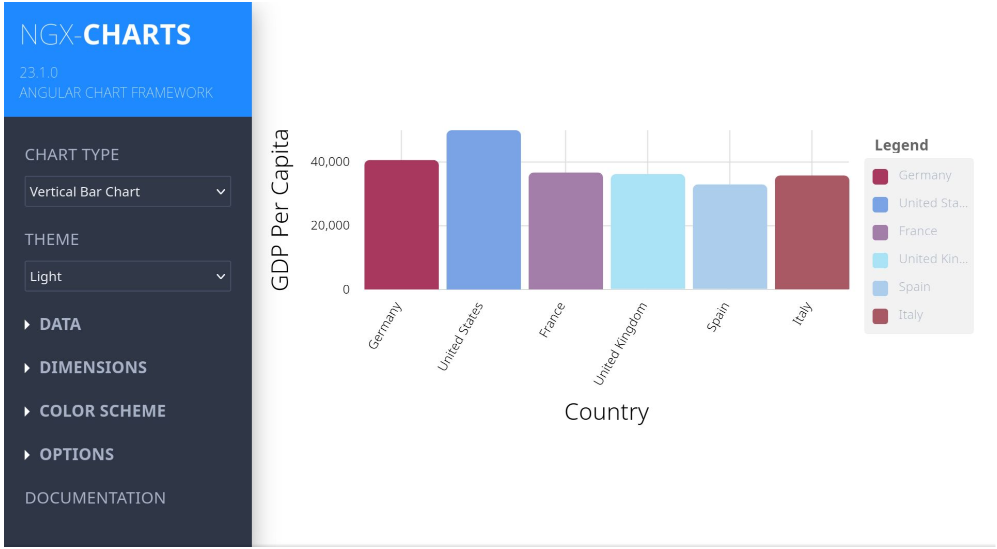
Refatoração smells Angular

Refatoração do projeto - ngx-charts, framework de gráficos declarativos

Sobre ngx-charts

É um framework de gráficos que combina o poder do Angular (para desenhar e animar) com a matemática do D3, permitindo visualizações bonitas, leves e totalmente customizáveis via CSS.



