



Paradigmas de Programação

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br



Linguagem de Programação

Linguagem: qualquer meio sistemático de comunicar ideias ou sentimentos através de signos convencionais, sonoros, gráficos, gestuais etc.

Programação: é o ato da escrita dos códigos entendíveis pelos computadores, permitindo assim que eles realizem as funções pedidas.



Linguagem de Programação

A linguagem de programação é um método padronizado, formado por um conjunto de regras sintáticas e semânticas, de implementação de um código fonte - que pode ser compilado e transformado em um programa de computador ou usado como script interpretado - que informará instruções de processamento ao computador.



O que são paradigmas de programação?

Um paradigma pode ser entendido como um tipo de **estruturação** ao qual a **linguagem deverá respeitar**. A depender do objetivo proposto, a solução que a linguagem oferecerá obedece a um tipo de paradigma. Portanto, o que vai definir o paradigma utilizado será a tratativa dada ao problema.



O que são paradigmas de programação?

É como se você tivesse que resolver um simples problema durante a execução de uma mudança doméstica. Imagine que você tivesse que levar um piano do primeiro ao terceiro andar de sua residência. O problema, já se sabe qual é. Mas de qual forma (paradigma) resolver essa questão?



O que são paradigmas de programação?

Possíveis soluções:

- Pode-se optar por carregá-lo manualmente dois andares acima;
- Elevá-lo por meio de um carrinho;
- Elevar o piano pelo lado de fora da edificação utilizando um motor fixado no terceiro andar e acionado por controle remoto.



O que são paradigmas de programação?

A escolha da solução do problema anteriormente listado depende de múltiplos fatores como: custo, quantidade de pessoas disponíveis, tamanho do plano, estrutura do prédio, etc.

Todos os métodos de elevação tem o mesmo resultado, porém sempre tem algum melhor para o caso.



Por que aprender sobre paradigmas de programação?

Cada paradigma surgiu de necessidades diferentes. Dado isso, cada um apresenta maiores vantagens sobre os outros dentro do desenvolvimento de determinado sistema. Sendo assim, **um paradigma pode oferecer técnicas apropriadas para uma aplicação específica.**



Por que aprender sobre paradigmas de programação?

Escolhido o paradigma de desenvolvimento adequado ao projeto, isso permitirá que sejam desenvolvidas aplicações com grande produtividade. Haverá unicidade na orientação de escrita do código entre a equipe, tornando-o mais legível e criando facilidade de manutenção ao longo de sua existência.



Por que aprender sobre paradigmas de programação?

Além disso, manter o mesmo paradigma determinará o objetivo da pessoa que trabalha com programação sobre a composição da estrutura e execução do sistema. Isso permite que as técnicas adequadas sejam utilizadas no projeto em questão. É essencial manter essa metodologia durante todo o trabalho.



Por que aprender sobre paradigmas de programação?

Saber qual o tipo escolher na resolução de um problema quer dizer que antes de refletir sobre a solução de um problema, você pensará na modelagem dessa solução, ou seja, refletirá sobre o paradigma a ser utilizado.



Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



Paradigma Imperativo

Também é chamado de paradigma procedural. Nesse tipo de construção, **as instruções devem ser passadas ao computador na sequência em que devem ser executadas**. Vários tipos de linguagem de programação suportam esse tipo de paradigma, como Cobol, Fortran e Pascal.



Paradigma Imperativo

Nesse tipo de programação, a **pessoa** passará uma espécie de **passo-a-passo dos procedimentos** que a máquina deverá executar (daí o nome procedural). Nesse caso, a solução do problema será muito dependente da experiência e criatividade de quem trabalha com a programação. O foco da resolução estará em “**como**” deve ser feito.



Paradigma Imperativo

Esse tipo de programação é recomendada em projetos nos quais não se espera que haja mudanças significativas ao longo do tempo (programa estático), quando se tratar de uma operação complexa — em que os diferentes estados da aplicação necessitem de visualização explícita—, ou quando não existiam muitos elementos compartilhados.



