Nesse caso vamos precisar de uma lista para retornar todas as series que tem aquele ator.

Observa que estamos criando uma lista porque pode retorna uma serie ou varias

Também estamos pegando somente o nome do ator e avaliação, do que vem do banco!

private void buscarSeriesPorAtor() {  
 System.*out*.println("Qual nome do ator?");  
 String nomeAtor = leitura.nextLine();  
  
 **List<Serie> seriesEncontradas = repository.findByAtoresContainingIgnoreCase(nomeAtor);  
 System.*out*.println("Serie que " + nomeAtor + " trabalhou: ");  
 seriesEncontradas.forEach(s -> System.*out*.println(s.getTitulo() + " avaliação: " + s.getAvaliacao()));**  
 }  
}

public interface SerieRepository extends JpaRepository<Serie, Long> {  
  
  
 **List<Serie> findByAtoresContainingIgnoreCase(String nomeAtor);**}

## Agora vamos imaginar que eu queira exibir somente as series que a avaliação seja maior que 8

List<Serie> findByAtoresContainingIgnoreCase**AndAvaliacaoGreaterThanEqual(String nomeAtor, double avaliacao);**

Repare o código e repare que colocamos outro parâmetro

private void buscarSeriesPorAtor() {  
 System.*out*.println("Qual nome do ator?");  
 var nomeAtor = leitura.nextLine();  
  
 System.*out*.println("Avaliação a partir de que valor?");  
 var avaliacao = leitura.nextDouble();  
 leitura.nextLine();  
  
 List<Serie> seriesEncontradas = repository.findByAtoresContainingIgnoreCaseAndAvaliacaoGreaterThanEqual(nomeAtor, avaliacao);  
 System.*out*.println("Serie que " + nomeAtor + " trabalhou: ");  
 seriesEncontradas.forEach(s -> System.*out*.println(s.getTitulo() + " avaliação: " + s.getAvaliacao()));  
}