NGX-**CHARTS**

23.1.0

ANGULAR CHART FRAMEWORK

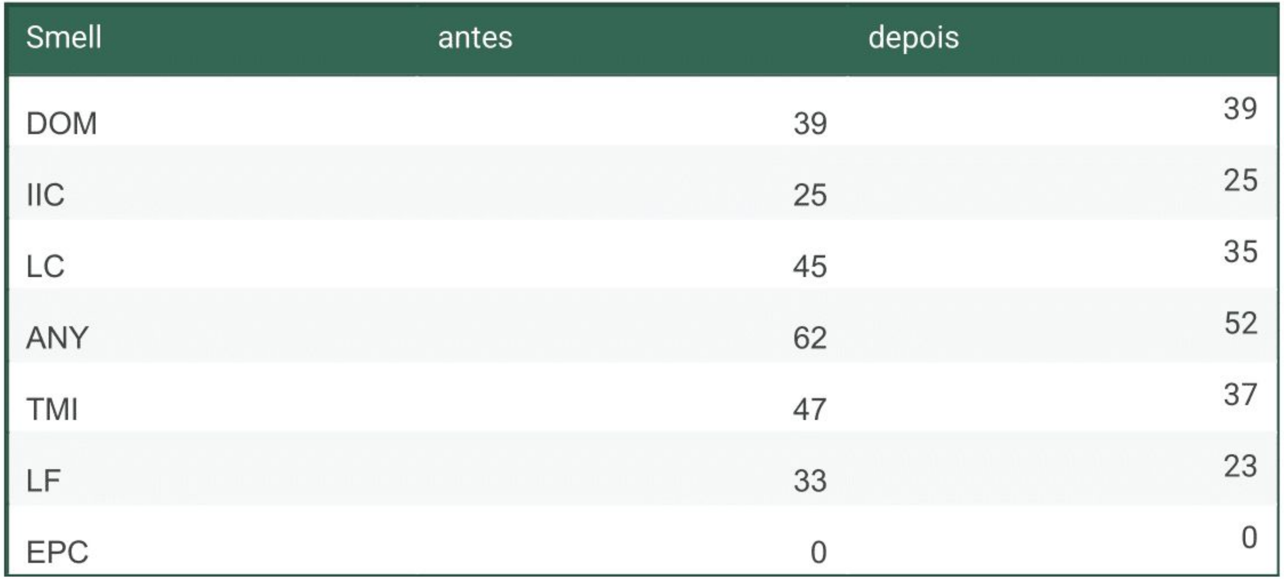
CHART TYPE

Vertical Bar Chart

THEME

Light

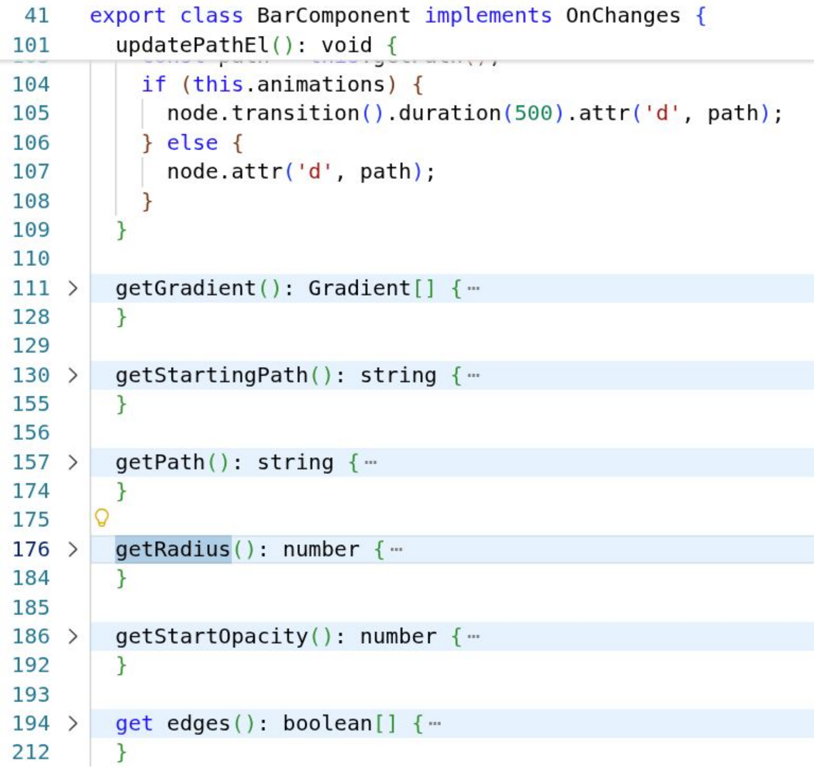
Contagem de smells



Smell	antes	depois
DOM	39	39
IIC	25	25
LC	45	35
ANY	62	52
TMI	47	37
LF	33	23
EPC	0	0

