

una	Centro Universitário Uma – Campus Barreiro		
	CURSO: Análise e Desenvolvimento de Sistemas		
	UC: Usabilidade, Web, Mob, Games	Professores: Diego e Montanha	
	Atividade: Atividade Avaliativa		Data: 12/04/21
	Aluno (a):	Valor: 10,0	Nota:

Proposição – Crie um projeto e faça o formulário, que irá receber nome, CPF, CEP e endereço. Ao sair do CEP, o endereço deve ser automaticamente preenchido.

Os arquivos abaixo devem ser modificados ou acrescentados ao projeto:

Arquivo cliente.ts (Model – classe normal)

```
export class Cliente {
  Nome: string;
  CPF: string;
  CEP: string;
  Endereco: string;

  CPFValido(): boolean {
    if (this.CPF == null) {
      return false;
    }
    if (this.CPF.length != 11) {
      return false;
    }
    // tslint:disable-next-line:max-line-length
    if ((this.CPF == '0000000000') || (this.CPF == '1111111111') || (this.CPF == '2222222222') || (this.CPF == '3333333333') || (this.CPF == '4444444444') || (this.CPF == '5555555555') || (this.CPF == '6666666666') || (this.CPF == '7777777777') || (this.CPF == '8888888888') || (this.CPF == '9999999999')) {
      return false;
    }
    let numero: number = 0;
    let caracter: string = '';
    let numeros: string = '0123456789';
    let j: number = 10;
    let somatorio: number = 0;
    let resto: number = 0;
    let digito1: number = 0;
    let digito2: number = 0;
    let cpfAux: string = '';
    cpfAux = this.CPF.substring(0, 9);
    for (let i: number = 0; i < 9; i++) {
      caracter = cpfAux.charAt(i);
      if (numeros.search(caracter) == -1) {
        return false;
      }
    }
  }
}
```

```

        numero = Number(caracter);
        somatorio = somatorio + (numero * j);
        j--;
    }
    resto = somatorio % 11;
    digito1 = 11 - resto;
    if (digito1 > 9) {
        digito1 = 0;
    }
    j = 11;
    somatorio = 0;
    cpfAux = cpfAux + digito1;
    for (let i: number = 0; i < 10; i++) {
        caracter = cpfAux.charAt(i);
        numero = Number(caracter);
        somatorio = somatorio + (numero * j);
        j--;
    }
    resto = somatorio % 11;
    digito2 = 11 - resto;
    if (digito2 > 9) {
        digito2 = 0;
    }
    cpfAux = cpfAux + digito2;
    if (this.CPF != cpfAux) {
        return false;
    }
    else {
        return true;
    }
}
}

```

Arquivo app.module.ts

```

import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    AppRoutingModule,
    FormsModule
  ],

```

```

    providers: [],
    bootstrap: [AppComponent]
  })
  export class AppModule { }

```

Arquivo – app.component.ts

```

import { Component } from '@angular/core';
import { Cliente } from './cliente';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'form-tutorial';
  cliente: Cliente = new Cliente();

  constructor(private http: HttpClient) {

  }

  VerificaCEP() {
    this.http.get(`https://viacep.com.br/ws/${this.cliente.CEP}/json/`).toPromise().then(
      data => {
        const dado: any = data;
        this.cliente.Endereco = dado.logradouro;
      }
    );
  }

  salvar() {
    console.log(this.cliente);
  }
}

```

Arquivo – app.component.html

```

<div>
  <form>
    <label for="nome">Nome: </label>
    <input type="text" name="nome" [(ngModel)]="cliente.Nome">
    <label for="cpf">CPF: </label>
    <input type="text" name="cpf" [(ngModel)]="cliente.CPF">
    <label for="cep">CEP: </label>
    <input type="text" name="cep" [(ngModel)]="cliente.CEP" (blur)="VerificaCEP()">
    <label for="endereco">Endereço: </label>
    <input type="text" name="endereco" [(ngModel)]="cliente.Endereco">
    <div *ngIf="!cliente.CPFValido()">
      CPF Inválido
    </div>
  </form>
</div>

```

```
</div>
<div>
  <input type="button" (click)="salvar()" value="Salvar" [disabled]="!cliente.CPFValid
o()" />
</div>
</form>
</div>
```

ATENÇÃO

Em caso de trabalhos repetidos (copiados) os que estiverem iguais serão zerados.