

YOU AIN'T GONNA NEED  
IT (YAGNI)

(VOCÊ NÃO VAI PRECISAR DISSO)



GABRIEL, GUILHERME, ARTUR, ÍTALO

**O que sugere a  
orientação  
?**

Não adicionar  
funcionalidades ao  
código fonte de um  
programa até que  
estas sejam realmente  
necessárias.

---

# **SLIDE EXTRA**

**(caso precise de tempo)**

Na prática!

# Em quais situações devo aplicá-lo ?

1. **YAGNI** é um princípio por trás da prática do **XP** de "fazer a coisa mais simples que poderia funcionar"
2. **Deve** ser usado em combinação com **várias outras práticas**
3. Usada **sem refatoração** contínua



**Mas o que eu  
ganho com isso  
?**

"Se é barato fazer  
agora e barato  
depois, deixe para  
depois. Se é barato  
agora, mas ficará  
caríssimo depois, faça  
agora."

---

# PADROES DE PROJETO

## *SINGLETON*

QUANDO VOCÊ NECESSITA DE SOMENTE UMA  
INSTÂNCIA DA CLASSE

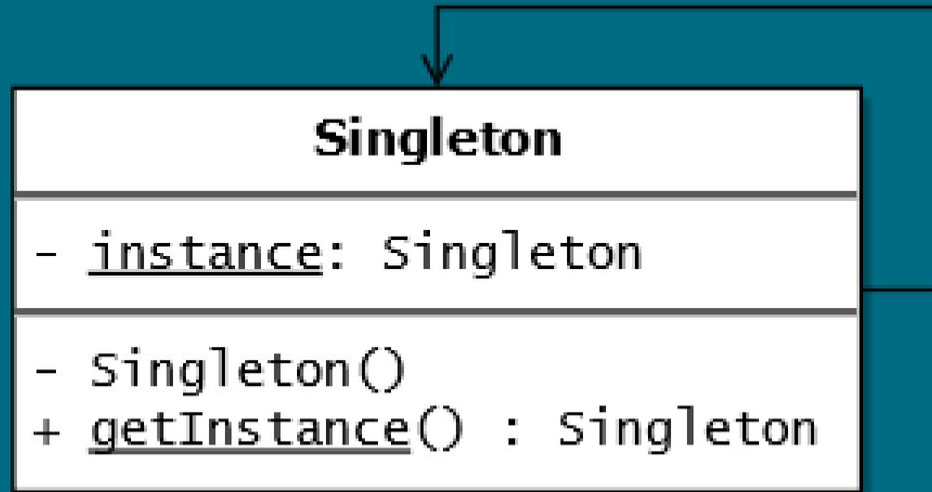


# O QUE SIGNIFICA

Singleton é um padrão de projeto de software (Design Pattern). Este padrão garante a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto.

O Padrão Singleton tem como definição garantir que uma classe tenha apenas uma instância de si mesma

# DIAGRAMA UML





# COMO IMPLEMENTAR

1. Deixar o construtor privado, pois assim ninguém deve conseguir instanciar a classe, apenas o próprio Singleton.
2. Criar um atributo privado e estático do mesmo tipo da classe (instance). Algumas linguagens não tipadas não irão precisar do tipo, caso do PHP, por exemplo.

# COMO IMPLEMENTAR

3. Método `getInstance()` é o principal ponto da classe. Ele verifica se a variável `instance` já foi iniciada, caso não tenha sido, ele faz sua criação pela primeira e única vez.

4. Para fazer a conexão, devemos chamar o `getInstance` da seguinte forma:  
`ClasseSingleton.getInstance()`

# ONDE USAR

## Banco de Dados:

Vamos supor que você terá que chamar diversas vezes a conexão com o banco de dados em um código na mesma execução, se você instanciar toda vez a classe de banco, haverá grande perda de desempenho, assim usando o padrão singleton, é garantida que nesta execução será instanciada a classe somente uma vez.

# Exemplo real de uso



```
1. public class Singleton {
2.     private static Singleton uniqueInstance;
3.     private String nomeDiretor;
4.     private String idadeDiretor;
5.
6.     //Construtor
7.     private Singleton(String nome, String idade) {
8.         this.nomeDiretor = nome;
9.         this.idadeDiretor = idade;
10.    }
11.    public static synchronized Singleton getInstance() {
12.        if (uniqueInstance == null)
13.            uniqueInstance = new Singleton("Fantini", "45 anos");
14.        return uniqueInstance;
15.    }
16. }
```

# THANK YOU

## Espaço para dúvidas

### REFERENCIAS YAGNI

<https://pt.wikipedia.org/wiki/YAGNI>

<https://martinfowler.com/bliki/Yagni.html>

### REFERENCIAS SINGLETON

[pt.stackoverflow.com](https://pt.stackoverflow.com)

<https://pt.wikipedia.org/wiki/Singleton>

<https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java>