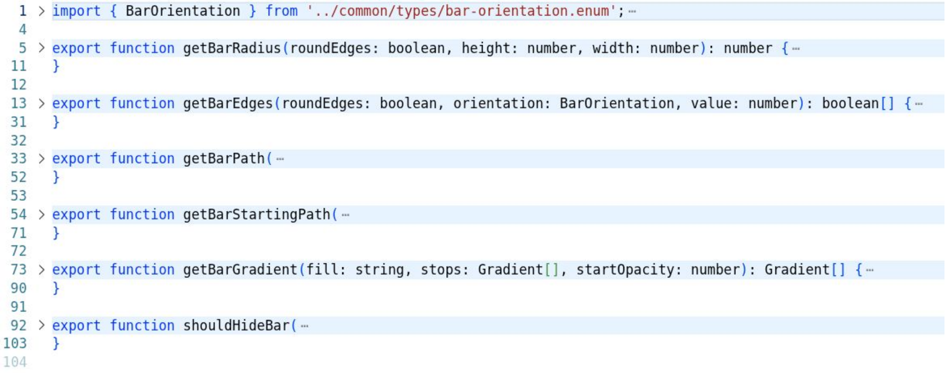
Smell LC

- Coesão
- Legibilidade

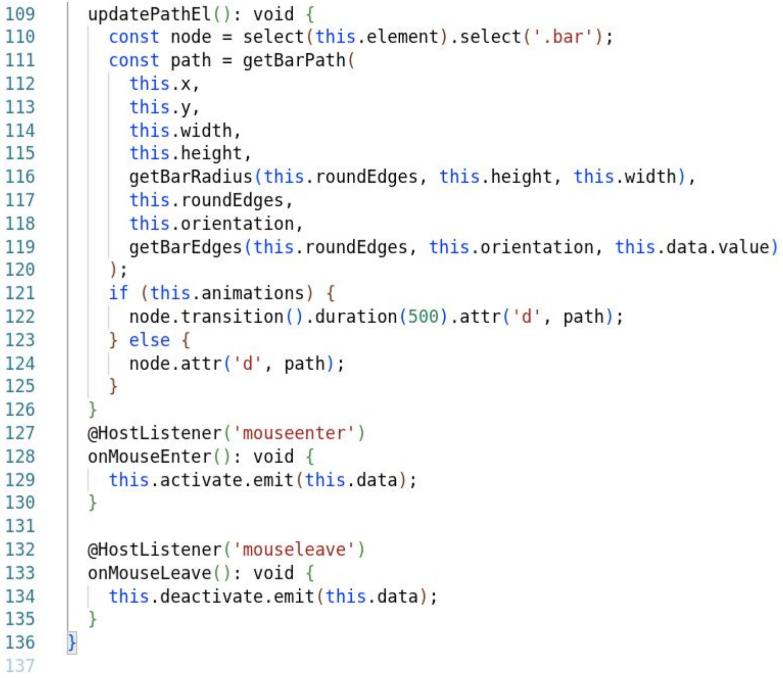


```
export class BarComponent implements OnChanges {
  updatePathEl(): void {
    const path = this.getPath();
    if (this.animations) {
      node.transition().duration(500).attr('d', path);
    } else {
      node.attr('d', path);
    }
  }

  getGradient(): Gradient[] {⋯
  }

  getStartingPath(): string {⋯
  }

  getPath(): string {⋯
  }

  getRadius(): number {⋯
  }

  getStartOpacity(): number {⋯
  }

  get edges(): boolean[] {⋯
  }
```

Exemplo de correção do LC



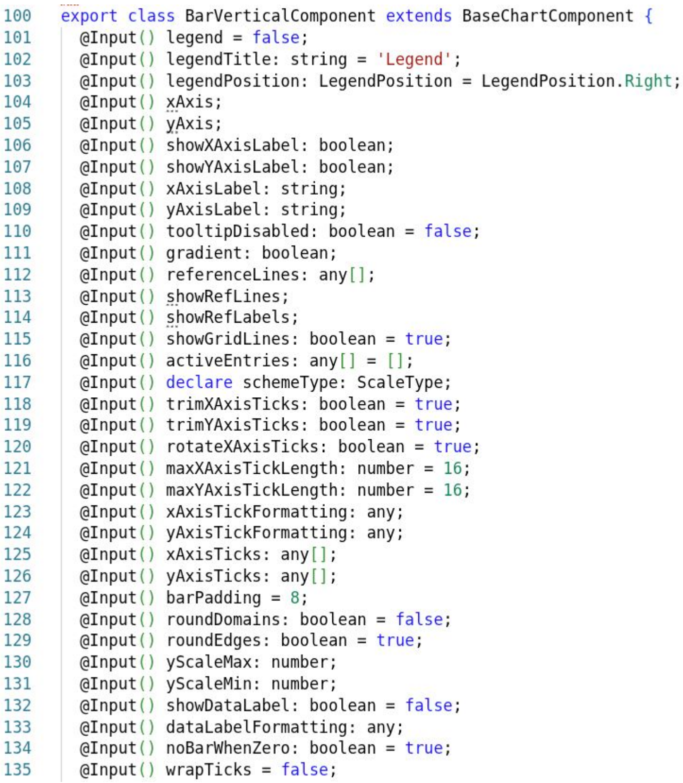
```
import { BarOrientation } from '../common/types/bar-orientation.enum';⋯

export function getBarRadius(roundEdges: boolean, height: number, width: number): number {⋯
}

export function getBarEdges(roundEdges: boolean, orientation: BarOrientation, value: number): boolean[] {⋯
}

export function getBarPath(⋯
}

export function getBarStartingPath(⋯
}

export function getBarGradient(fill: string, stops: Gradient[], startOpacity: number): Gradient[] {⋯
}

export function shouldHideBar(⋯
}
```



```
  updatePathEl(): void {
    const node = select(this.element).select('.bar');
    const path = getBarPath(
      this.x,
      this.y,
      this.width,
      this.height,
      getBarRadius(this.roundEdges, this.height, this.width),
      this.roundEdges,
      this.orientation,
      getBarEdges(this.roundEdges, this.orientation, this.data.value)
    );
    if (this.animations) {
      node.transition().duration(500).attr('d', path);
    } else {
      node.attr('d', path);
    }
  }
  @HostListener('mouseenter')
  onMouseEnter(): void {
    this.activate.emit(this.data);
  }

  @HostListener('mouseleave')
  onMouseLeave(): void {
    this.deactivate.emit(this.data);
  }
}
```

