PUC Minas

Covering

Alunos:

- Arthur Amaral
- Guilherme Antônio

Orientadores:

- José Laerte Xavier
- Marco Rodrigo Costa

Categorização

CBsoft

Área de Teste de Software e Engenharia de Software

Desenvolvedores

Benefícios aos desenvolvedores de software durante o processo de desenvolvimento e software

Características

Qualidade da construção de testes medindo a cobertura do projeto e exibindo em tempo real dados de teste

Objetivos

Covering

Padronização das entregas	Configuração do repositório
Velocidade no desenvolvimento de testes	Extensão para o VSCode
Maior cobertura de testes	Qualidade dos testes
Melhor identificação dos trechos sem cobertura	Navegação dentro da IDE
Fácil visualização das métricas geradas	Exibição da porcentagem

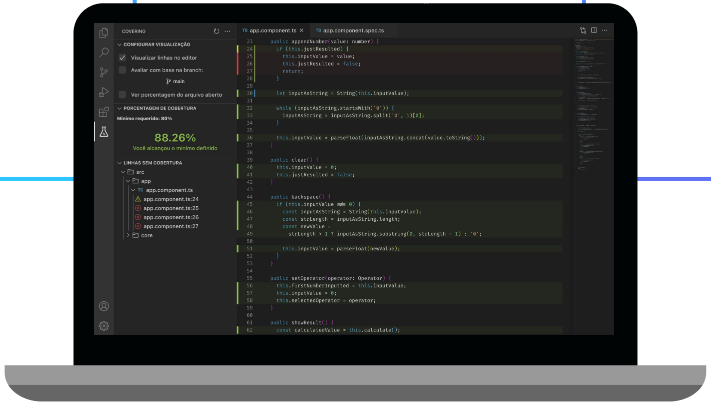
Escopo

Extensão pra o VSCode

Leitura de arquivo de teste

Destaque das linhas

Vizualização de cobertura



COVERING

CONFIGURAR VISUALIZAÇÃO

[✓] Visualizar linhas no editor

[ ] Avaliar com base na branch: main

[ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

88.26%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

- src
  - app
    - app.component.ts
      - app.component.ts:24
      - app.component.ts:25
      - app.component.ts:26
      - app.component.ts:27
  - core

app.component.ts | app.component.spec.ts

```typescript
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

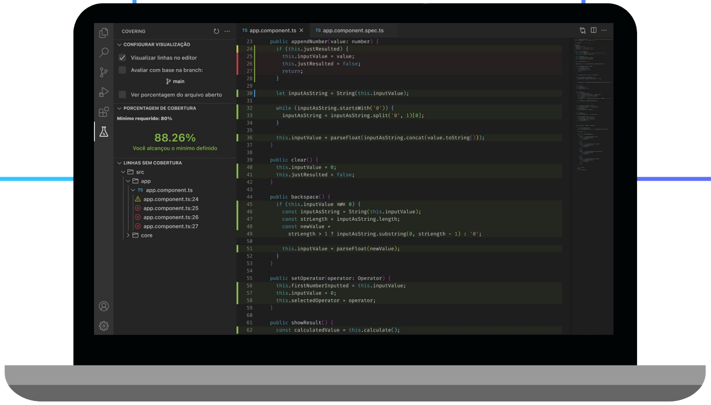
Escopo

Comparação de conbertura entre *branches*

Bloqueio de *push*

Configuração da extensão

Suporte oficial às linguagens Javascript e Typescript



- [✓] Visualizar linhas no editor
- [ ] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

Mínimo requerido: 80%

88.26%
Você alcançou o mínimo definido

```
src
  app
    app.component.ts
      app.component.ts:24
      app.component.ts:25
      app.component.ts:26
      app.component.ts:27
  core
```

