



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

# Mecanismos de Sincronização - Resolução do problema proposto

Victor Bittencourt Mafra Veríssimo

20170022115

Natal - RN

Novembro/2022

# 1.Resolução do problema

A ideia que imaginei para a resolução do problema veio após observar os exemplos feitos em sala de aula do produtor-consumidor usando os mecanismos de sincronização, feitos pelo professor Everton Cavalcanti. Após analisar um pouco o funcionamento e dinâmica do código produtor-consumidor, comecei a adequar ao exercício proposto. Primeiro tive que entender como funcionaria o Banheiro, já que seria o recurso compartilhado usado pelas pessoas. Então comecei adicionando os atributos mais comuns que estavam descritos na atividade, precisaria da quantidade máxima de vagas dentro do banheiro que seria passada por linha de comando, uma lista das pessoas que estavam usando, para poder fazer as ações de entrar e sair do banheiro, um Lock para travar as ações de entrar e sair, e três Conditions para verificar se estava cheio, vazio e se as pessoas que estão entrando são do mesmo sexo. Porém comecei a fazer e ainda faltava um atributo, então depois de um tempo adicionei uma string sexoAtual que receberia o sexo da primeira pessoa que entrou no banheiro e mais tarde receberia o outro sexo que foi barrado.

Portanto, a lógica principal para resolução do problema, foi que quando a primeira pessoa que foi instanciada tentasse entrar no banheiro, verificaria se teria alguém lá dentro, se estivesse vazio, essa pessoa poderia entrar e somente pessoas desse mesmo sexo entrariam, além disso, quando a quantidade de vagas internas do banheiro fosse alcançada, informaria as pessoas que são aceitas ao banheiro para esperarem e conforma forem saindo elas podem entrar. E ao final, quando a última pessoa do sexo sair e o banheiro estiver vazio, as pessoas de sexo diferente são alertadas, podendo assim entrar com tranquilidade.

## 1.1 Mecanismos usados

Foram usados 3 Conditions e um Lock(ReentrantLock) para resolução do problema, todos providos pela biblioteca `java.util.concurrent.locks`.

Todos os mecanismos ficam na classe Banheiro, e foi ativado o atributo justiça na instanciação do objeto Lock. Tanto o Lock como os Conditions se comunicam em somente dois métodos, o método `entrarBanheiro` e `sairBanheiro`, onde escolhi dar o lock no início de ambos os métodos e fazer as sinalizações e os awaits dos Conditions após as verificações dentro de cada método.

## 1.2 Dificuldades no processo

Durante o processo de desenvolvimento, tiveram alguns problemas como a forma de receber os dados a serem usados e diferenciação dos homens e mulheres, depois de pensar muito, resolvi usar um boolean para diferenciar e assim cria-los usando um random, e também do funcionamento da classe `Math.Random`, pois no processo de geração de números aleatórios, alguns deram problemas, pois estavam gerando a mesma sequência de números aleatórios.

## 1.3 Instruções para execução

O projeto foi criado no Eclipse e os valores são colocados via linha de comando na configuração de execução do código. O PRIMEIRO valor significa a quantidade de vagas disponíveis dentro no banheiro, já o SEGUNDO é a quantidade de pessoas que serão criadas para utilização do banheiro unissex. Exemplo de entrada válida: 4 15

São mostradas mensagens no terminal que indicam quem entrou no banheiro, quem tentou entrar no banheiro, quem saiu do banheiro e quem está esperando para entrar no banheiro.

## 1.4 Repositório

<https://github.com/VictorBitt15/BanheiroUnisex>