Paradigma Imperativo

Esse paradigma tem a vantagem de ser eficiente e de permitir uma modelagem tal qual o mundo real, além de ser bem estabelecido e bastante flexível. Por outro lado, o código fonte gerado é de difícil legibilidade.

Parad

Essa
modelagem
é bastante
legível

```
C 1 2 3 4 5 6
C234567890123456789012345678901234567890123456789012345
PROGRAM BASKHARA

C
REAL :: A,B,C, DELTA, X1,X2, RE, IM

C
PRINT *, "Este programa resolve uma equação de 2o.grau"
PRINT *, "no formato: a*x**2 + b*x + c = 0"

C
PRINT 10, "Digite a, b, c: "
10 FORMAT( A, 1X, $)
20 READ(*, *, ERR=20)A, B, C

C
DELTA= B**2-4.0*A*C

C
IF( DELTA.GT.0 )THEN      ! (DUAS RAÍZES REAIS)
  X1= ( -B-SQRT(DELTA) ) / ( 2.0*A )
  X2= ( -B+SQRT(DELTA) ) / ( 2.0*A )
  PRINT *, "RAIZES: X1= ", X1
  PRINT *, "X2= ", X2
ELSE
  IF( DELTA.EQ.0 ) THEN ! (DUAS RAÍZES REAIS IGUAIS)
    X1= -B / ( 2.0*A )
    X2= X1
    PRINT *, "RAIZES: X1=X2= ", X1
  ELSE      ! (DUAS RAÍZES COMPLEXAS)
    RE= -B / ( 2.0*A )
    IM= SQRT( -DELTA ) / ( 2.0*A )
    PRINT *, "RAIZES COMPLEXAS: X1= ", RE,"- ", IM, "i"
    PRINT *, "X2= ", RE, "+ ", IM, "i"
  END IF
END IF

C
END PROGRAM BASKHARA
```

Obter uma
resposta
é difícil



Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



Paradigma Declarativo

O paradigma declarativo, por sua vez, foca mais em “o quê” deve ser resolvido do que, necessariamente, em “como” isso deve ser feito. Sendo assim, o nível de abstração é maior, ao passo que a pessoa que o programa tem a preocupação em descrever determinada sequência lógica e qual o resultado esperado. Logo, o foco deixa de ser como esse resultado deve ser computado.



Paradigma Declarativo

Nesse paradigma, há **declarações iniciais de verdades lógicas que são imutáveis**. Assim, depois de realizadas algumas interações entre elas, o resultado encontrado será sempre igual (para as declarações outrora feitas). Isso entra em contraste com a programação imperativa, na qual um mesmo trecho de código pode retornar resultados diferentes.