```
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

Fora do Escopo

Gerar o arquivo de relatório de cobertura

Propor ações para incrementar cobertura de teste

Funcionamento fora do VSCode

Principais Necessidades e Funcionalidades

01
Visualização dos dados de cobertura atual

02
Visualização dos dados de cobertura com base na diferença entre versões de código

03
Definição de padrões de cobertura de teste para o projeto

04
Exibição de linhas de código sem cobertura

05
Controle da execução dos testes através de interface gráfica que independa da biblioteca utilizada

Descrição dos Atores

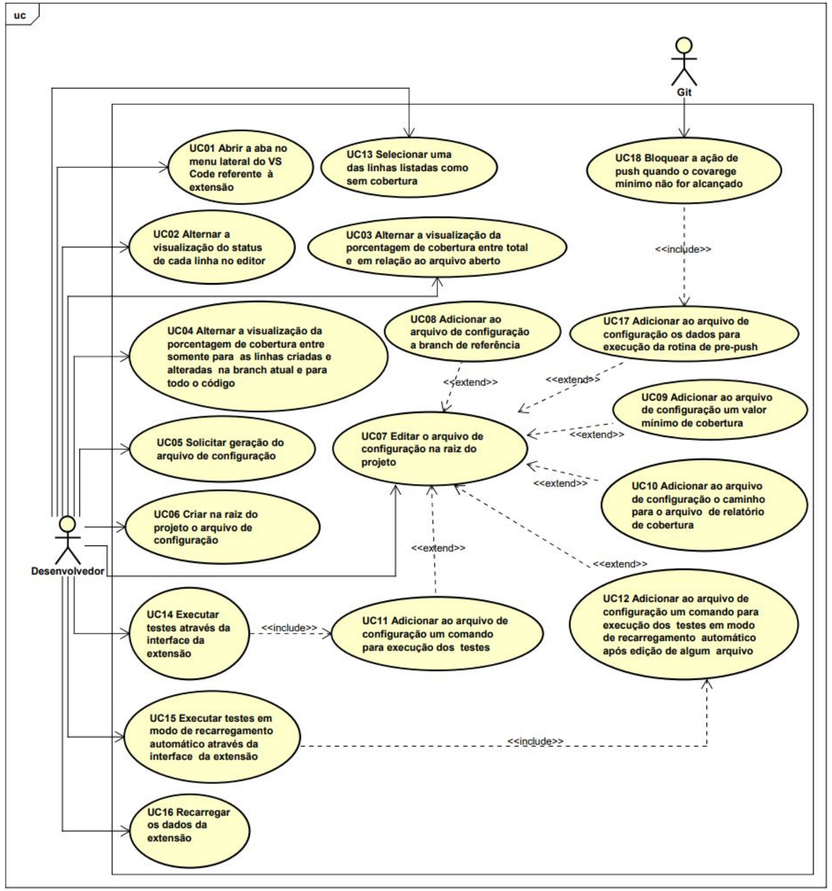
Desenvolvedor

Utilizador da extensão na IDE

Git

Íntegra com a extensão, para que as funcionalidades sejam implementadas

Diagrama de Caso de Uso


uc
Git
Desenvolvedor
UC01 Abrir a aba no menu lateral do VS Code referente à extensão
UC13 Selecionar uma das linhas listadas como sem cobertura
UC18 Bloquear a ação de push quando o covarege mínimo não for alcançado
UC02 Alternar a visualização do status de cada linha no editor
UC03 Alternar a visualização da porcentagem de cobertura entre total e em relação ao arquivo aberto
<<include>>
UC04 Alternar a visualização da porcentagem de cobertura entre somente para as linhas criadas e alteradas na branch atual e para todo o código
UC08 Adicionar ao arquivo de configuração a branch de referência
UC17 Adicionar ao arquivo de configuração os dados para execução da rotina de pre-push
<<extend>>
<<extend>>
UC09 Adicionar ao arquivo de configuração um valor mínimo de cobertura
UC05 Solicitar geração do arquivo de configuração
UC07 Editar o arquivo de configuração na raiz do projeto
<<extend>>
<<extend>>
UC10 Adicionar ao arquivo de configuração o caminho para o arquivo de relatório de cobertura
UC06 Criar na raiz do projeto o arquivo de configuração
<<extend>>
<<extend>>
UC12 Adicionar ao arquivo de configuração um comando para execução dos testes em modo de recarregamento automático após edição de algum arquivo
UC14 Executar testes através da interface da extensão
<<include>>
UC11 Adicionar ao arquivo de configuração um comando para execução dos testes
UC15 Executar testes em modo de recarregamento automático através da interface da extensão
<<include>>
UC16 Recarregar os dados da extensão

