

Programação funcional em Dart

```
int calculaMaximo(List<int> numeros) {  
    ...  
}
```



Função que recebe um List<int> e retorna um int

Programação funcional em Dart

```
int calculaMaximo(List<int> numeros) {  
    ...  
}
```



Função que recebe um List<int> e retorna um int

```
void main() {  
    Function funcaoMax = calculaMaximo;  
}
```

Programação funcional em Dart

```
int calculaMaximo(List<int> numeros) {  
    ...  
}
```

Função que recebe um List<int> e retorna um int

```
void main() {  
    Function funcaoMax = calculaMaximo;  
  
    funcaoMax( 'ola' );  
}
```

função não-tipificada

não dá erro de compilação 😬

Programação funcional em Dart

```
int calculaMaximo(List<int> numeros) {  
    ...  
}
```



Função que recebe um List<int> e retorna um int

```
void main() {  
    int Function(List<int>) funcaoMax = calculaMaximo;  
  
    funcaoMax( 'ola' );  
}
```

← dá erro de compilação! ainda bem!

Exercício

```
int soma(int numero1, int numero2) {  
    ...  
}
```

```
void main() {  
     funcaoSoma = soma;  
  
}
```

Exercício

```
int soma(int numero1, int numero2) {  
    ...  
}
```

```
void main() {  
    int Function(int,int) funcaoSoma = soma;  
  
}
```

Programação funcional em Dart



Programação funcional em Dart



```
List<int> transformLista(List<int> numeros, int Function(int) f) {  
    ...  
}
```


Exercício

```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    ...  
}
```

???

```
void main() {  
    List<int> result = transformaLista([3,4,6], dobro);  
}
```

Resolução

```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    ...  
}
```

```
int dobro(int numero) {  
    return numero * 2;  
}
```

```
void main() {  
    List<int> result = transformaLista([3,4,6], dobro);  
}
```

Resolução

```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    ...  
}
```

```
int dobro(int numero) {  
    return numero * 2;  
}
```

ou

```
int dobro(int numero) => numero * 2;
```

```
void main() {  
    List<int> result = transformaLista([3,4,6], dobro);  
}
```

Lambda

```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    ...  
}
```

```
int dobro(int numero) {  
    return numero * 2;  
}
```

ou

```
int dobro(int numero) => numero * 2;
```

```
void main() {
```

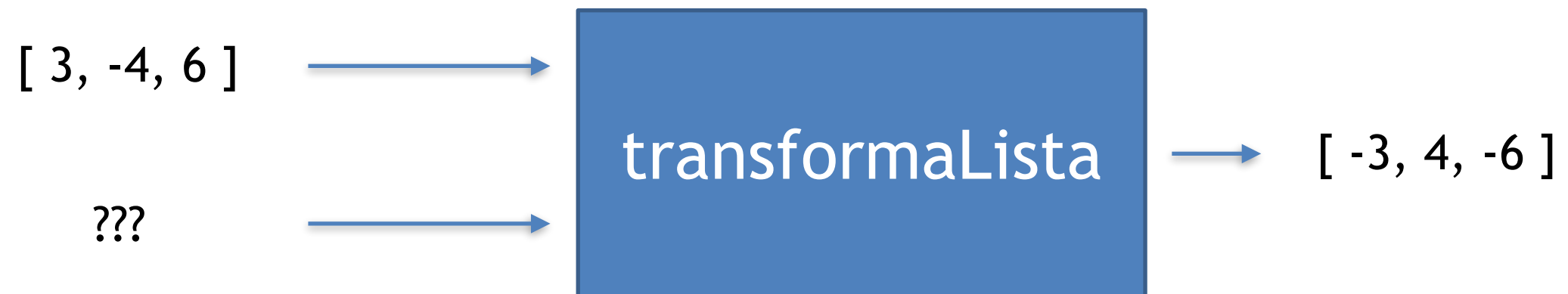
```
    List<int> result = transformaLista([3,4,6], (n) => n*2);
```

```
}
```

