

# Programação funcional em Erlang

Grupo de Sistemas Distribuídos  
Universidade do Minho

## Objectivos

Familiarização com o compilador e ambiente de execução de Erlang. Exploração do núcleo funcional de Erlang, implementando tipos abstractos de dados puramente funcionais.

## Mecanismos

Módulos, funções, listas e tuplos.

## Exercícios propostos

1. Escreva um módulo que implemente um tipo abstracto de dados `queue`, que permita inserir e remover itens com uma semântica FIFO. Deverão ser disponibilizadas as funções:

```
create() -> Queue  
enqueue(Queue, Item) -> Queue  
dequeue(Queue) -> empty | {Queue, Item}
```

Estas operações deverão permitir criar uma queue vazia, inserir um elemento e remover um elemento (caso possível) respectivamente. Escreva um outro módulo que faça uso destas funções.

2. Escreva um módulo que implemente um tipo abstracto de dados `priorityqueue`, que generalize uma queue introduzindo a noção de prioridade: um dequeue deve remover o item mais antigo da classe com maior prioridade. Deverão ser disponibilizadas as funções:

```
create() -> PriQueue  
enqueue(PriQueue, Item, Priority) -> PriQueue  
dequeue(PriQueue) -> empty | {PriQueue, Item}
```