



## Trabalho prático

### Programação concorrente em Go

#### Objetivo

O objetivo deste trabalho é colocar em prática o projeto e a implementação de programas concorrentes utilizando a linguagem de programação Go (<https://golang.org/>).

#### O Problema

Uma pequena *lan house* em uma pequena cidade possui oito computadores, disponíveis por ordem de chegada. Quando todos os computadores estão em uso, a próxima pessoa na fila tem de esperar até que um computador seja liberado. Em uma determinada manhã, um total de 26 adolescentes estavam esperando pela abertura da *lan house*. Em média, cada um desses adolescentes gastaria entre 15 minutos e 2 horas *on-line*.

Escreva um programa em Go que simule o funcionamento da *lan house* na manhã em questão para atender os adolescentes. Um exemplo de saída a ser exibida pelo programa seria:

```
Adolescente D. está on-line
Adolescente Y. está on-line
Adolescente I. está on-line
Adolescente O. está on-line
Adolescente A. está on-line
Adolescente P. está on-line
Adolescente B. está on-line
Adolescente E. está on-line
Adolescente Q. está aguardando
Adolescente F. está aguardando
Adolescente C. está aguardando
Adolescente R. está aguardando
Adolescente G. está aguardando
Adolescente L. está aguardando
Adolescente M. está aguardando
```

Adolescente J. está aguardando  
Adolescente S. está aguardando  
Adolescente N. está aguardando  
Adolescente K. está aguardando  
Adolescente T. está aguardando  
Adolescente V. está aguardando  
Adolescente W. está aguardando  
Adolescente X. está aguardando  
Adolescente H. está aguardando  
Adolescente U. está aguardando  
Adolescente B. liberou a máquina após passar 15 minutos on-line  
Adolescente Q. está on-line  
Adolescente E. liberou a máquina após passar 18 minutos on-line  
Adolescente F. está on-line  
Adolescente O. liberou a máquina após passar 40 minutos on-line  
Adolescente C. está on-line  
Adolescente P. liberou a máquina após passar 41 minutos on-line  
Adolescente R. está on-line  
Adolescente F. liberou a máquina após passar 40 minutos on-line  
Adolescente G. está on-line  
Adolescente D. liberou a máquina após passar 60 minutos on-line  
Adolescente L. está on-line  
Adolescente Q. liberou a máquina após passar 64 minutos on-line  
Adolescente M. está on-line  
Adolescente A. liberou a máquina após passar 88 minutos on-line  
Adolescente J. está on-line  
Adolescente I. liberou a máquina após passar 100 minutos on-line  
Adolescente S. está on-line  
Adolescente Y. liberou a máquina após passar 114 minutos on-line  
Adolescente N. está on-line  
Adolescente S. liberou a máquina após passar 24 minutos on-line  
Adolescente K. está on-line  
Adolescente G. liberou a máquina após passar 88 minutos on-line  
Adolescente T. está on-line  
Adolescente R. liberou a máquina após passar 107 minutos on-line  
Adolescente V. está on-line  
Adolescente C. liberou a máquina após passar 110 minutos on-line  
Adolescente W. está on-line  
Adolescente L. liberou a máquina após passar 100 minutos on-line  
Adolescente X. está on-line  
Adolescente M. liberou a máquina após passar 87 minutos on-line  
Adolescente H. está on-line  
Adolescente T. liberou a máquina após passar 26 minutos on-line  
Adolescente U. está on-line  
Adolescente K. liberou a máquina após passar 65 minutos on-line  
Adolescente X. liberou a máquina após passar 32 minutos on-line  
Adolescente H. liberou a máquina após passar 28 minutos on-line

```
Adolescente J. liberou a máquina após passar 110 minutos on-line
Adolescente W. liberou a máquina após passar 57 minutos on-line
Adolescente N. liberou a máquina após passar 98 minutos on-line
Adolescente V. liberou a máquina após passar 106 minutos on-line
Adolescente U. liberou a máquina após passar 111 minutos on-line
```

```
A lan-house está finalmente vazia e todos foram atendidos
```

Um detalhe importante a observar é que, neste problema, o acesso ao recurso compartilhado deve se dar **na ordem de chegada** das requisições para obtenção de acesso. No caso, o primeiro adolescente que precisar aguardar devido ao fato de que todos os computadores da *lan house* (o recurso compartilhado em questão) estão ocupados deverá ser o primeiro a utilizar algum computador uma vez que esteja disponível.

