



DIGITAL
INNOVATION
ONE



Fullstack Developer #2

Typescript - Início prático

Lucas Santos

Software Engineer - Take Blip

Objetivos da Aula



DIGITAL
INNOVATION
ONE



1. Aprender sobre a história e aplicação do Typescript;
2. Entender sobre linguagem transpilada e sua utilização;
3. Aprender a utilizar typescript em um projeto real.

Requisitos Básicos



DIGITAL
INNOVATION
ONE



- ✓ Programação orientada a objetos
- ✓ Conhecimento básico em JavaScript



DIGITAL
INNOVATION
ONE



Parte 1: História do Typescript



Fullstack Developer #2

História do Typescript



DIGITAL
INNOVATION
ONE



- Proveniente da evolução em relação ao uso do JavaScript;
- Criada pela Microsoft em 2012;
- Projeto de código aberto;
- Lançada em 2014 junto com o suporte para a linguagem no Visual Studio 2013.

Link para o projeto no github ->



Qual problema o TS resolve?



DIGITAL
INNOVATION
ONE



"A POO sempre foi um problema ao ser aplicada em JavaScript, devido a sua sintaxe não permitir escrever classes, por exemplo, de forma tão clara, além da fraca tipagem de dados. O TypeScript oferece então uma forma de corrigir ou contornar esses problemas"

- Devmedia

Conceitos do Typescript



DIGITAL
INNOVATION
ONE



- Tipagem estática opcional;
- Implementação de conceitos de POO;
- Possibilidade de validação de erros pré-compilação(ou pré-traspilação);
- Utiliza o compilador Babel para transformar o código em JavaScript;
- Integra completamente com a ECMAScript.

Curiosidade

Anders Hejlsberg



DIGITAL
INNOVATION
ONE



Quando usar Typescript?



DIGITAL
INNOVATION
ONE



Quando você tiver algumas dessas necessidades no seu projeto:

- Padronização;
- Escalabilidade;
- Utilização constante de tipos avançados;
- Necessidade de utilização de conceitos de POO.



DIGITAL
INNOVATION
ONE



Parte 2: O Typescript

Fullstack Developer #2

Tipos

- String
- Boolean
- Number
- Array
- Tuple
- Enum
- Any
- Void
- Unknown
- Never
- Function (callback)

Relembrando os conceitos iniciais de POO:

- Encapsulamento
- Herança
- Abstração
- Polimorfismo

Obs.: As técnicas do POO possibilitam a aplicação de muitos padrões de projeto como SOLID e GoF

```
interface IVeiculo{  
    ligar(): void;  
}  
  
class Veiculo{  
    private _chassi: string;  
  
    get Chassi(): string {  
        return this._chassi;  
    }  
  
    constructor(chassi: string) {  
        this._chassi = chassi;  
    }  
}
```

Encapsulamento/Herança



DIGITAL
INNOVATION
ONE



```
interface IVeiculo{  
    ligar(): void;  
}  
  
class Veiculo{  
    private _chassi: string;  
  
    get Chassi(): string {  
        return this._chassi;  
    }  
  
    constructor(chassi: string) {  
        this._chassi = chassi;  
    }  
}
```

```
class Carro extends Veiculo implements IVeiculo{  
    constructor(chassi:string) {  
        super(chassi);  
    }  
  
    ligar(){  
        console.log('liguei carro');  
    }  
}
```

Abstração/Polimorfismo



DIGITAL
INNOVATION
ONE



```
interface IVeiculo{  
    ligar(): void;  
}
```

```
class Carro extends Veiculo implements IVeiculo{  
    constructor(chassi:string) {  
        super(chassi);  
    }  
  
    ligar(){  
        console.log('liguei carro');  
    }  
}
```

```
class Moto extends Veiculo implements IVeiculo{  
    constructor(chassi:string) {  
        super(chassi);  
    }  
  
    ligar(){  
        console.log('liguei Moto');  
    }  
}
```

POO: Exemplos Avançados



DIGITAL
INNOVATION
ONE



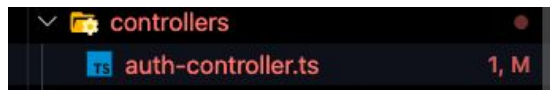
```
interface GenericInterface<U> {  
  value: U  
  getIdentity: () => U  
}  
  
class IdentityClass<T> implements GenericInterface<T> {  
  value: T  
  
  constructor(value: T) {  
    this.value = value  
  }  
  
  getIdentity () : T {  
    return this.value  
  }  
}
```

```
@Controller('auth')  
export class AuthController {  
  constructor(private _authService: AuthService) {}  
  
  @Post('login')  
  @HttpCode(200)  
  async login(@Body() loginData: LoginBody) {  
    try {  
      return await this._authService.login(  
        loginData.username,  
        loginData.password  
      );  
    } catch (error) {  
      throw this.getAuthError(error.message);  
    }  
  }  
}
```

Compilação



DIGITAL
INNOVATION
ONE





DIGITAL
INNOVATION
ONE



Parte 3: Mão na massa!

Fullstack Developer #2

Desafios



DIGITAL
INNOVATION
ONE



1. Implementar Botões de Editar e Excluir utilizando o gerenciador de estado atual;
2. Implementar funcionalidade de check de pagamentos;
3. SUPER: Implementar data no churras evoluindo para Rateio do Rolê;

Desafio 1



DIGITAL
INNOVATION
ONE



Tarefa: Implementar Botões de Editar e Excluir utilizando o gerenciador de estado atual.

Estória: Como ChurrasManager quero poder excluir e editar algum dado que foi inserido para maior controle das informações.

Desafio 2



DIGITAL
INNOVATION
ONE



Tarefa: Implementar funcionalidade de check de pagamentos.

Estória: Como ChurrasManager quero poder marcar quem já me pagou para ter mais controle de quem eu ainda devo cobrar.

Desafio 3 SUPER



DIGITAL
INNOVATION
ONE



Tarefa: Implementar data no churras evoluindo para Rateio do Rolê

Estória: Como ChurrasManager quero poder colocar quais dias a pessoa participou do rolê para que seja cobrado apenas o valor referente aos dias que ela participou do rolê.



DIGITAL
INNOVATION
ONE



Dúvidas?

Fullstack Developer #2

Achou que acabou ?



DIGITAL
INNOVATION
ONE



Dicas do Lucão:

- Autoconhecimento(Conheça a forma que você aprende melhor);
- Esteja atento a fontes de informação;
- Troque informações;
- Pratique.

Obrigado!



DIGITAL
INNOVATION
ONE



Vamos manter contato!



DIGITAL
INNOVATION
ONE



← Siga o CodingSA na Twitch



← Discord do CodingSA



Minhas redes



Fontes



DIGITAL
INNOVATION
ONE



Introdução ao Typescript → <https://www.devmedia.com.br/introducao-ao-typescript/36729>

TypeScript: O guia definitivo →

<https://oieduardorabelo.medium.com/typescript-o-guia-definitivo-1a63b04259cc>

O mínimo que você precisa saber sobre TypeScript →

<https://medium.com/@matheusmariano/o-mínimo-que-você-precisa-saber-sobre-typescript-58d1b418f78b>

Typescript - Tipos de dados → <https://bognarjunior.wordpress.com/2018/09/16/typescript-tipos-de-dados/>

Everyday Types → <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html>

TypeScript - Entendendo Generics por completo →

<https://oieduardorabelo.medium.com/typescript-entendendo-generics-por-completo-40a372aeaa5>