Código completo

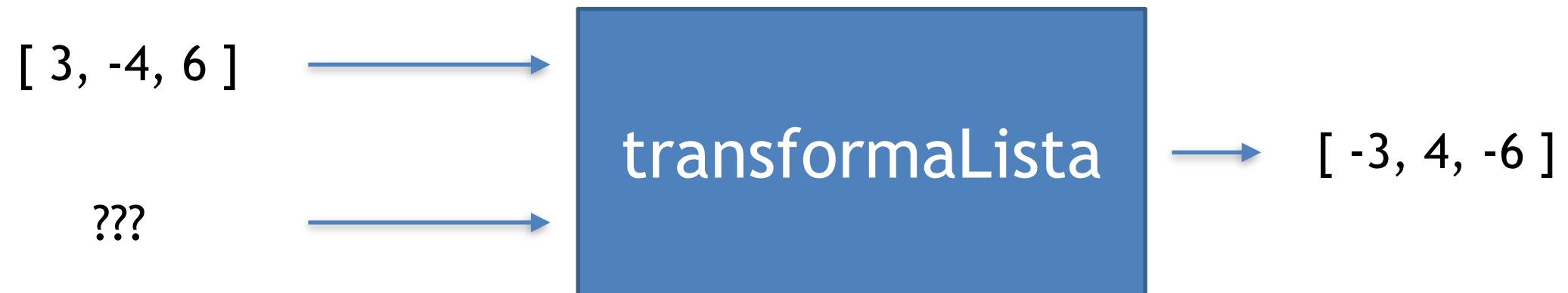
```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    List<int> resultado = [];  
    for (var numero in numeros) {  
        resultado.add(f(numero));  
    }  
  
    return resultado;  
}  
  
void main() {  
    List<int> result = transformaLista([3,4,6], (n) => n*2);  
}
```

Exercício



```
void main() {  
    List<int> result = transformLista([ 3, -4, 6 ], ???);  
}
```

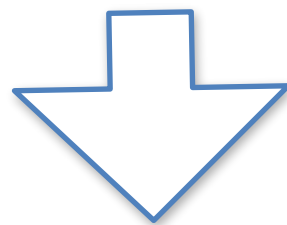
Resolução



```
void main() {  
    List<int> result = transformLista([3,-4,6], (n) => -n);  
}
```

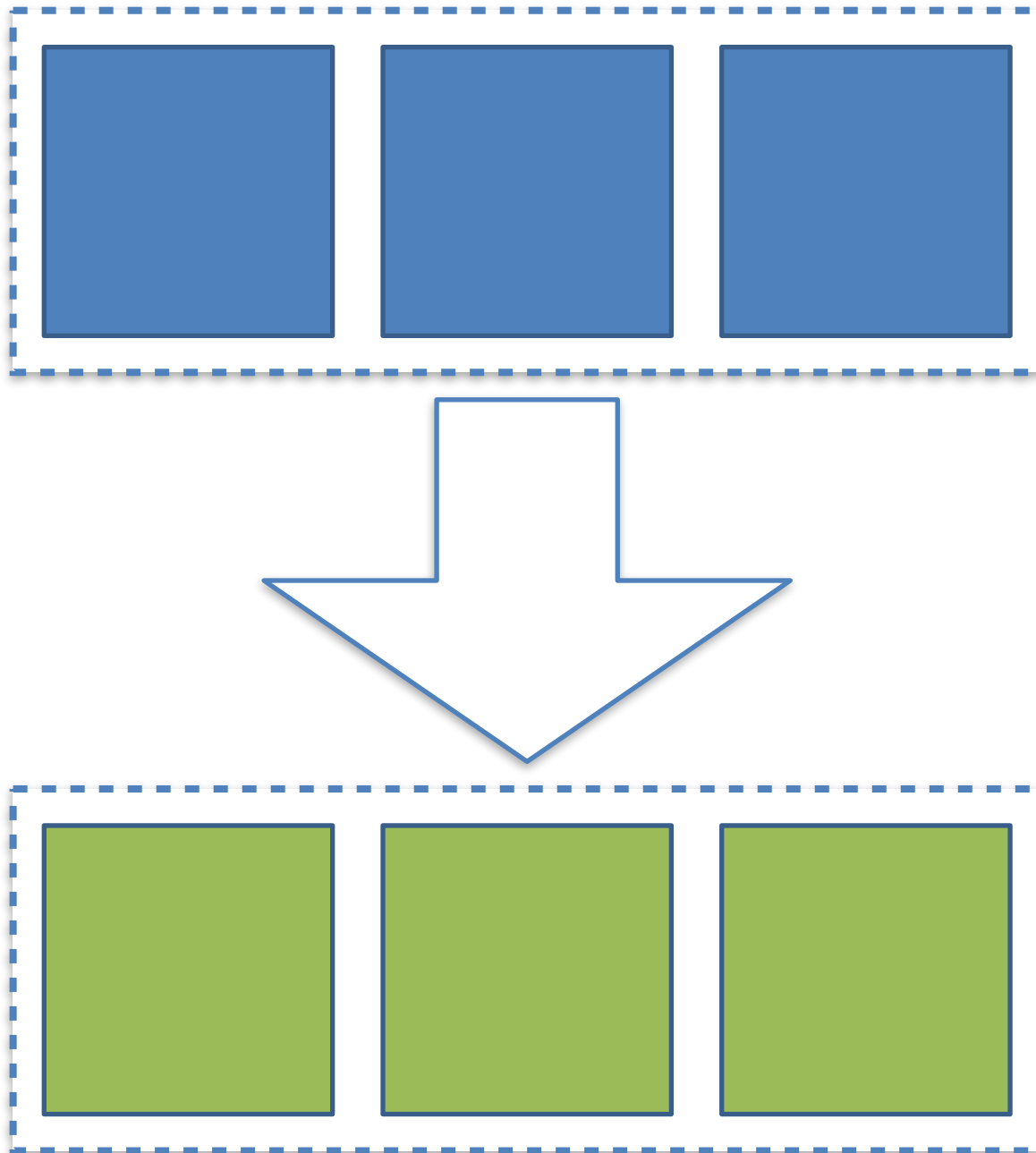
map

```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    List<int> resultado = [];  
    for (var numero in numeros) {  
        resultado.add(f(numero));  
    }  
  
    return resultado;  
}
```

