



# **Programação Funcional com Elixir**



# **Programação Funcional e Paralelismo/Concorrência**

# Programação Funcional e Paralelismo/Concorrência

- A ideia de programação concorrente não é nova, e até algum tempo atrás os processadores possuíam apenas um núcleo, sendo assim, executar dois programas ao mesmo tempo, exigia que o processador executasse “um pouquinho” de cada, fazendo parecer que eles estavam funcionando ao mesmo tempo, quando na verdade eles estavam funcionando “**concorrentemente**”.

# Programação Funcional e Paralelismo/Concorrência

- Atualmente temos processadores que facilmente são multicore, ou seja, com vários núcleos (dual core, quad core, etc)
- No entanto, muitos programas não tiram proveito disso, trabalhando ainda como “antigamente”, sem aproveitar todo potencial que os processadores dispõem.

# Programação Funcional e Paralelismo/Concorrência

- Se executarmos um programa e cada parte dele for executada por um núcleo processador, podemos dizer que ele está sendo executado em **paralelo**.
- Ou ainda, se forem executados dois programas e cada um for executado em um core diferente, também teremos paralelismo, pois eles não disputam o mesmo core.
- Sendo assim temos situações distintas de “**paralelismo**” e “**concorrência**” na execução de um sistema.

# Programação Funcional e Paralelismo/Concorrência

- O paralelismo/concorrência, acaba se beneficiando quando usamos programação funcional, visto que muitos dos padrões aplicados (imutabilidade, funções puras, etc) facilitam a execução de programas em múltiplos cores.
- Em relação ao Elixir, temos ainda uma vantagem, pois essa linguagem funciona em cima de uma máquina virtual (BEAM) que foi projetada para esse fim.