



06. A nova API de datas

📅 Date	@01/11/2022
📁 Categoria	Java
📖 Curso	Java 8: conheça as novidades dessa versão

Tópicos

- A nova API de datas
- Data e hora com time-zone
- Data atual com LocalDate
- Criando uma data com LocalDate
- Criando um período entre duas datas
- Formatando suas datas

A nova API de datas

Vamos começar criando uma data, a data de hoje. Para representar uma data em java agora eu posso utilizar a classe `LocalDate`, presente no pacote `java.time`. Repare como é facil ter a data atual utilizando o método `now()`:

```
public class Datas {  
  
    public static void main(String[] args) {  
  
        LocalDate hoje = LocalDate.now();  
        System.out.println(hoje);  
    }  
}
```

```
}  
}
```

Assim como este existem diversos outros métodos estáticos nas novas classes da API de datas. Vamos conhecer vários no decorrer do capítulo.

Repare que o valor impresso neste caso será `2014-05-28`, logo veremos como formatar essa saída de outras formas.

Trabalhando com Period

Ao executar esse código temos o resultado esperado, que neste caso é 2 anos. Mas e se quiséssemos descobrir a diferença de dias e meses também? Daria pra fazer da mesma forma, mas sempre que você tiver que fazer um trabalho dessa na mão você pode ter certeza que já existe algo pronto pra te ajudar de alguma forma. Nesse caso podemos utilizar a classe `Period`.

Para saber a diferença entre duas datas podemos utilizar seu método `between`, da seguinte forma:

```
Period periodo = Period.between(hoje, olimpiadasRio);  
System.out.println(periodo);
```

Repare que a saída desse nosso `println` vai ser um pouco estranha, um exemplo seria: `P2Y8D`.

Formatando suas datas

Para formatar nossas datas podemos utilizar o `DateTimeFormatter`. Existem diversos já prontos, mas há ainda a alternativa de você criar o seu próprio formatador no padrão já conhecido de `dd/MM/yyyy`.

Para fazer isso basta você utilizar o método `ofPattern`:

```
DateTimeFormatter formatador = DateTimeFormatter.ofPattern("dd/MM/yyyy");
```

Trabalhando com medida de tempo

Por enquanto só estamos trabalhando com datas, fazendo formatações e manipulando seu resultado. Mas é muito comum eu também precisar trabalhar com horas, minutos e segundos. Ou seja, trabalhar com uma medida de data com tempo.

Para isso podemos utilizar a classe `LocalDateTime`, de forma bem similar podemos fazer:

```
LocalDateTime agora = LocalDateTime.now();
```