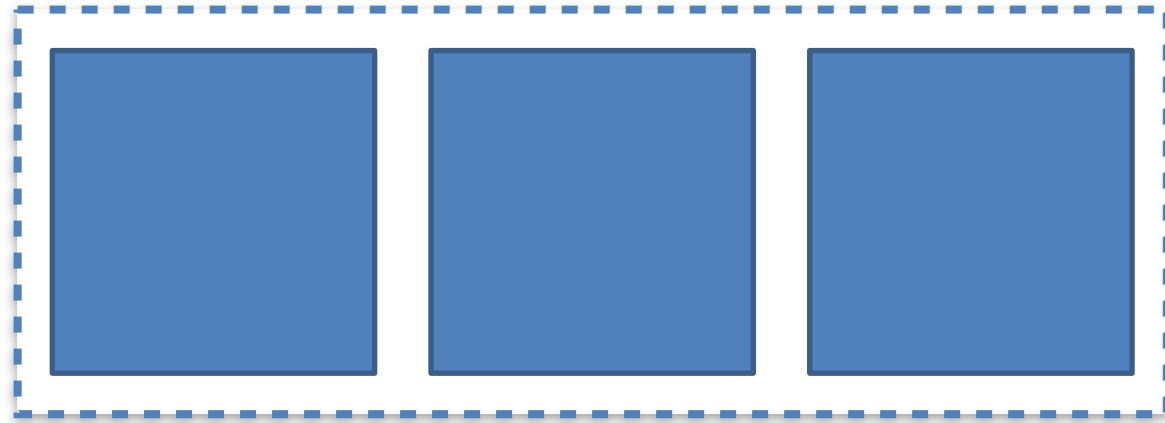


```
List<int> transformaLista(List<int> numeros, int Function(int) f) {  
    return numeros.map((n) => f(n)).toList();  
}
```