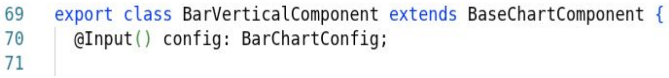
Smell TMI

- Legibilidade e poluição
- Acoplamento
- coesão

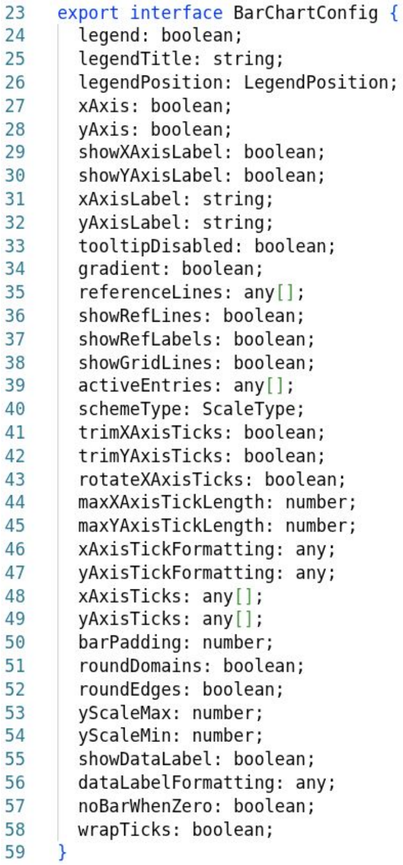


```
export class BarVerticalComponent extends BaseChartComponent {
  @Input() legend = false;
  @Input() legendTitle: string = 'Legend';
  @Input() legendPosition: LegendPosition = LegendPosition.Right;
  @Input() xAxis;
  @Input() yAxis;
  @Input() showXAxisLabel: boolean;
  @Input() showYAxisLabel: boolean;
  @Input() xAxisLabel: string;
  @Input() yAxisLabel: string;
  @Input() tooltipDisabled: boolean = false;
  @Input() gradient: boolean;
  @Input() referenceLines: any[];
  @Input() showRefLines;
  @Input() showRefLabels;
  @Input() showGridLines: boolean = true;
  @Input() activeEntries: any[] = [];
  @Input() declare schemeType: ScaleType;
  @Input() trimXAxisTicks: boolean = true;
  @Input() trimYAxisTicks: boolean = true;
  @Input() rotateXAxisTicks: boolean = true;
  @Input() maxXAxisTickLength: number = 16;
  @Input() maxYAxisTickLength: number = 16;
  @Input() xAxisTickFormatting: any;
  @Input() yAxisTickFormatting: any;
  @Input() xAxisTicks: any[];
  @Input() yAxisTicks: any[];
  @Input() barPadding = 8;
  @Input() roundDomains: boolean = false;
  @Input() roundEdges: boolean = true;
  @Input() yScaleMax: number;
  @Input() yScaleMin: number;
  @Input() showDataLabel: boolean = false;
  @Input() dataLabelFormatting: any;
  @Input() noBarWhenZero: boolean = true;
  @Input() wrapTicks = false;
```

