

# Abstração



Para você ser, é preciso você fazer.

Sabemos que qualquer sistema de mensagens instantâneas realiza as mesmas operações de Enviar e Receber Mensagem, dentre outras operações comuns ou exclusivas de cada aplicativo disponível no mercado.

Mas será que as ações realizadas, contém o mesmo comportamento ? Acreditamos que não.

📘 Já imaginou a **Microsoft** falar para o **Facebook**: *"Ei, toma meu código do MSN!"*.

O que vale destacar para compreender, é que todo e qualquer sistema de mensagem precisa sim, no mínimo Enviar e Receber Mensagem, logo, consideramos se firmar um "contrato" para qualquer um que queira se apresentar assim para o mercado.

Observem a nova estruturação dos códigos abaixo, com base na implementação apresentada no pilar [Herança](#).

ServicoPai	MSN	Facebook	Telegram
------------	-----	----------	----------

```
public abstract class ServicoMensagemInstantanea {  
    public abstract void enviarMensagem();  
    public abstract void receberMensagem();  
}
```

📘 Em Java, o conceito de abstração é representado pela palavra reservada **abstract** e métodos que **NÃO** possuem corpo na classe abstrata (pai).

✅ É muito difícil falar de abstração e **NÃO** mencionar polimorfismo.



Previous  
[Herança](#)

Next  
[Polimorfismo](#)



Last modified 1mo ago