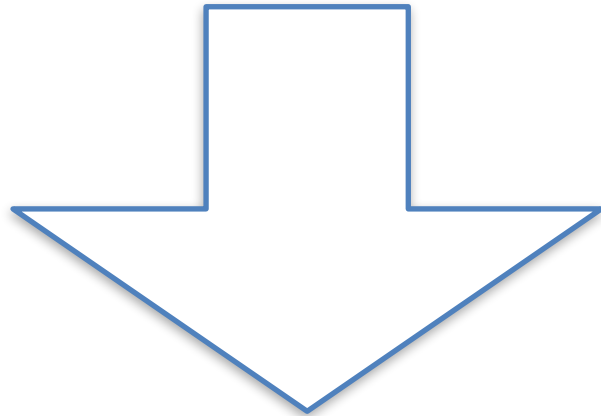

map



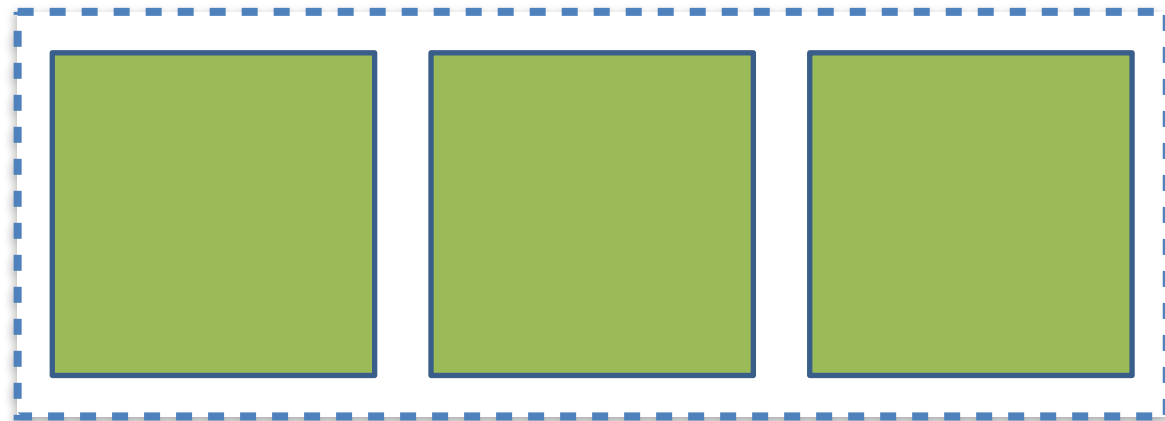
`lista.map(transformador)`



map



`lista.map(transformador)`



Exercício

```
final result = [2,3,7].map( (n) => "$n${n+1}" );
```

Qual o valor da variável result?

Resolução

```
final result = [2,3,7].map((n) => "$n${n+1}");
```



```
["23","34","78"]
```

Callbacks

(problema motivador)

```
class Aluno {  
    int _numero;  
    bool _inscrito = true;  
  
    Aluno(this._numero);  
  
    void cancelarInscricao() {  
        _inscrito = false;  
    }  
}
```

Callbacks

(problema motivador)

```
class Escola {  
    int _numAlunosInscritos = 0;  
  
    Aluno inscreveAluno(int numero) {  
        _numAlunosInscritos++;  
        return Aluno(numero);  
    }  
}
```

```
class Aluno {  
    int _numero;  
    bool _inscrito = true;  
  
    Aluno(this._numero);  
  
    void cancelarInscricao() {  
        _inscrito = false;  
    }  
}
```

Callbacks

(problema motivador)

```
class Escola {
    int _numAlunosInscritos = 0;

    Aluno inscreveAluno(int numero) {
        _numAlunosInscritos++;
        return Aluno(numero);
    }
}

class Aluno {
    int _numero;
    bool _inscrito = true;

    Aluno(this._numero);

    void cancelarInscricao() {
        _inscrito = false;
    }
}

void main() {
    Escola escola = Escola();
    Aluno aluno123 = escola.inscreveAluno(123);
    // ...
    aluno123.cancelarInscricao();
}
```

Qual o problema deste programa?

Callbacks

(problema motivador)

```
class Escola {
    int _numAlunosInscritos = 0;

    Aluno inscreveAluno(int numero) {
        _numAlunosInscritos++;
        return Aluno(numero);
    }
}
```

```
class Aluno {
    int _numero;
    bool _inscrito = true;

    Aluno(this._numero);

    void cancelarInscricao() {
        _inscrito = false;
    }
}
```