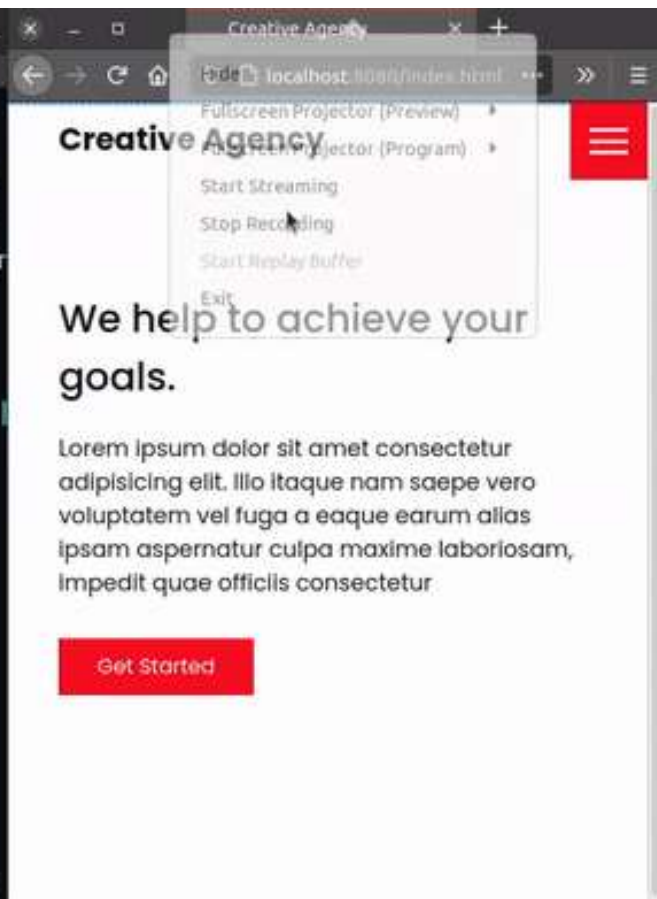
Paradigma Declarativo

O nível de abstração aqui é muito maior e as principais linguagens declarativas são também de marcação: HTML, XML, XSLT e XAML.

```
[Extension Development Host] - index.html - creative-agency-website - Visual Studio Code ...
File Edit Selection View Go Run Terminal Help

index.html x contact.html services.html

index.html > html > head
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6   <meta name="viewport" content="width=device-width, initial-scal
7   <link rel="stylesheet" href="css/style.css" />
8   <title>Creative Agency</title>
9 </head>
10 <body>
11   <header>
12     <div class="logo">Creative Agency</div>
13     <div class="toggle"></div>
14   </header>
15
16   <section class="home">
17     
18     <div class="home-content">
19       <h1>We help to achieve your goals.</h1>
20       <p>
21         Lorem ipsum dolor sit amet consectetur adipisicing elit.
22         eaque earum alias ipsam aspernatur culpa maxime laboriosa
23       </p>
24       <a href="contact.html" class="btn">Get Started</a>
25     </div>
26   </section>
27
28   <script src="js/script.js"></script>
29 </body>
```





Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



Paradigma Funcional

No paradigma de programação funcional, o uso de funções é destaque (daí seu nome). O problema é dividido em blocos e, para sua resolução, são implementadas funções que definem variáveis em seu escopo e retornam algum resultado. São exemplos de linguagens suportadas por esse paradigma o LISP, o Scheme, JavaScript e o Haskell.



Paradigma Funcional

É bastante indicado quando a solução requerida é fortemente dependente de uma base matemática. Assim, subdivide-se o problema proposto e as funções implementadas farão os cálculos matemáticos. Ao final, a pessoa programadora deve também integrar a solução entregue.



Paradigma Funcional

O paradigma funcional tem alocação de memória automática. Isso elimina possíveis “efeitos colaterais” nos cálculos matemáticos das funções. Assim, sua recursividade pode ter vários formatos, podendo ser uma técnica mais eficiente que a implementação de laços da programação imperativa.

```
;;; Common Lisp Examples.
```

```
(defun hello-world ()
  "Print 'hello, world' message."
  (format t "hello, world~%"))
```

```
(defun factorial (n)
  "Compute factorial of n."
  (if (zerop n)
      1
      (* n (factorial (- n 1)))))
```

```
(defun fibonacci (n)
  "Compute nth Fibonacci number."
  (if (< n 2)
      n
      (+ (fibonacci (- n 1)) (fibonacci (- n 2)))))
```

```
; SLIME 2.26.1
```

```
CL-USER> (hello-world)
```

```
hello, world
```

```
NIL
```

```
CL-USER> (factorial 6)
```

```
720
```

```
CL-USER> (trace factorial)
```

```
(FACTORIAL)
```

```
CL-USER> (factorial 6)
```

```
0: (FACTORIAL 6)
```

```
1: (FACTORIAL 5)
```

```
2: (FACTORIAL 4)
```

```
3: (FACTORIAL 3)
```

```
4: (FACTORIAL 2)
```

```
5: (FACTORIAL 1)
```

```
6: (FACTORIAL 0)
```

```
6: FACTORIAL returned 1
```

```
5: FACTORIAL returned 1
```

```
4: FACTORIAL returned 2
```

```
3: FACTORIAL returned 6
```

```
2: FACTORIAL returned 24
```

```
1: FACTORIAL returned 120
```

```
0: FACTORIAL returned 720
```

```
720
```

```
CL-USER> (fibonacci 6)
```

```
8
```

```
CL-USER> 
```



Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



Paradigma Lógico

O paradigma lógico é um tanto distinto dos demais paradigmas e deriva do declarativo. Fundamentalmente, **utiliza formas de lógica simbólica como padrões de entrada e saída**. A partir daí, realiza inferências para produzir os resultados.



