REFERÊNCIAS:

Polimorfismo:

<https://www.devmedia.com.br/conceitos-e-exemplos-polimorfismo-programacao-orientada-a-objetos/18701>

<https://medium.com/@atandaoluchiaminat/overload-vs-override-in-object-oriented-programming-oop-a38ca0ccaf40>

Abstrata:

<https://www.devmedia.com.br/conceitos-classes-abstratas-programacao-orientada-a-objetos/18812>

Interface:

<https://www.devmedia.com.br/interfaces-programacao-orientada-a-objetos/18695>

<https://www.dio.me/articles/interfaces-x-classes-abstratas>

**Comparação entre Classes Abstratas e Interfaces:**

[**https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html**](https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html)

[**https://docs.oracle.com/javase/tutorial/java/IandI/createinterface.html**](https://docs.oracle.com/javase/tutorial/java/IandI/createinterface.html)

**Resuma os principais pontos abordados no trabalho:**

[**https://www.baeldung.com/java-abstract-class-vs-interface**](https://www.baeldung.com/java-abstract-class-vs-interface)