Exemplo de correção TMI



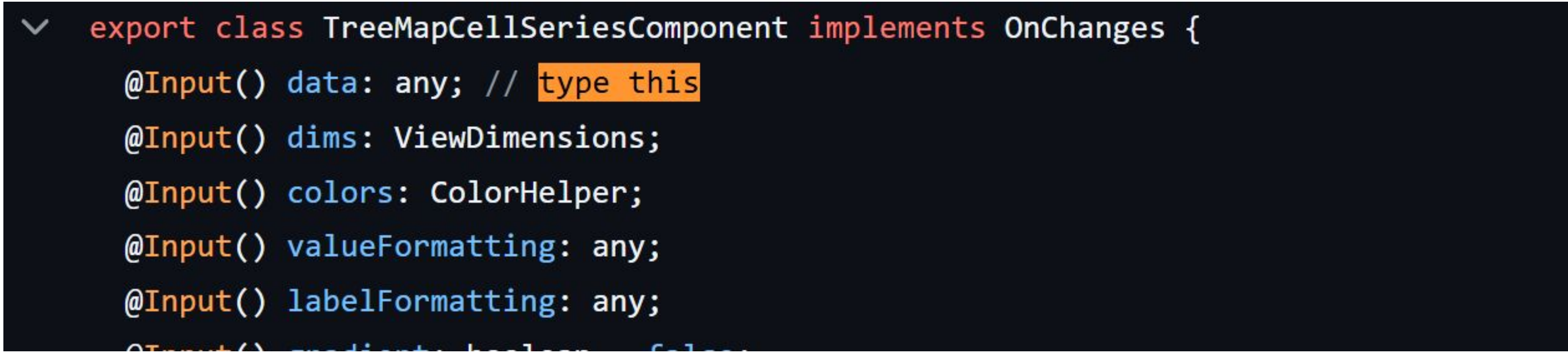
```typescript
export class BarVerticalComponent extends BaseChartComponent {
  @Input() config: BarChartConfig;

```



```typescript
export interface BarChartConfig {
  legend: boolean;
  legendTitle: string;
  legendPosition: LegendPosition;
  xAxis: boolean;
  yAxis: boolean;
  showXAxisLabel: boolean;
  showYAxisLabel: boolean;
  xAxisLabel: string;
  yAxisLabel: string;
  tooltipDisabled: boolean;
  gradient: boolean;
  referenceLines: any[];
  showRefLines: boolean;
  showRefLabels: boolean;
  showGridLines: boolean;
  activeEntries: any[];
  schemeType: ScaleType;
  trimXAxisTicks: boolean;
  trimYAxisTicks: boolean;
  rotateXAxisTicks: boolean;
  maxXAxisTickLength: number;
  maxYAxisTickLength: number;
  xAxisTickFormatting: any;
  yAxisTickFormatting: any;
  xAxisTicks: any[];
  yAxisTicks: any[];
  barPadding: number;
  roundDomains: boolean;
  roundEdges: boolean;
  yScaleMax: number;
  yScaleMin: number;
  showDataLabel: boolean;
  dataLabelFormatting: any;
  noBarWhenZero: boolean;
  wrapTicks: boolean;
}
```