Paradigma Lógico

Para exemplificar bem seu uso, podemos imaginar a tentativa de prova de um dado teorema, na qual são explicitadas algumas premissas e, sendo elas verdadeiras, a conclusão de tal teorema torna-se verdade também.



Paradigma Lógico

Dentre as linguagens de programação que utilizam esse paradigma, podemos citar QLISP, Mercury e Prolog — esta última sendo a mais popular de todas. São utilizadas na solução de problemas que envolvem inteligência artificial, criação de programas especialistas e comprovação de teoremas.


```
1 ?- consult(person).
```

```
% person compiled 0.00 sec, 11 clauses
```

```
true.
```

```
2 ?- forall( (person(Name,_,_,Height), Height>1.70),  
|           writeln([Name,Height])).
```

```
[Bob,1.92]
```

```
[Eva,1.72]
```

```
[Joe,1.79]
```

```
[Meg,1.75]
```

```
[Sam,1.95]
```

```
[Tom,1.72]
```

```
[Zoe,1.71]
```

```
true.
```

```
3 ?-
```



Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



Paradigma Orientado o Objetos

Esse paradigma é bastante conhecido. Foi popularizado na década de 90 com a linguagem de programação Java, **ao permitir uma programação multiplataforma da mesma maneira**. Antes disso, não era possível realizar tal tipo de trabalho.



Paradigma Orientado o Objetos

O paradigma orientado a objetos surgiu como uma grande aposta para resolver gargalos da indústria de software, como produzir programas de forma mais rápida, com maior confiabilidade e a um custo menor. Para isso, buscou apoiar-se nas características de classe e objeto ao tentar retratar a programação tal qual se enxerga o mundo real.



Paradigma Orientado o Objetos

Segundo esse paradigma, todos os objetos têm determinados estados e comportamentos. Esses estados são descritos pelas classes como atributos. Já a forma como eles se comportam (sua funcionalidade) é definida por meio de métodos, que são equivalentes às funções do paradigma funcional.



Paradigma Orientado o Objetos

Para que uma linguagem de programação seja do tipo de paradigma orientado a objetos, deve implementar seus três alicerces básicos, que são conceito de herança, polimorfismo e encapsulamento. Alguns exemplos de linguagens orientadas a objetos são Java, C++, C# e Python.

```
package rentalStore;
import java.util.Enumeration;
import java.util.Vector;

class Customer {
    private String _name;
    private Vector<Rental> _rentals = new Vector<Rental>();

    public Customer(String name) {
        _name = name;
    }

    public String getMovie(Movie movie) {
        Rental rental = new Rental(new Movie("", Movie.NEW_RELEASE), 10);
        Movie m = rental._movie;
        return movie.getTitle();
    }

    public void addRental(Rental arg) {
        _rentals.addElement(arg);
    }

    public String getName() {
        return _name;
    }
}
```



Quais os principais paradigmas de programação?

Basicamente, existem seis principais tipos de paradigmas. A depender da linguagem de programação utilizada, pode-se usar mais de um deles.

- Imperativo;
- Declarativo;
- Funcional;
- Lógico;
- Orientado a Objetos;
- Orientado a Eventos.



