



# Agendamento de Tarefas Scheduler



# Tarefas agendadas com Scheduler

- São tarefas agendadas e configuradas pelo Spring.
- Existem regras simples que devemos seguir para implementar uma tarefa agendada:
  - Cada tarefa agendada será um método
  - O método não deve retornar nada (void)
  - O método não deve esperar nenhum parâmetro

```
public void tarefa() {  
  
}
```

# Tipos de Agendamento

```
// executa conta o tempo à partir do início da execução do método  
@Scheduled(fixedRate = 5000)
```

```
// executa conta o tempo à partir do fim da execução do método  
@Scheduled(fixedDelay = 5000)
```

```
// espera o tempo (initialDelay) para executar a tarefa  
@Scheduled(fixedDelay = 1000, initialDelay = 1000)
```

# Tipos de Agendamento - **Cron Expressions**

```
@Scheduled(cron = "* * * * *")
```

- "0 0 \* \* \* \*" = todo dia à meia noite.
- "\*/10 \* \* \* \* \*" = cada 10 segundos.
- "0 0 8-10 \* \* \* \*" = 8, 9 e 10 horas de todos os dias.
- "0 0 6,19 \* \* \* \*" = 6h e 19h todos os dias.
- "0 0/30 8-10 \* \* \* \*" = 8h, 8h30, 9h00, 9h30, 10h00 e 10h30 todos os dias.
- "0 0 9-17 \* \* MON-FRI" = das 9h às 17h, dias úteis
- "0 0 0 25 12 ?" = todo Natal à meia noite

# Dependência

- Dependência para testes

```
<dependency>
  <groupId>org.awaitility</groupId>
  <artifactId>awaitility</artifactId>
</dependency>
```

- <https://spring.io/guides/gs/scheduling-tasks>
- <https://www.baeldung.com/awaitility-testing>

# Ativando o agendamento - Scheduling

```
@SpringBootApplication
@EnableScheduling
public class SchedulerApplication {

    public static void main(String[] args) { SpringApplication.run(SchedulerApplication.class, args); }

}
```



Vamos praticar!



# Exercício #1

- Criar uma rotina programada para **cada 10 segundos** no seu projeto que imprima uma frase de **log: “Agendamento (data atual)”**, para testar a funcionalidade.



Obrigado!

DBC

DIGITAL BUSINESS COMPANY®



 /dbc.company

 /dbccompany

 /dbccompany.com.br

 /company/dbc-company