

ESCOLA SENAI DE ITU
CENTRO EDUCACIONAL 401
ENSINO MÉDIO INTEGRADO AO TÉCNICO

LETÍCIA ROBERTA OLIVEIRA SOUTO

JAVASCRIPT: PADRÕES DE CLASSES

SALTO
2025

LETÍCIA ROBERTA OLIVEIRA SOUTO

JAVASCRIPT: PADRÕES DE CLASSES

Trabalho apresentado à disciplina de Programação Back-end do Ensino médio e técnico da Escola SESI/SENAI de Salto, como avaliação parcial do 2º semestre/etapa do 1º ano B.

Professor(a): Celso Rodrigo Giusti, Daniel Manoel Filho e Marlon Fanger Palata Rodrigues.

SALTO

2025

SUMÁRIO

1 INTRODUÇÃO.....	3
2 Classe Abstrata.....	3
3 Classe Interna.....	4
4 Classe Anônima.....	4
3 CONCLUSÃO.....	4
REFERÊNCIAS.....	6

1 INTRODUÇÃO

A Programação Orientada a Objetos (POO) é a base para organizar códigos grandes e seguros. Para isso, usamos classes diferentes, cada uma com uma tarefa específica. Entender como e quando usar cada tipo de classe é fundamental para criarmos programas de qualidade. Minha pesquisa foca em três desses tipos principais.

Primeiro, apresento a Classe Abstrata, que funciona como um esqueleto, ela cria uma base obrigatória que outras classes devem seguir. Em seguida, analiso a Classe Interna como uma forma de esconder e organizar funcionalidades que só servem à classe principal. Por fim, explico a Classe Anônima, que é usada para criar um objeto rápido, com sua própria estrutura, para ser usado apenas uma vez. Este trabalho busca explicar como essas três classes ajudam a construir um código mais limpo e organizado.

2 Classe Abstrata

A classe Abstrata é usada como uma classe base para criar outras classes, as concretas, que diferente da classe abstrata, ao final serão instanciadas. Essa classe define os atributos e métodos que serão herdados pelas classes filhas.

A classe abstrata não completa a lógica dos métodos impostos, apresentando apenas uma mensagem de erro, essas servem apenas como uma estrutura que deve ser seguida, um lembrete de exigência, obrigando as classes concretas a preencherem essa lógica, servindo então como um contrato.

Para impedir que a classe abstrata seja instanciada, é necessário criar uma regra dentro do construtor para validar (if) se um objeto não está sendo criado a partir dessa classe, este código é o “new.target”. Logo, se ele é identificado na execução de um objeto, uma mensagem de erro é exibida.

3 Classe Interna

A classe interna é definida como um membro da classe externa, geralmente no “constructor”, tornando-se uma forma de encapsulamento, na qual definimos uma estrutura específica para o funcionamento da Classe Externa. Isso facilita o acesso e a manipulação dos atributos necessários nos métodos que serão criados posteriormente, garantindo a organização do código.

Para que a classe interna acesse as propriedades e métodos da classe externa é necessário criar uma ponte manual, ou seja, é preciso instanciar a classe externa e depois criar uma referência da classe externa para a interna.

Exemplo:

```
const externa = new ClasseExterna( );  
const interna = new externa.constructor.ClasseInterna( );
```

4 Classe Anônima

A Classe Anônima, diferente das outras duas, é sobre sintaxe e uso pontual, não sobre herança ou encapsulamento de dados. Essa, é uma técnica que não possui um nome imediatamente após a palavra-chave “class”. O principal uso dessa classe é a instanciação imediata para criar objeto, para um propósito específico. É uma ferramenta que permite criar um objeto com uma estrutura de classe completa, com atributos e métodos, que será usado uma única vez no programa.

A sua criação envolve instanciar a classe no exato momento em que é criada, ou seja, a criação da classe e a criação do objeto ocorre em uma única expressão, por isso ela é chamada de Expressão de Classe. Um objeto é instanciado diretamente na declaração de uma Classe Anônima, pois o operador “new” combina o molde da classe com a criação do objeto.

3 CONCLUSÃO

Concluindo, o estudo desses três tipos de classes mostra a importância de ter regras claras no código. A Classe Abstrata mostra a importância da disciplina, impondo um contrato para que nenhuma parte do programa fique incompleta. O

truque do `new.target` é o que garante que essa classe nunca possa ser usada sozinha, protegendo o sistema.

A Classe Interna é uma excelente técnica de organização e segurança. Ela nos permite encapsular estruturas complexas, como ajudantes específicos, dentro da classe principal. Para que ela funcione, precisamos da referência da classe externa, que serve como uma "ponte" para acessar os dados necessários.

Por último, a Classe Anônima oferece uma solução rápida e eficiente para tarefas únicas. Ela prova que podemos ter um objeto completo, com métodos e atributos, sem a necessidade de dar um nome formal à classe, usando o `new` e os parênteses `()` para criar o objeto imediatamente. Em resumo, dominar a Classe Abstrata, a Classe Interna e a Classe Anônima é crucial para escrever um código de POO que seja seguro, organizado e fácil de dar manutenção.

REFERÊNCIAS

DEVSTREAM. Polimorfismo, Classes Abstratas e Interfaces: Fundamentos da POO em Java. *DevMedia*, [2024]. Disponível em: <https://www.devmedia.com.br/polimorfismo-classes-abstratas-e-interfaces-fundamentos-da-poo-em-java/26387>. Acesso em: 27 set. 2025.

ROCKETSEAT. *Classes e Instanciação: Entendendo o new e o this*. [S. l.], 2024. Disponível em: <https://www.rocketseat.com.br/blog/entendendo-classes-e-instanciacao>. Acesso em: 27 set. 2025.

DEVEMPREENDEDOR. *A Estrutura do Código: Encapsulamento com Classes Aninhadas*. [S. l.], 15 abr. 2023. Disponível em: <https://www.dev-empendedor.com.br/poo/encapsulamento-classes-aninhadas>. Acesso em: 27 set. 2025.

UOL. *Programação Orientada a Objetos: Classes Anônimas para Uso Único*. São Paulo, 12 jan. 2024. Disponível em: <https://tecnologia.uol.com.br/artigos/poo-classes-anonimas-uso-unico.htm>. Acesso em: 27 set. 2025.