



Técnicas de Programação I

Aula 01

Prof. Henrique Louro
henrique.louro@fatec.sp.gov.br



GOVERNO DO ESTADO
DE SÃO PAULO

Agenda

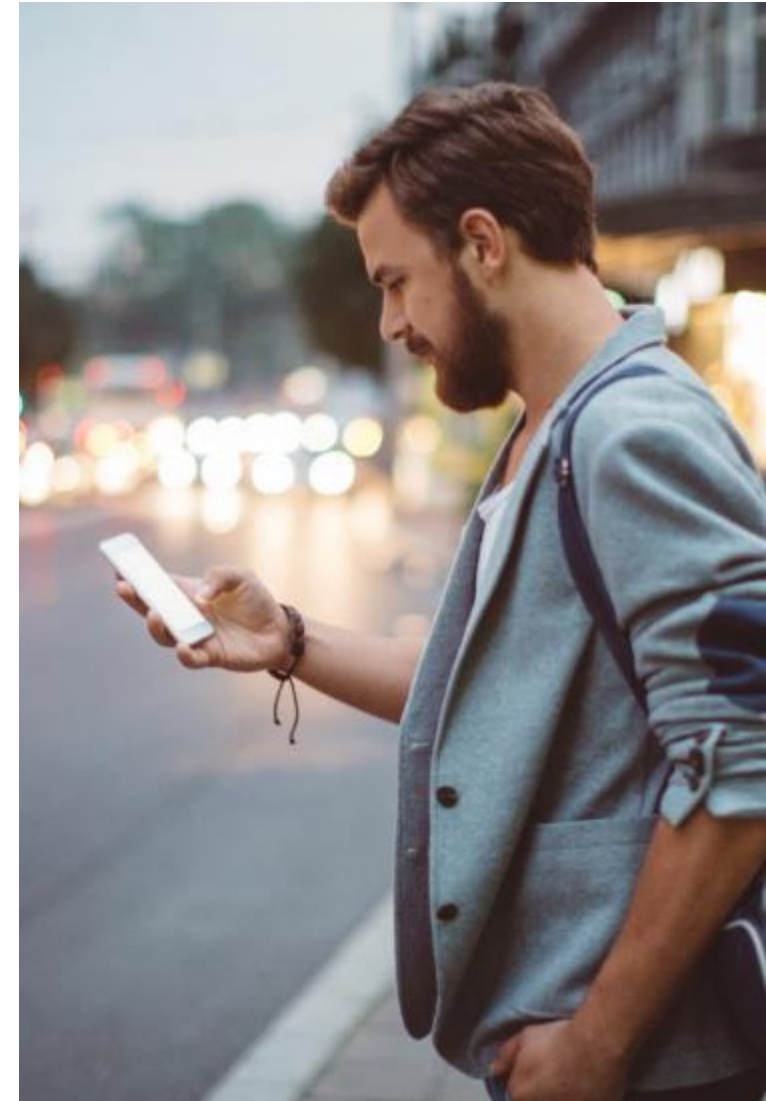
1) Apresentação do Professor

2) Apresentação dos Alunos

3) Apresentação da Disciplina

4) TypeScript

5) POO – Programação Orientada a Objetos





Professor



Henrique Duarte Borges Louro

Bacharel em Sistemas de Informação pelo Centro
Universitário Módulo

Pós-graduado em MBIS em Desenvolvimento de Sistemas
Web pelo Centro Universitário Módulo/Universidade
Cruzeiro do Sul

Licenciatura em Informática pelas Faculdades Chafic

Consultor, Analista e Programador (Desenvolvedor?)

Basic TRS80, Basic IBM-PC, Fortran, Cobol, DBASE II, DBASE III, Clipper,
Java, Python, C, C++, PHP, Javascript, TypeScript (?), Ajax, MySQL,
PostgreSQL, Apache Cassandra, etc



Alunos





Disciplina

- TypeScript
- POO – Programação Orientada a Objetos
- Tipos dados e Variáveis
- Função
- Abstração
- Objetos
- Classes
- Herança
- Polimorfismo
- Visibilidades
- Getters e Setters
- Modificadores, Static e Readonly
- Classe Abstrata
- Tipos Genéricos
- SGBD: Conexão e Persistência
- CRUD e MVC



A group of people in a meeting, with a woman in the foreground looking up and smiling, and others in the background holding up sticky notes.

DISCIPLINA

TypeScprit

1



TS

TypeScript

JavaScript com sintaxe para tipos

- Tipagem estática: as variáveis têm um tipo definido em tempo de compilação
- O TS adiciona a capacidade de definir tipos para variáveis, parâmetros de função, retornos de função e outros elementos do código, o que pode ajudar a detectar erros em tempo de desenvolvimento
- oferece suporte a recursos como classes, interfaces, herança, módulos e genéricos: POO



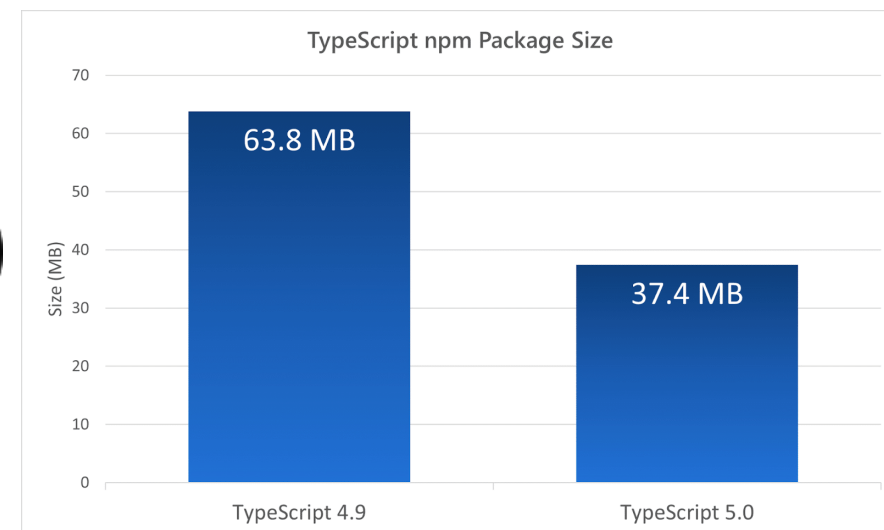
<https://www.typescriptlang.org/>



Compilação com tsc: código TypeScript transformado para JavaScript.