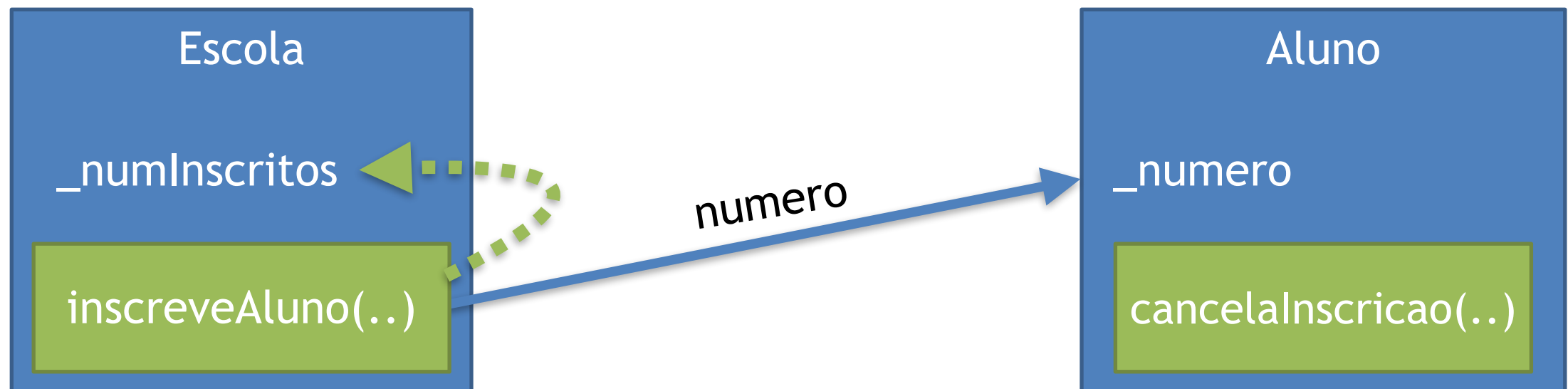
```
void main() {
    Escola escola = Escola();
    Aluno aluno123 = escola.inscreveAluno(123);
    // ...
    aluno123.cancelarInscricao();
}
```

Qual o problema deste programa?

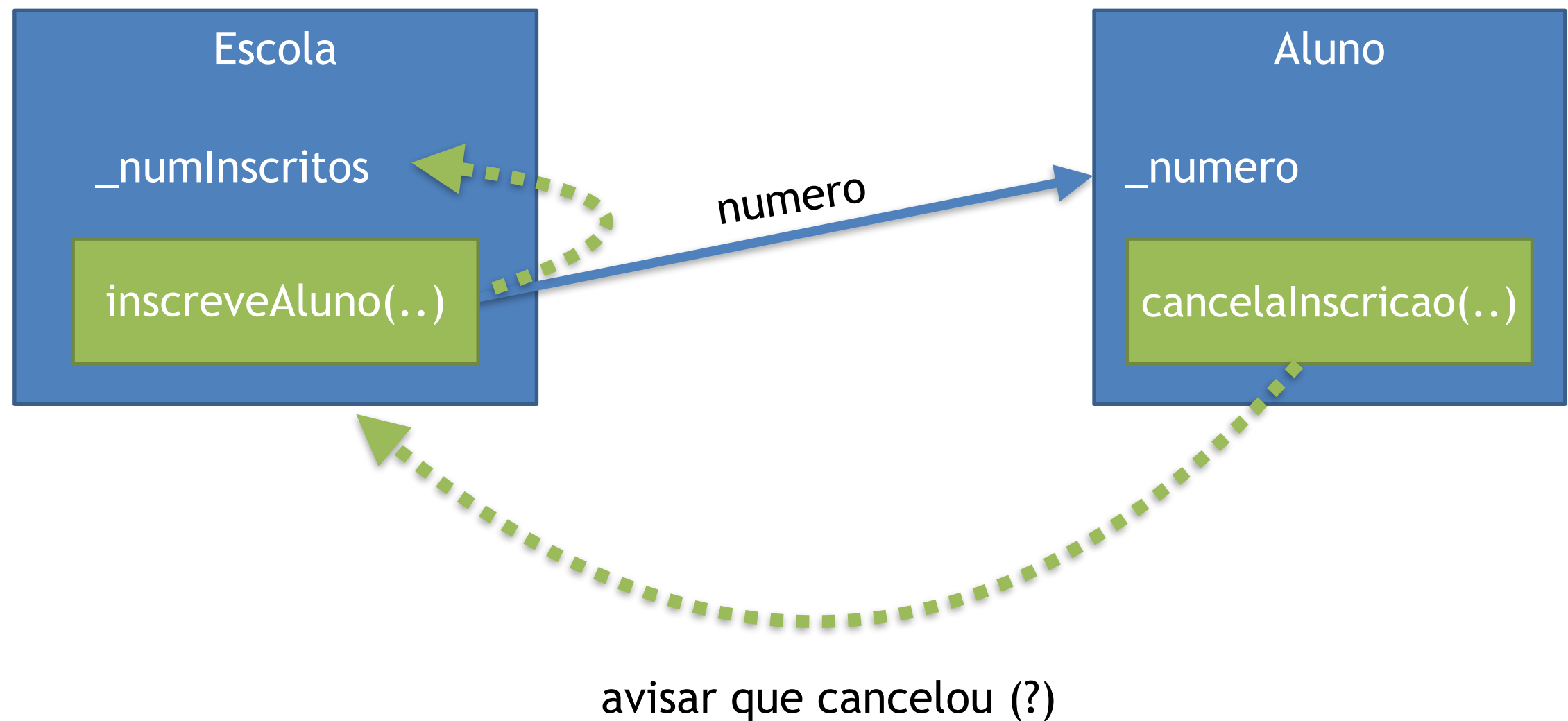
_numAlunosInscritos não é atualizado quando o aluno cancela a inscrição

