

## Enunciado do 2º Trabalho Prático

### Objetivo

Desenvolvimento exercícios e comparação de desempenho em LPs funcionais e imperativas modernas.

### Introdução

Cientistas da computação há trinta anos ou mais (matemáticos), buscaram uma forma em que as linguagens de programação fossem mais próximas da "linguagem matemática". O paradigma de programação funcional tem este propósito e enfatiza a aplicação de funções, em contraste com a programação imperativa que enfatiza as mudanças de um programa.

Este trabalho tem como objetivo conhecer e aprender novas práticas da programação funcional, sugerindo a implementação de problemas simples (e alguns mais complexos), para melhor entendimento das construções funcionais. Além disso utilizar estes problemas como comparação à uma linguagem imperativa, analisando o formato de implementação e desempenho.

### Descrição

As aplicações **simples** para desenvolvimento são:

- Testar se um elemento é membro de uma lista
- Calcular o tamanho de uma lista
- Calcular a soma dos elementos de uma lista
- Calcular o produto dos elementos de uma lista
- Reversão de lista
- Testar se duas listas são iguais
- Concatenação de duas listas
- Interseção de duas listas

Duas aplicações são especificadas a seguir, para que seja realizado a **análise de desempenho entre linguagens funcionais VS imperativas**. São eles:

- Ordenação (*quicksort*)
- Problema do Corte de Estoque:
  - O problema de corte de estoque envolve otimizar o corte de materiais (geralmente barras de comprimento fixo) em pedaços menores para atender a uma série de pedidos com comprimentos específicos. Este problema é comumente encontrado em indústrias como a de madeira, metal e tecidos. O objetivo é maximizar a utilização do material e minimizar o desperdício. Para adaptar esse problema em função de um parâmetro N, você pode considerar que existem N pedidos de diferentes comprimentos e barras de estoque com um comprimento fixo. O desafio será encontrar a combinação ideal de cortes para atender a todos os pedidos, maximizando o número de pedidos atendidos.

Deverá ser desenvolvido TODOS os problemas em uma linguagem de programação funcional, inclusive (1) os dois últimos problemas. Estes dois últimos problemas deve ser desenvolvido também em uma linguagem imperativa em duas versões:

- (2) Sem a utilização de construções funcionais;
- (3) Maximizando a utilização de construções funcionais.

Por fim, deve ser comparado o **tempo de execução** entre as implementações das versões (1), (2) e (3), com três diferentes tamanhos de N (calibrar em função da LP funcional):

- ~1 segundo
- ~10 segundos
- ~1 minuto

Lista de LPs funcionais:

- Closure
- Erlang
- Haskell

Lista de Lps imperativas:

- Java
- JavaScript
- Python

**Pontos importantes:**

- A utilização correta da linguagem de programação funcional e imperativa, quanto ao seu “estilo” padrão de programação;
- Qualquer tipo de cópia (trabalhos de colegas, internet, etc) **anulará** o trabalho, seja porções de código ou simplesmente o trabalho completo;
- O trabalho é **no máximo em trios**;
- Valor de **2,5 pontos** (sendo a apresentação para a sala 1,0 ponto da nota, mostrando itens interessantes da linguagem utilizada);
- A **entrega** do trabalho e início da apresentação será no dia **04/12/2023**;

**Bom trabalho!**