

## Introducao

O paradigma funcional tem ganho notorieadade junto de grandes empresas e programadores em detrimento de outros paradigmas pois este paradigma permite que em poucas linhas de código (quando comparado com outros paradigmas) se consiga criar soluções robustas e eficientes.

Neste documento abordaremos as vantagens do paradigma funcional, como pode ser feita a sua implementação em **C++** e compararemos **C++** em estilo funcional com uma linguagem funcional, **Haskell**.

Iremos detalhar alguns aspectos de Programação Funcional tais como:

- Imutabilidade
- Lazy Evaluation
- Composicao
- ADTs

Destes, composicao é , provavelmente, o mais importante e talvez o único inerente a Programação Funcional.

A ideia central de Programação Funcional é que construindo pecas pequenas, facéis de entender e de provar como correctas, é também fácil construir um sistema, mesmo que complexo, correctamente.

De seguida, imutabilidade, a ideia em que objectos não são alterados, mas copiados, para implementar mudanças. Esta propriedade ajuda a evitar erros comuns em Programação Imperativa, causados pela partilha de memória e a não especificação da relação entre estados.

Lazy Evaluation, não sendo adoptada como estratégia de avaliação, pode ser usada como estratégia de optimização, especialmente quando combinada com imutabilidade e partilha de memória.

Finalmente, ADTs (*Algebraic Data Types*) são um forma de definir formalmente novos tipos de dados a partir de tipos já existentes. Apesar de não serem essenciais para a Programação Funcional, é desejavel criar abstrações no sistema de tipos que ajudem a descrever o problema com que nos deparamos, dando significado a valores e tentando limitar o conjunto de valores possíveis aos estritamente válidos.

Para cada um destes, mostraremos e analisaremos exemplos de como se faz em **Haskell** e como se pode fazer em **C++**. A unica excepção será Lazy Evaluation, visto que em **Haskell** é adoptada como estrategia de avaliação e, como tal, não há necessidade de mostrar como se faz em **Haskell**.

Ao longo do documento serão usados pequenos programas ou excertos de código **Haskell** e **C++** de forma a auxiliar a análise e a comparação das linguagens.