Como “avisar” a escola que o aluno cancelou a inscrição?

Callbacks

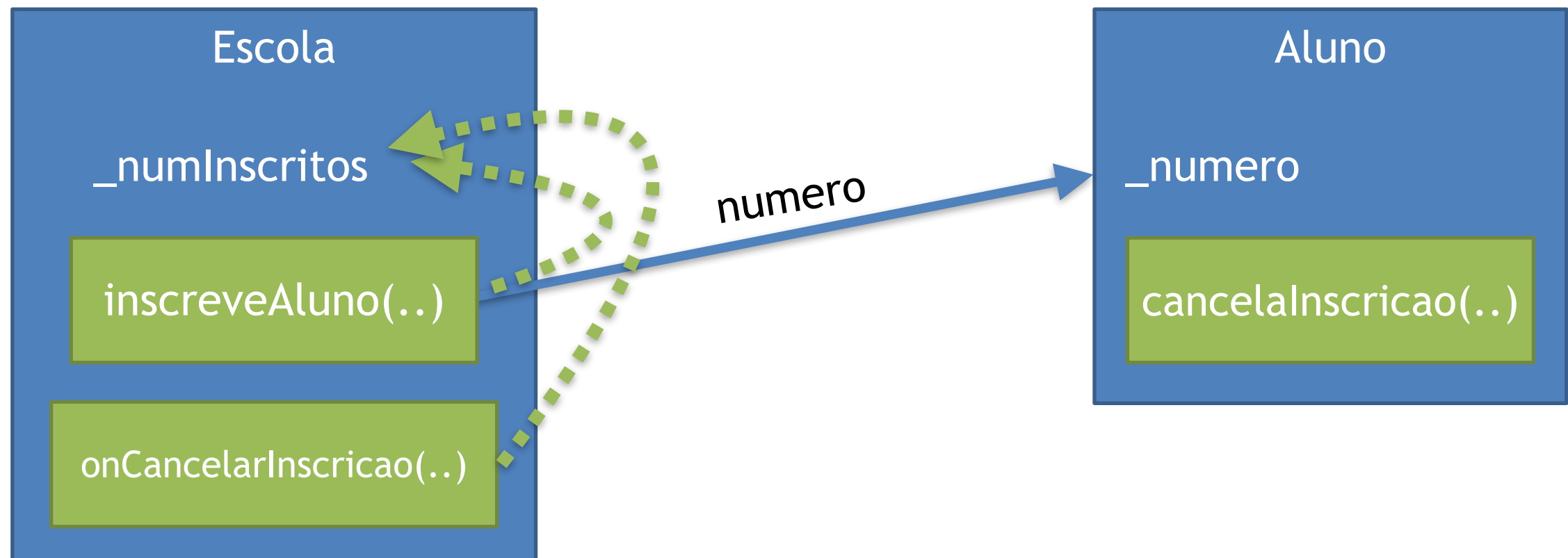


Callbacks



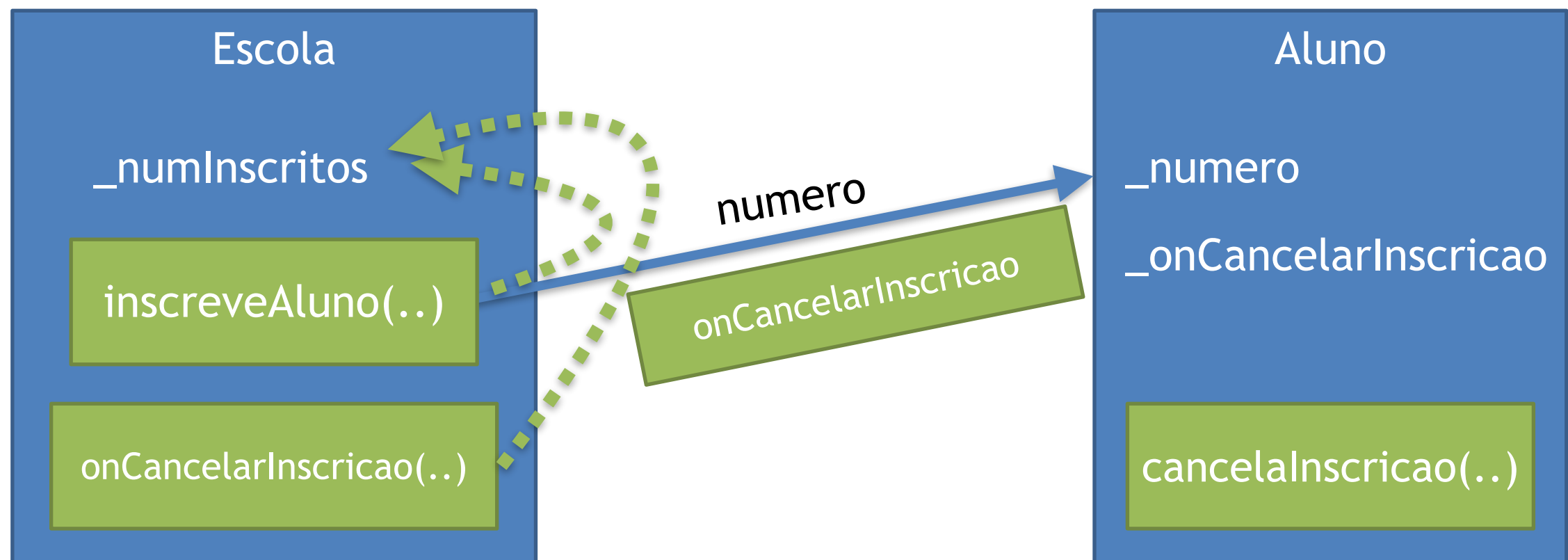
Callbacks

1. Criar uma função `onXXX` que será executada sempre que XXX ocorre



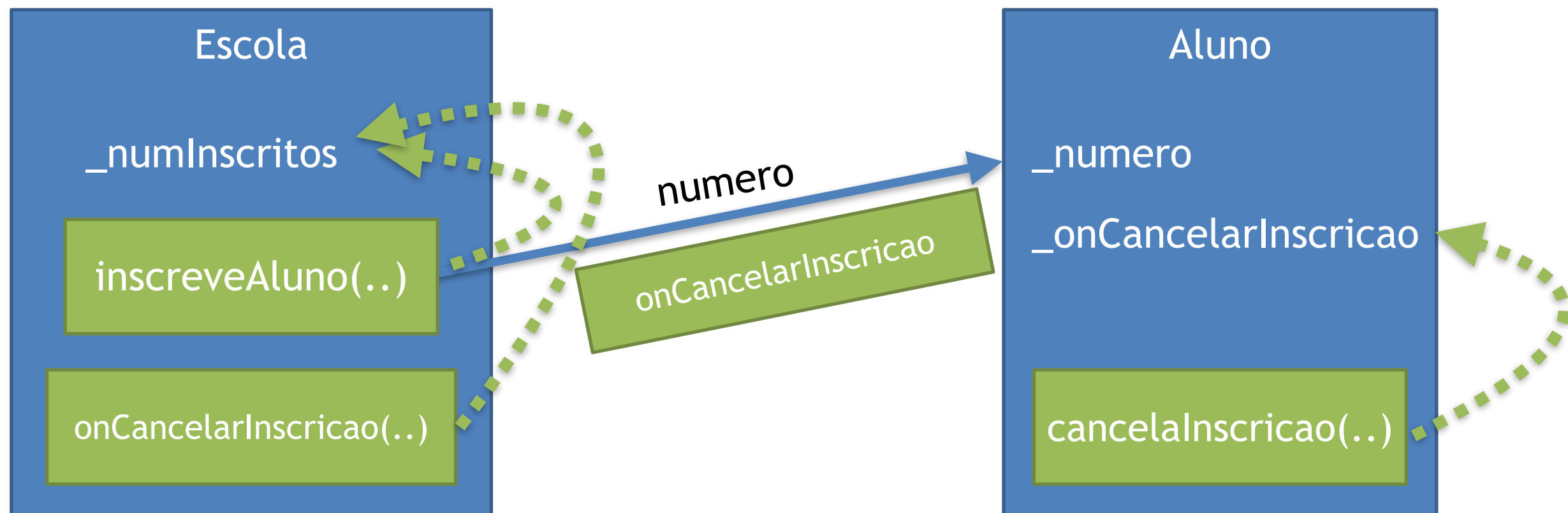
Callbacks

2. Passar essa função ao objecto que faz XXX



Callbacks

3. Executar essa função quando acontece XXX



Callbacks

1. Criar uma função `onXXX` que será executada sempre que XXX ocorre

```
class Escola {  
    int _numAlunosInscritos = 0;  
  
    Aluno inscreveAluno(int numero) {  
        _numAlunosInscritos++;  
        return Aluno(numero);  
    }  
  
    void onCancelarInscricao() {  
        _numAlunosInscritos--;  
    }  
}
```




Callbacks

2. Passar essa função ao objecto que faz XXX

```
class Escola {  
    int _numAlunosInscritos = 0;  
  
    Aluno inscreveAluno(int numero) {  
        _numAlunosInscritos++;  
        return Aluno(numero, onCancelarInscricao);  
    }  
  
    void onCancelarInscricao() {  
        _numAlunosInscritos--;  
    }  
}
```

