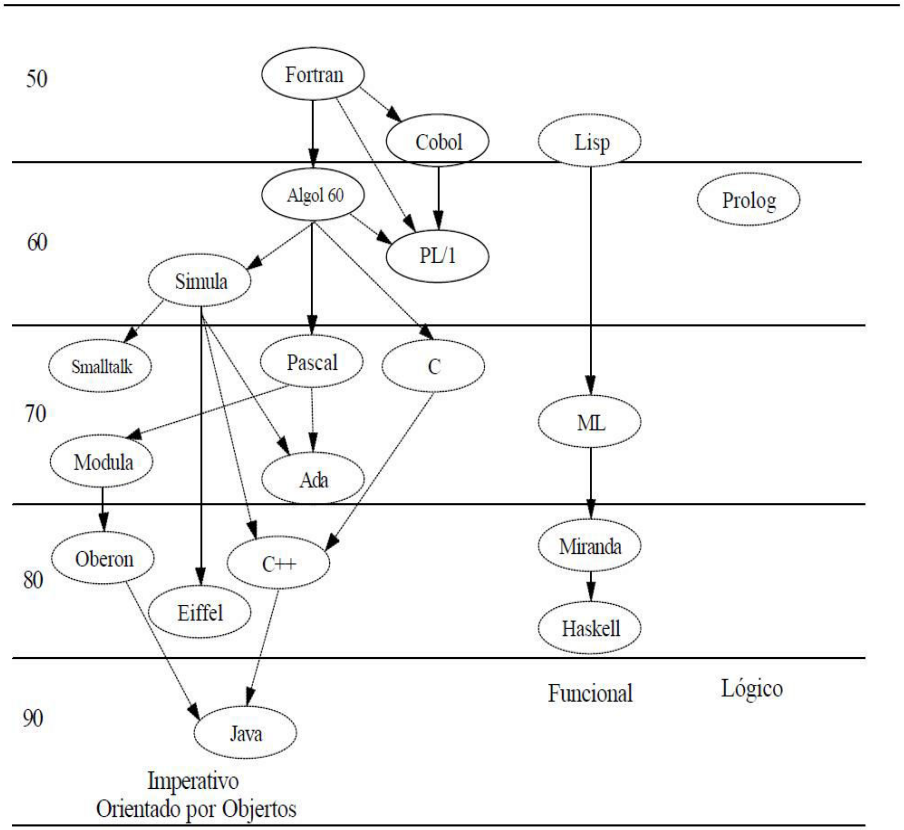
Linguagem de Programação

Origem

- Final dos anos 80
- Linguagem Miranda



Classificação

- Funcional
- Estática

Writability

- Haskell vs Java

Haskell

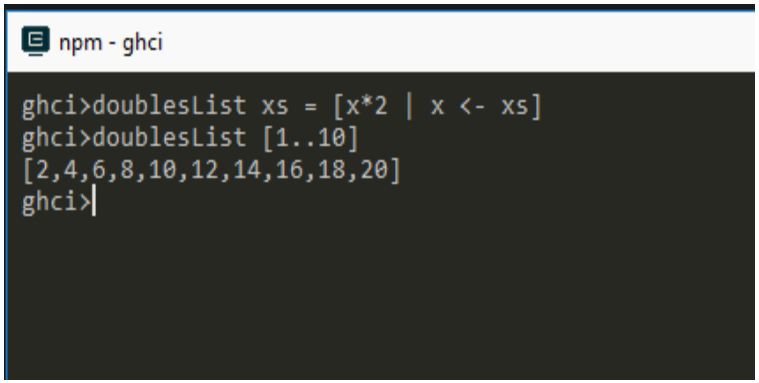
```
let shoppingList = ["Eggs", "Milk"]
```

Java

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

Compreensão de listas

- Haskell vs C#




```
public class Program
{
    public static void Main(string[] args)
    {
        var list = new List<int>();
        list.AddRange(new int[] { 1,2,3,4,5,6,7,8,9,10 });
        var result = DoublesList(list);
    }

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

Lazy Evaluation

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

```
npm - ghci

ghci>list = [1..]
ghci>100 `elem` list
True
ghci>1000000000 `elem` list
True
ghci>
```

```
public static void Main(string[] args)
{
    var list = new List<int>();
    list.AddRange(new int[] { 1,2,3,4,5,6,7,8,9,10 });

    var result = list.Contains(100);

    Console.WriteLine($"Lista contém o número 100 ? {result}");
    Console.ReadKey();
}
```

```
C:\Program Files\dotnet\dotnet.exe
Lista contém o número 100 ? False
```

```
public static void Main(string[] args)
{
    var list = new List<int>();
    int count = 1;

    while (true) {
        list.Add(count);

        count++;
    }
}
```

Pattern Matching

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

Haskell

```java
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

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

Haskell

```c
#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

Currying Functions

Haskell VS C#

```
ghci> max 4 5
5
ghci> (max 4) 5
5
```

```
npm - ghci

ghci>addThreeNumbers x y z = x + y + z
ghci>addThreeNumbers 2 4 6
12
ghci>
```

```
public static void Main(string[] args)
{
    Func<int,int,int> add = (x,y) => x + y;
    int a = add(2,3);

    Console.WriteLine($"Resultado: {a}");
    Console.ReadKey();
}
```

```
C:\Program Files\dotnet\dotnet.exe
Resultado: 5
```