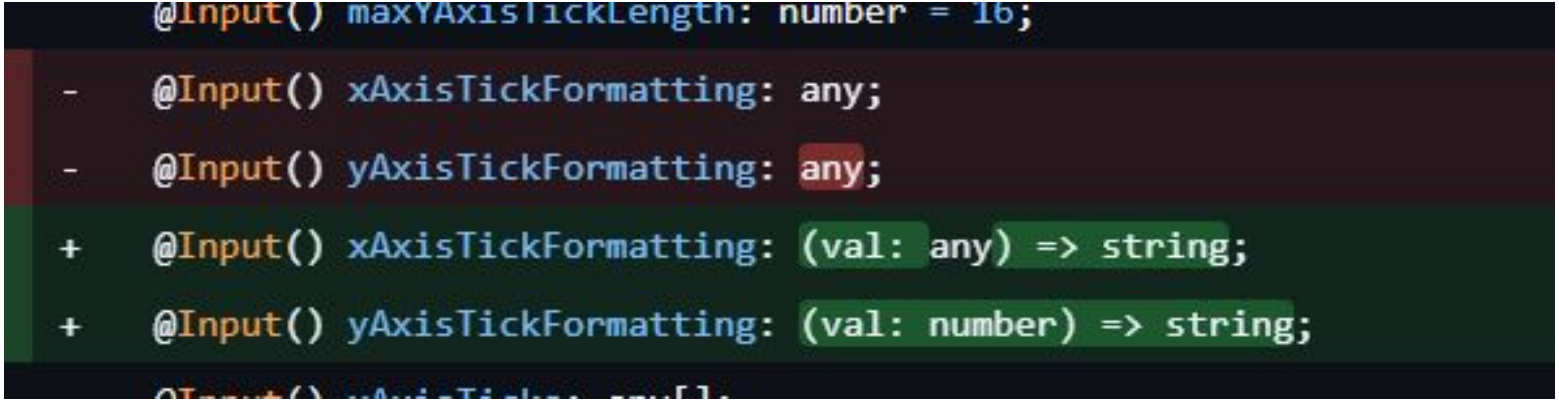
Smell Any Overusing

O uso de any desliga a verificação de tipos do TypeScript, permitindo que qualquer valor seja atribuído ou acessado, o que esconde bugs e dificulta o uso do IntelliSense



```
export class TreeMapCellSeriesComponent implements OnChanges {
  @Input() data: any; // type this
  @Input() dims: ViewDimensions;
  @Input() colors: ColorHelper;
  @Input() valueFormatting: any;
  @Input() labelFormatting: any;
  @Input() gradient: boolean = false;
```

Exemplo Correção Any Overusing



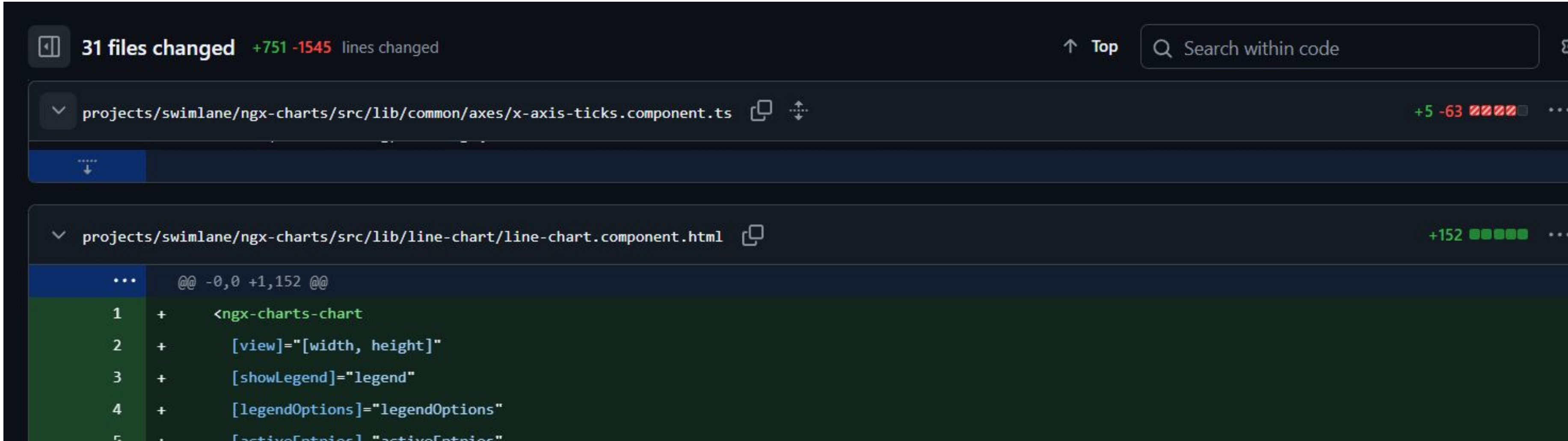
```
    @Input() maxYAxisTickLength: number = 16;
-   @Input() xAxisTickFormatting: any;
-   @Input() yAxisTickFormatting: any;
+   @Input() xAxisTickFormatting: (val: any) => string;
+   @Input() yAxisTickFormatting: (val: number) => string;
    @Input() yAxisTicks: any[];
```

Smell Large File

Ocorre quando um único arquivo de código cresce excessivamente em número de linhas.

Exemplo de correção Large File

remoção de 152 linhas de html do arquivo do line-chart.component.ts e criação do arquivo line-chart.component.html



projects/swimlane/ngx-charts/src/lib/common/axes/x-axis-ticks.component.ts +5 -63

projects/swimlane/ngx-charts/src/lib/line-chart/line-chart.component.html +152

```
@@ -0,0 +1,152 @@
+     <ngx-charts-chart
+       [view]="[width, height]"
+       [showLegend]="legend"
+       [legendOptions]="legendOptions"
+       [activeEntries]="activeEntries"
```

Conclusão

Dificuladades:

Paulo Victor:

1. Pouca experiência com o angular
2. dificuldade em trabalhar em um projeto antigo

Tiago:

1. Pouca experiência com frameworks frontend javascript
2. Conciliar essa entrega com as das outras cadeiras