

FRAMEWORKS

ENGENHARIA DE SOFTWARE



DEFINIÇÃO

Abstração

Fornece uma **API** para programadores

Funcionalidade **genérica extensível**

CARACTERÍSTICAS

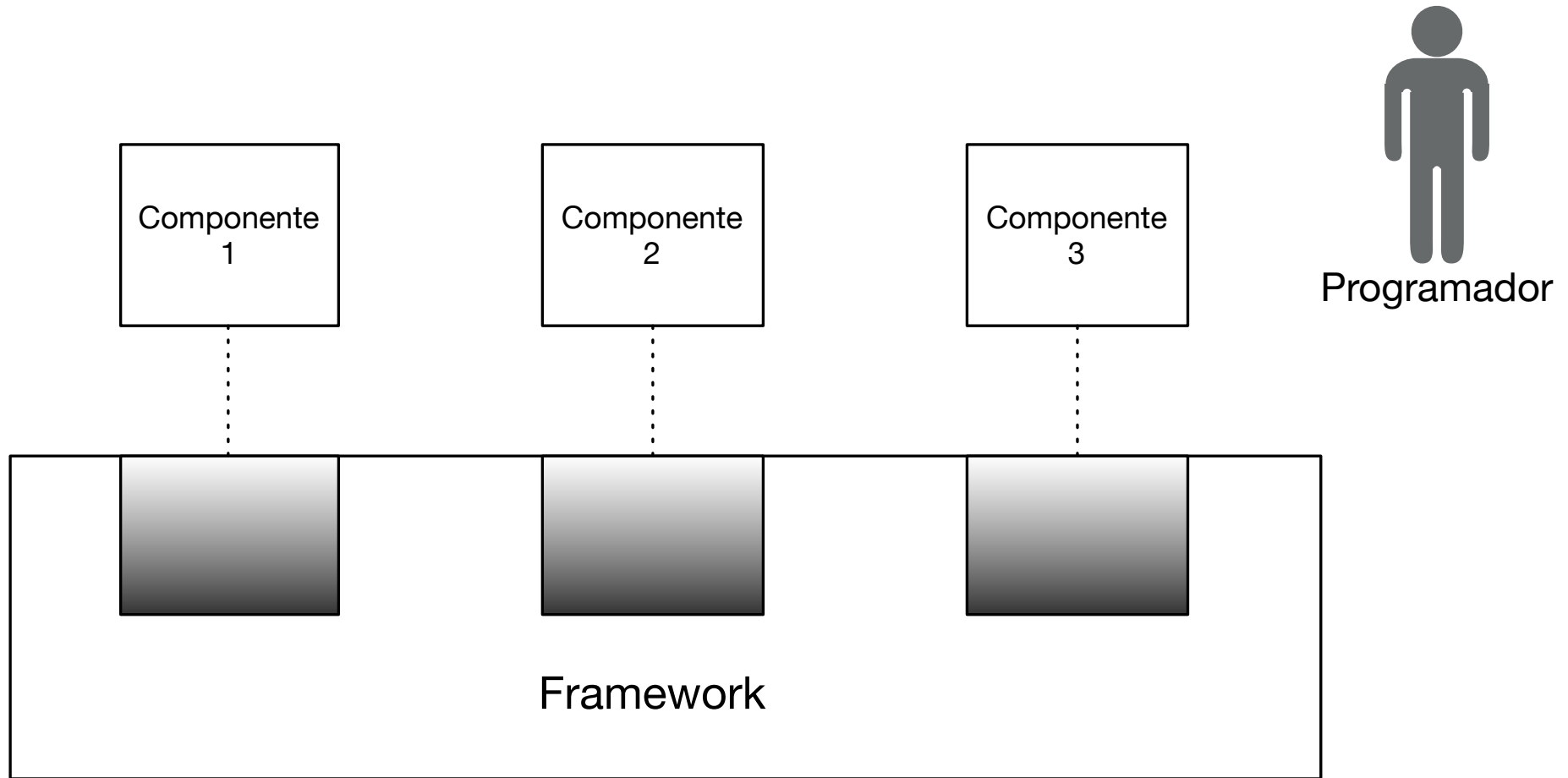
Inversão de Controlo

Fluxo da aplicação controlado pelo *framework*

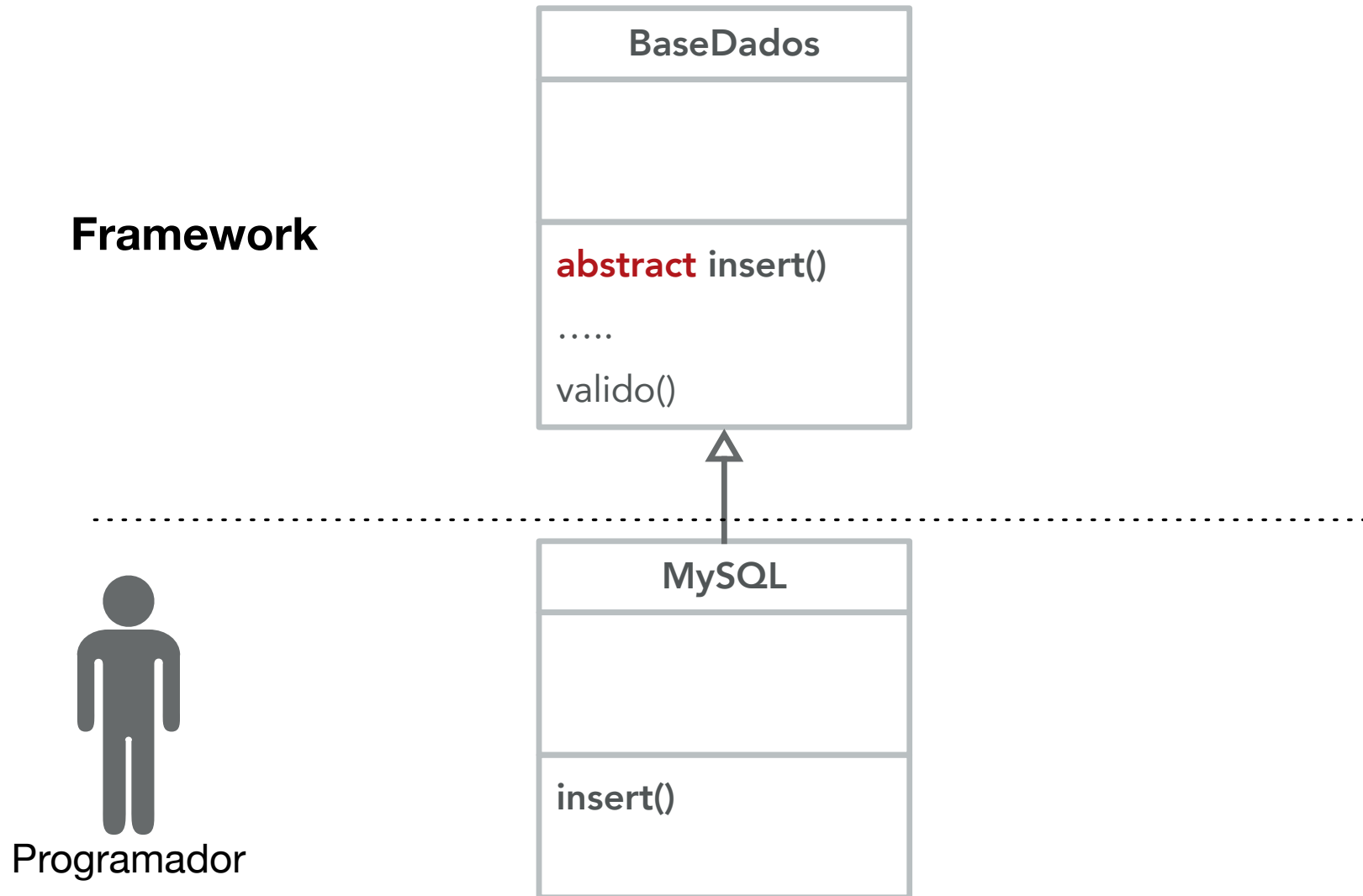
Comportamento *default*

Extensibilidade

ESTRUTURA



EXTENSIBILIDADE



EXEMPLO

```
//imagine this is the entry point for a framework, it can not be changed
public class Main {
    public static void main(String[] args) {
        Human h = new Human(new Walk());
        h.doMove();
    }
}
```

```
public abstract class Move {
    public abstract void action();
}
```

```
public class Walk extends Move {

    @Override
    public void action() {
        // TODO Auto-generated method stub
        System.out.println("5 miles per hour - it is slow!");
    }
}
```

```
public class Human {
    private Move move;

    public Human(Move m){
        this.move = m;
    }

    public void doMove(){
        this.move.action();
    }
}
```

EXEMPLO 2

```
class Solido{  
public:  
    virtual double volume(void) = 0;  
    virtual double area(void) = 0;  
};
```

```
class Esfera : public Solido{  
public:  