**Dica:** Na execução do programa, **não é necessário** simular de forma literal o tempo *on-line* de cada pessoa. Por exemplo, os tempos de 15 minutos e 2 horas podem ser representados respectivamente por tempos de 7,5 e 60 segundos (pelos quais a gorotina em questão será suspensa), porém a exibição dos tempos na saída padrão deverá ainda preservar o tempo equivalente entre 15 e 120 minutos. Uma sugestão seria utilizar o seguinte trecho de código como ponto de partida:

```
intervalo := ... // Numero aleatorio entre 15 e 120
time.Sleep(time.Duration(intervalo / 2) * time.Second) // Gorotina posta para dormir
fmt.Println(intervalo, "minutos") // Exibicao na saida padrao
```

## Tarefas

A tarefa central a ser realizada neste trabalho consiste em projetar e implementar uma solução concorrente para o problema anteriormente descrito utilizando a linguagem de programação Go. Em cada um dos problemas, você poderá utilizar o método `Sleep` provido pelo pacote `time` para simular a realização de atividades. Os diferentes intervalos de tempo deverão ser simulados de forma randômica por meio de facilidades providas pelo pacote `math/rand`, porém é necessário garantir valores realmente randômicos e diferentes a cada execução do programa.

O desenvolvimento da solução deve de antemão visar pela busca de desenvolvimento de *software* de qualidade, isto é, funcionando correta e eficientemente, exaustivamente testado, bem documentado e com tratamento adequado de eventuais exceções. Mais ainda, a implementação deverá garantir corretude do programa com relação a concorrência e aplicar de forma adequada as facilidades de gorotinas. Para promover sincronização entre as gorotinas, você poderá fazer uso de canais de comunicação (com ou sem *buffer*) e/ou de mecanismos de sincronização avançada providos pelo pacote `sync` da linguagem. Seu código fonte deverá estar **obrigatoriamente** comentado, a fim de prover entendimento adequado daquilo que foi realizado.

## Autoria e política de colaboração

O trabalho deverá ser feito **individualmente**. O trabalho em cooperação entre estudantes da turma é estimulado, sendo aceitável a discussão de ideias e estratégias. Contudo, tal interação não deve ser entendida como permissão para utilização de (parte de) código fonte de outras equipes, o que pode caracterizar situação de plágio. Trabalhos copiados em todo ou em parte de outras equipes ou da Internet serão sumariamente rejeitados e receberão nota zero.

A critério do professor, qualquer estudante pode ser eventualmente convocada para uma entrevista cujo objetivo é confirmar a autoria do trabalho desenvolvido. Durante a entrevista, o estudante deverá ser capaz de explicar, com desenvoltura, qualquer parte do trabalho. Portanto, é possível que ocorra, após a entrevista, ajuste na nota do estudante entrevistado.

## Entrega

Você deverá submeter um único arquivo no formato `.go` com o código fonte resultante da implementação deste trabalho, sem erros de compilação e devidamente testado e documentado, **até as 23h59 do dia 29 de novembro de 2017** através da opção *Tarefas* na Turma Virtual do SIGAA. Se for o caso, é possível fornecer, no campo *Comentários* do formulário eletrônico de submissão da tarefa, o endereço de um repositório remoto destinado ao controle de versões, porém esta opção não exclui a necessidade de submissão do arquivo via SIGAA.

## Avaliação

A avaliação deste trabalho será feita principalmente sobre os seguintes critérios: (i) utilização correta de goroutines e dos mecanismos de sincronização providos pela linguagem de programação Go; (ii) a correteza da execução do programa implementado, tanto com relação a funcionalidades quanto a concorrência, e; (iii) a aplicação de boas práticas de programação, incluindo legibilidade, organização e documentação de código fonte. O trabalho possuirá nota máxima de 5,0 (cinco) pontos, nota essa que será contabilizada para a terceira unidade da disciplina.