



## Datas com Java - Parte 1

---

Técnicas de Programação I

# Datas com Java - Parte I

Técnicas de Programação I

A manipulação de datas no Java foi modificada com o novo pacote **java.time**, presente na linguagem a partir da versão **1.8**.

Deste então, você precisa evitar usar os pacotes *java.util.Date*, *java.util.Calendar* e *java.text.DateFormat*, considerados antigos e legados.

Neste módulo iremos apresentar em como utilizar o pacote **java.time** e suas classes na criação, formatação e operações sobre datas e horários. \

Além da sua formatação e conversão para o formato String.

# Datas com Java - Parte I

Técnicas de Programação I

Categoria	Descrição	Classes
Datas e horários	Representa a data e horário local.	<i>java.time.LocalDate, java.time.LocalTime, java.time.LocalDateTime</i>
Data/hora com fuso horário	Representa data e hora com informação de fuso horário.	<i>java.time.ZonedDateTime, java.time.OffsetDateTime</i>
Formatadores de datas e horas	Analizam e imprimem datas e horas com padrões e em uma variedade de estilos.	<i>java.time.format.DateTimeFormatter</i>
Ajustes de datas e horas	Ajustam e manipulam datas e horários com incrementos temporais.	<i>java.time.temporal.TemporalAdjusters, java.time.temporal.ChronoUnit</i>
Períodos, durações e instantes de tempo	Representam uma quantidade de tempo, períodos de dias ou superiores e durações de curtos períodos de tempo como minutos ou segundos, como um específico instante de tempo.	<i>java.time.Periods, java.time.Durations, java.time.Instants</i>

Tabela: Lista de classes do pacote java.time e sua utilização



# Classes LocalDate, LocalDateTime e LocalTime

Técnicas de Programação I

A partir da versão **1.8 do Java**, a representação de datas e horários são representadas pelas classes `LocalDate`, `LocalDateTime` e `LocalTime`, respectivamente.

## Instanciando objetos com o método `now()`

Para capturarmos a data e/ou hora corrente, utilizamos o método `.now()`.

```
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;

public class InstanciandoDatas {

    public static void main(String[] args) {
        LocalDate data = LocalDate.now();
        LocalTime horario = LocalTime.now();

        LocalDateTime dataHora = LocalDateTime.of(data, horario); // 2025-12-01T16:25:40.605042
        System.out.println("Data e hora usando LocalDateTime.of(data, horario): " + dataHora);

        dataHora = LocalDateTime.now(); // 2025-12-01T16:25:40.606267
        System.out.println("Data e hora usando LocalDateTime.now(): " + dataHora);
    }
}
```

# Classes LocalDate, LocalDateTime e LocalTime

Técnicas de Programação I

## Instanciando datas personalizadas - método of()

A API fornece os seguintes métodos estáticos para a criação de objetos `java.time` customizados:

- `LocalDate.of(YYYY, MM, DD)`
  - retorna um `LocalDate` com o ano (`YYYY`), mês (`MM`) e dia (`DD`) fornecidos.
- `LocalTime.of(hh, mm, ss)`
  - retorna um `LocalTime` com a hora (`hh`), minutos (`mm`) e segundos (`ss`) fornecidos.
- `LocalDateTime.of(YYYY, MM, DD, hh, mm, ss)`
  - retorna um `LocalDateTime` com o ano (`YYYY`), mês (`MM`), dia (`DD`), hora (`hh`), minutos (`mm`) e segundos (`ss`) fornecidos.

Obrigada