    Esfera(double raio){ r = raio; }  
    double volume(void) { return (4.0 * 3.14*r*r*r /3.0); }  
    double area(void) { return (4.0*3.14*r*r); }  
    void setRaio(double raio){ r = raio; }  
private:  
    double r;  
};
```

```
class Cubo : public Solido{  
public:  
    Cubo(double lado_tam) { lado = lado_tam; }  
    double volume(void) { return(lado*lado*lado); }  
    double area(void){ return(6.0*lado*lado); }  
    void setLado(double tamLado){ lado = tamLado; }  
private:  
    double lado;  
};
```

AOP NA CRIAÇÃO DE FRAMEWORKS


```
class Pessoa{
    int tipo;
    String nome;
    String morada;

    //...

    @UPDATEBD
    void setNome(String nome){
        this.nome= nome;
    }
}

public aspect BDGestor{
    @After(@annotation(UPDATEBD) &&
        execution(* *(..)))
    void updateBD(){
        BD.execute("UPDATE Pessoa ...");
    }
}
```

Framework



Programador

BIBLIOGRAFIA

Introduction to Patterns and Frameworks

www.cs.wustl.edu/~schmidt/PDF/patterns-intro4.pdf

Model View Controller

<http://www.martinfowler.com/eaCatalog/modelViewController.html>

Applying Patterns and Frameworks to Develop Object-Oriented Communication Software

<http://www.cse.wustl.edu/~schmidt/PDF/HPL.pdf>