Callbacks

3. Executar essa função quando acontece XXX

```
class Aluno {  
    int _numero;  
    bool _inscrito = true;  
    Function _onCancelarInscricao;   
  
    Aluno(this._numero, this._onCancelarInscricao);   
  
    void cancelarInscricao() {  
        _inscrito = false;  
        _onCancelarInscricao();   
    }  
}
```



```

class Escola {
    int _numAlunosInscritos = 0;

    void onCancelarInscricao() {
        _numAlunosInscritos--;
    }

    Aluno inscreveAluno(int numero) {
        _numAlunosInscritos++;
        return Aluno(numero, onCancelarInscricao);
    }
}

class Aluno {
    int _numero;
    bool _inscrito = true;
    Function _onCancelarInscricao;

    Aluno(this._numero, this._onCancelarInscricao);

    void cancelaInscricao() {
        _inscrito = false;
        _onCancelarInscricao();
    }
}

void main() {
    Escola escola = Escola();
    Aluno aluno123 = escola.inscreveAluno(123);
    // ...
    aluno123.cancelaInscricao();
}

```

Callbacks

(programa completo)

```

class A {
    int _x = 0;
    int _y = 2;

    void onX(int i) {
        _x += i;
    }

    void onY() {
        _y *= 2;
    }

    B criaB(int i) {
        return B(i, onX, onY);
    }
}

```

```

class B {
    int _i;
    Function(int) _onX;
    Function _onY;

    B(this._i, this._onX, this._onY);

    void fazCoisas() {
        _onX(_i);
        _onY();
    }
}

```

Exercício

Qual o valor de _x e _y no final do programa?

```

void main() {
    A a = new A();
    B b1 = a.criaB(3);
    B b2 = a.criaB(4);
    b1.fazCoisas();
    b1.fazCoisas();
    b2.fazCoisas();
}

```

```

class A {
    int _x = 0;
    int _y = 2;

    void onX(int i) {
        _x += i;
    }

    void onY() {
        _y *= 2;
    }

    B criaB(int i) {
        return B(i, onX, onY);
    }
}

```

```

class B {
    int _i;
    Function(int) _onX;
    Function _onY;

    B(this._i, this._onX, this._onY);

    void fazCoisas() {
        _onX(_i);
        _onY();
    }
}

```

Resolução

```

_x = 10
_y = 16

```

```

void main() {
    A a = new A();
    B b1 = a.criaB(3);
    B b2 = a.criaB(4);
    b1.fazCoisas();
    b1.fazCoisas();
    b2.fazCoisas();
}

```