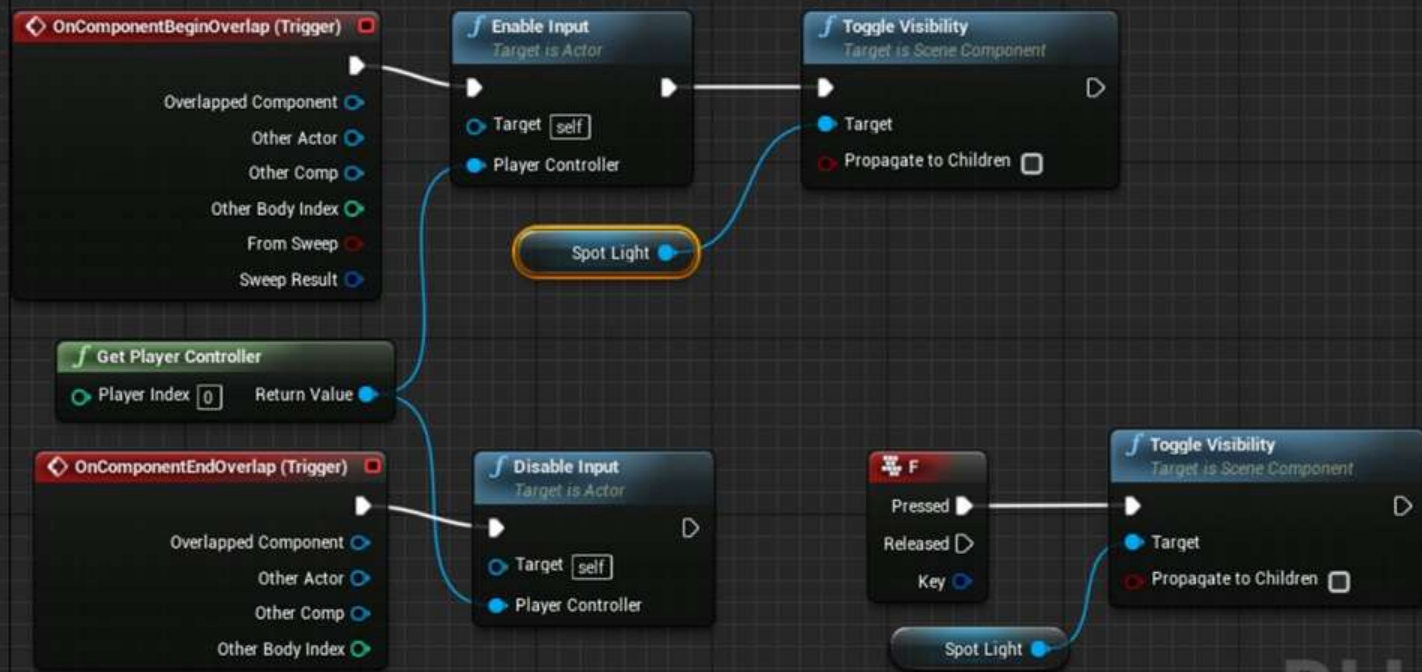
Paradigma Orientado a Eventos

O paradigma de orientação a eventos é **usado por toda linguagem de programação que tem uso de recursos gráficos, como jogos e formulários**. Dessa forma, a execução do programa se dá a medida que determinados eventos são disparados pelo usuário. Portanto, quem usa é responsável pelo momento em que o programa é executado.



Paradigma Orientado a Eventos

Imagine uma caixa de formulário que precisa do preenchimento do usuário. Os eventos descritos no código fonte serão executados à medida que se realiza o preenchimento dos campos solicitados. Além disso, ocorrem execuções também quando se decide enviar os dados clicando no botão de envio. São eventos que disparam outros eventos.



BLUEPRINT



TUTORIAL SERIES



Referências

- Noletto, C. Paradigmas de programação: o que são e quais os principais?. Disponível em : <<https://blog.betrybe.com/tecnologia/paradigmas-de-programacao/>> . Acesso em : 01/07/2022, 13:30:00.
- Silva E. Quais são os paradigmas de programação mais importantes?. Disponível em : <<https://blog.geekhunter.com.br/quais-sao-os-paradigmas-de-programacao/>>. Acesso em: 01/07/2022, 12:30:00.
- House, D. Paradigmas de programação: saiba quais são as mais usadas e como solucionar problemas. Disponível em: <<https://www.digitalhouse.com/br/blog/paradigmas-de-programacao/>>. Acesso em: 01/07/2022, 15:30:00.