

# Arquitetura de SW - TP1

---

Alexandre Alencar

# Objetivo

Corrigir um *anti-pattern* em Javascript usando AST

# Motivação

- Anti-pattern em Javascript é algo comum
- Refatoração atrasa o ciclo de desenvolvimento
- Melhorar manutenibilidade
- Melhorar legibilidade

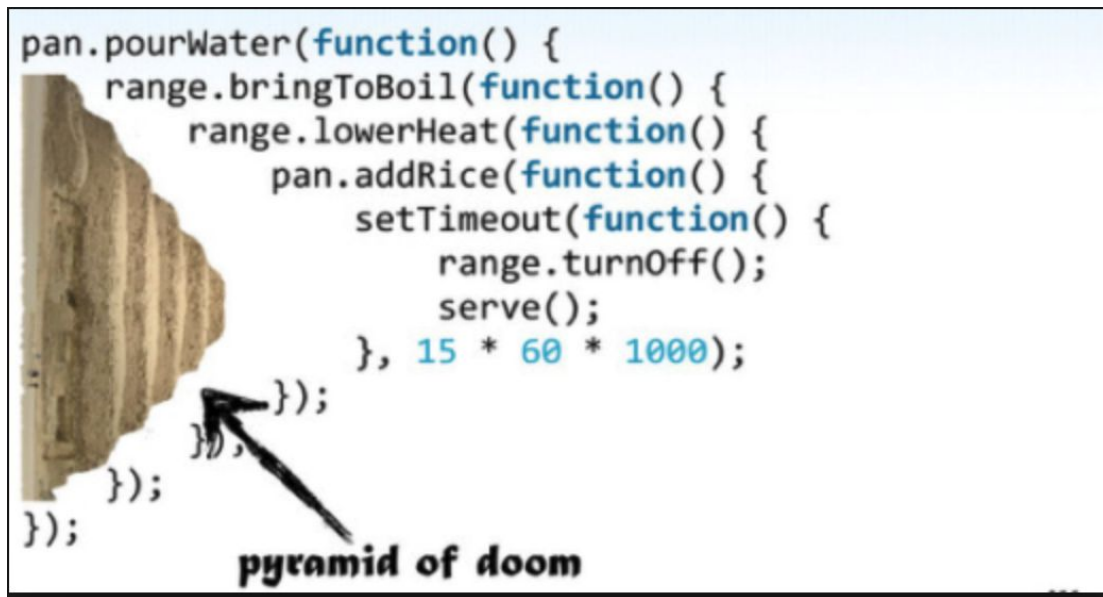
# Conceitos Básicos em JS

## Callbacks

```
1 function foo(x, fn) {  
2     return fn(x)  
3 }  
4  
5 foo(10, function(x) {  
6     return x * x;  
7 });
```

# Conceitos Básicos em JS

## Callback Hell / Pyramid of Doom



# Conceitos Básicos JS

## Promises (Anti-pattern)

```
2 // Node.js e Banco de Dados
3
4 consultarClientes('joão').then(function(cliente) {
5
6     cliente.consultarCompras().then(function(compras) {
7
8         compras.consultarBonus().then(function(bonus) {
9
10             console.log(bonus);
11
12         }).catch(errorHandler3);
13
14     }).catch(errorHandler2);
15
16 }).catch(errorHandler1);
```

# Resolução do problema

## Promises (Refatoradas)

```
1 consultarClientes('joão')
2   .then(function(cliente) {
3       return cliente.fetchCompras();
4   })
5   .then(function(compras) {
6       return compras.consultarBonus();
7   })
8   .then(function(bonus) {
9       console.log(bonus)
10  })
11  .catch(errorHandler1)
12  .catch(errorHandler2)
13  .catch(errorHandler3);
14
```

Obrigado!

---