Lint: ferramenta que analisa o código-fonte em busca de erros, inconsistências, más práticas e problemas de estilo. (ESLint)



TS

TypeScript

TIPOS DE DADOS

LEVANTAMENTO DE REQUISITOS

- Requisito funcional é uma funcionalidade específica que o sistema deve ter, ou seja, uma ação que o sistema precisa ser capaz de realizar.
- Requisito não funcional refere-se a características ou qualidades do sistema, como desempenho, segurança, facilidade de uso, confiabilidade, e assim por diante.



```
let valor = 12; // o tipo de dado da variável é number
```

```
valor = "12"; // o tipo de dado da variável é string
```

- string: sequências de caracteres;
- number: números inteiros ou reais;
- boolean: somente os valores true e false;
- null: representa um valor nulo.
- undefined: representa um valor indefinido (não definido);
- bigint: representa números inteiros maiores do que $(2^{53} - 1)$ e menores do que $-(2^{53} - 1)$.



```
let nome:string = "Ana";  
let idade:number = 25; //número inteiro  
let peso:number = 59.9; //número real  
let doador:boolean = true; //booleano  
let fone:null = null; //endereço de memória  
let cel:undefined = undefined; //não definido  
let distancia:bigint = 20n; //o literal n é usado para indicar que o número é bigint
```

TS

TypeScript

TIPOS DE DADOS

INSTALAÇÃO

<https://nodejs.org/en/download>

`node -v`

`npm -v`

`npx -v`

`npm i typescript -g`

`tsc --version`

`npm -init -y`

`npm i -D ts-node typescript`

`tsc -init`

```
{
  "name": "exemplo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "ts-node ./src/index"
  },
}
```

Programação Orientada a Objetos

2



TypeScript

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

- **Abstração**
- **Objetos**
- **Classes**
- Herança
- Polimorfismo
- Visibilidades
- Getters e Setters
- Modificadores, Static e Readonly
- Classe Abstrata

- A abstração de objetos é um dos quatro pilares fundamentais da POO.
- Se refere ao processo de ocultar os detalhes de implementação e exibir apenas as funcionalidades do objeto ao usuário.
- A abstração permite que você se concentre no que um objeto faz em vez de como ele faz.
- Isso é realizado criando classes, que são modelos para objetos.
- Uma classe define os atributos e métodos que caracterizam qualquer objeto da classe.
- A abstração ajuda a reduzir a complexidade e permite o reuso de código.



TypeScript

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

- Abstração
- Objetos
- **Classes**
- Herança
- Polimorfismo
- Visibilidades
- Getters e Setters
- Modificadores, Static e Readonly
- Classe Abstrata

```
class Pessoa {  
  nome:string = "Ana";  
  idade:number = 18;  
}
```

```
const x = new Pessoa();
```

```
console.log("Nome:", x.nome);
```

```
console.log("Idade:", x.idade);
```



TypeScript

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

- Abstração
- Objetos
- **Classes**
- Herança
- Polimorfismo
- Visibilidades
- Getters e Setters
- Modificadores, Static e Readonly
- Classe Abstrata

Método: é uma função definida no corpo da classe. Os métodos são chamados de operações no contexto da POO.

No exemplo a seguir todos os objetos do tipo de dado Pessoa possuirão as propriedades nome e idade e o método imprimir:

```
class Pessoa {  
  nome:string = "Ana";  
  idade:number = 18;  
  imprimir(){  
    console.log(`${this.nome} possui  
    ${this.idade} anos`);  
  }  
}  
  
const x = new Pessoa();  
  
// chama o método imprimir do  
objeto que está na variável x  
x.imprimir();
```




TypeScript

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

- Abstração
- Objetos
- **Classes**
- Herança
- Polimorfismo
- Visibilidades
- Getters e Setters
- Modificadores, Static e Readonly
- Classe Abstrata

Construtor: é um "método especial" responsável por criar e inicializar objetos.

É executado automaticamente quando um objeto é criado a partir da classe.

Para criar uma instância (objeto) da classe, usamos a palavra-chave **new** seguida pelo nome da classe.

```
class Pessoa {  
    constructor(){  
        console.log("Construiu");  
    }  
}  
  
const x = new Pessoa();  
const y = new Pessoa();
```



TypeScript

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

- Abstração
- Objetos
- **Classes**
- Herança
- Polimorfismo
- Visibilidades
- Getters e Setters
- Modificadores, Static e Readonly
- Classe Abstrata

Instância da classe e objeto da classe são termos sinônimos no contexto da POO. A classe é um template para a construção de objetos/instâncias da classe. Um objeto/instância é uma “cópia da classe”.

Dentro do corpo de uma classe:

- Uma **variável** recebe o nome de **propriedade** ou **atributo**;
- Uma **função** recebe o nome de **método**.