

Do Paradigma ao Resultado:

Como Diferentes Abordagens Resolvem
os Mesmos Problemas de Formas Distintas



Um Problema: Lig4





Implementações

Poo

```
Scanner scanner = new Scanner(System.in);  
BoardPoo board = new BoardPoo();  
VictoryManager victoryManager = new VictoryManager();  
Player player1 = new Player(id: 1);  
Player player2 = new Player(id: 2);
```

Funcional

Uso de objetos imutáveis (*record*)

Sem mutação de estado

Nova instância a cada jogada

Foco no “*o que fazer*”, não “*como*”

Facilidade para testes e manutenção

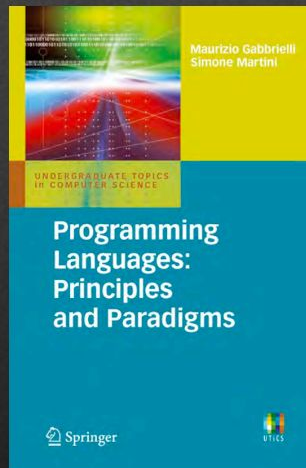
```
record BoardFuncional(int[][] grid) {
```

```
board = board.playMove(currentPlayer, column);  
board.display();
```

```
BoardFuncional playMove(int player, int column) {  
    int[][] newGrid = copyGrid();  
  
    OptionalInt rowToPlay = IntStream.iterate(seed:5, r -> r >= 0, r -> r - 1)  
        .filter(r -> newGrid[r][column] == 0)  
        .findFirst();  
  
    rowToPlay.ifPresent(r -> newGrid[r][column] = player);  
  
    return new BoardFuncional(newGrid);  
}
```

"[...] e o paradigma que resulta disso é o paradigma funcional. Esse paradigma enfatiza a definição e aplicação de funções, evitando o uso de dados mutáveis e efeitos colaterais."

Capítulo 11, Seção 11.1



Imperativo

Uso de loops tradicionais (`for`, `while`)

Mutação direta de estado

Controle passo-a-passo da execução

Objetos com estado interno mutável

Foco no “como fazer” (instruções), não no resultado final

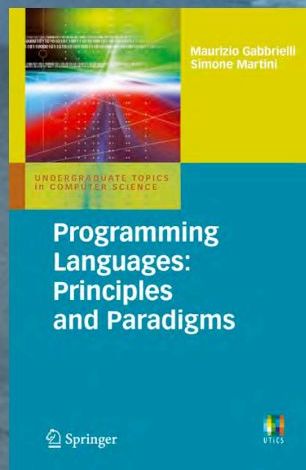
Mais difícil de testar com precisão

```
public void playMove(int player, int column) {  
    for (int row = 5; row >= 0; row--) {  
        if (grid[row][column] == 0) {  
            grid[row][column] = player;  
            break;  
        }  
    }  
}
```

```
private boolean checkHorizontal(int player) {  
    for (int row = 0; row < 6; row++) {  
        for (int col = 0; col <= 3; col++) {  
            if (grid[row][col] == player && grid[row][col + 1] == player  
                && grid[row][col + 2] == player && grid[row][col + 3] == player) {  
                return true;  
            }  
        }  
    }  
    return false;  
}
```


"[...] Segundo esse modelo, a variável é vista como uma espécie de recipiente ou local, ao qual se pode dar um nome e que contém valores [...]. Esses valores podem ser alterados ao longo do tempo, por meio da execução de comandos de atribuição"

Capítulo 6, Seção 6.2.1



Resultados:



Funcional

Escolha uma coluna de 1 a 7 para jogar (Jogador 1): 5

1 2 3 4 5 6 7

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|2|2|2|0|0|0|0|

|1|1|1|0|1|0|0|

Tempo total para verificar vitória: 12,058600 ms

Escolha uma coluna de 1 a 7 para jogar (Jogador 2): 5

1 2 3 4 5 6 7

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|2|2|2|0|2|0|0|

|1|1|1|0|1|0|0|

Tempo total para verificar vitória: 0,974800 ms

Imperativo (Original Refatorado)

Escolha uma coluna de 1 a 7 para jogar (Jogador 2): 5

1 2 3 4 5 6 7

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|2|2|2|0|2|0|0|

|1|1|1|0|1|0|0|

Tempo total para verificar vitória: 0,002400 ms

Escolha uma coluna de 1 a 7 para jogar (Jogador 1): 4

1 2 3 4 5 6 7

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|2|2|2|0|2|0|0|

|1|1|1|1|1|0|0|

Tempo total para verificar vitória: 0,002000 ms

Programação Orientada a Objetos

Escolha uma coluna de 1 a 7 para jogar (Jogador 1): 5

1 2 3 4 5 6 7

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|2|2|2|0|0|0|0|

|1|1|1|0|1|0|0|

Tempo total para verificar vitória: 0,022000 ms

Escolha uma coluna de 1 a 7 para jogar (Jogador 2): 5

1 2 3 4 5 6 7

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|0|0|0|0|0|0|0|

|2|2|2|0|2|0|0|

|1|1|1|0|1|0|0|

Tempo total para verificar vitória: 0,005000 ms

A knight in full plate armor, including a red plumed helmet, stands in shallow, dark water. He is facing a massive, multi-headed Hydra monster. The Hydra has several heads, each with a crown-like crest and a long, serpentine neck. The scene is dimly lit, with a bright light source on the left creating a reflection on the water's surface. The Hydra's heads are positioned around the knight, some looking directly at him. The overall atmosphere is dark and ominous.

Outro problema: Hydra

Módulo Futebol



Módulo Basquete



Módulo Vôlei



Hydra

Módulo Futebol



Módulo Basquete



Módulo Vôlei



Microsoft Visual Studio Solution File, Format Version 12.00

```
Project("{2150E333-8FDC-42A3-9474-1A3956D46DE8}") = "src", "src", "{C0C516B7-0D41-43AC-8D8A-78FFB73497B4}"
```

```
EndProject
```

```
Project("{2150E333-8FDC-42A3-9474-1A3956D46DE8}") = "tests", "tests", "{0A69FFCD-A8FA-481A-B772-CFB18B43BC1B}"
```

```
EndProject
```

```
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Application", "src\Application\Application.csproj", "{DA30AF1E-3467-41D7-80C0-000000000000}"
```

```
EndProject
```

```
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Application.Tests", "tests\Application.Tests\Application.Tests.csproj", "{DA30AF1E-3467-41D7-80C0-000000000000}"
```

```
EndProject
```

.sln

Soluções:

Utilizar F#, linguagem naturalmente funcional, imutável por padrão;

Composição por funções puras (O módulo expõe só a função)

Build automatizado com script (sem .sln, apenas .fsproj)

Separação real de módulos (sem trocar o paradigma)



Soluções:



[Merge Request](#)



[Merge Request](#)

Referências

MICROSOFT. O que é F#. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/fsharp/what-is-fsharp>. Acesso em: 12 maio 2025.

GABRIELLI, Maurizio; MARTINI, Simone. Programming Languages: Principles and Paradigms. London: Springer-Verlag, 2010.