Principais Casos de Uso

UC01	Abrir a aba no menu lateral do VSCode
UC02	Alternar a visualização do *status* de cada linha no editor
UC04	Visualização da cobertura para as linhas criadas na branch atual
UC06	Criar na raiz do projeto o arquivo de configuração
UC08	Adicionar ao arquivo de configuração a *branch* de referência
UC09	Adicionar ao arquivo de configuração um valor mínimo de cobertura

Diagrama de Componentes

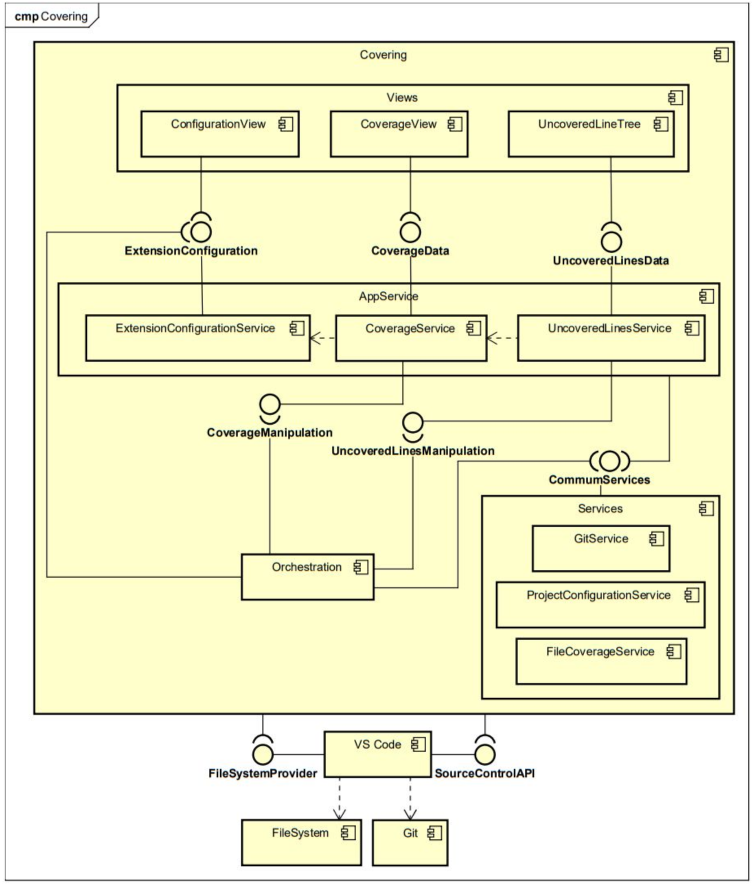


Diagrama de Componentes

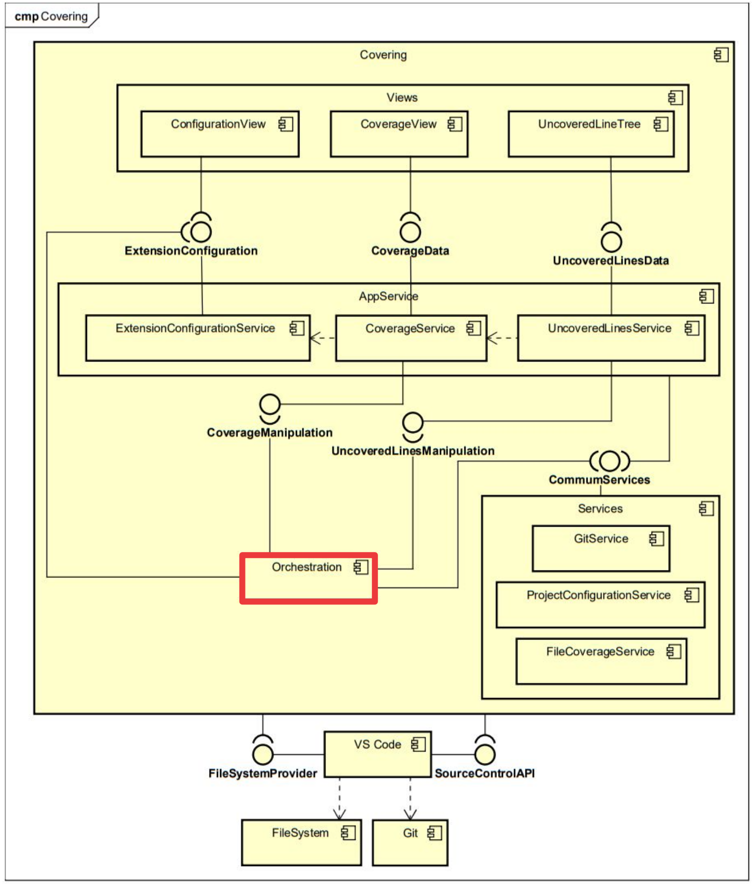


Diagrama de Componentes

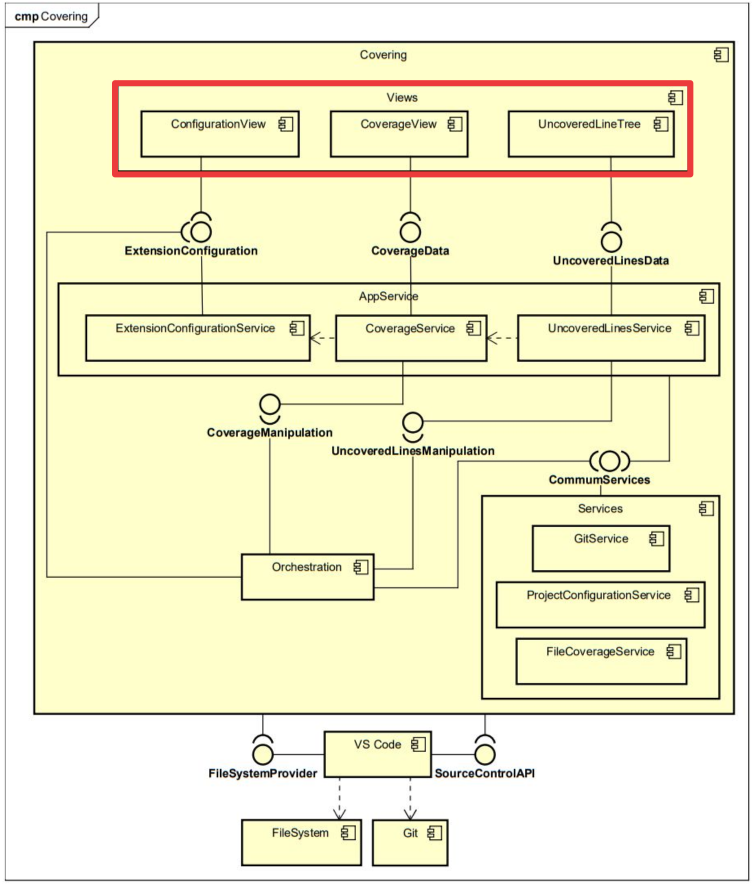


Diagrama de Componentes

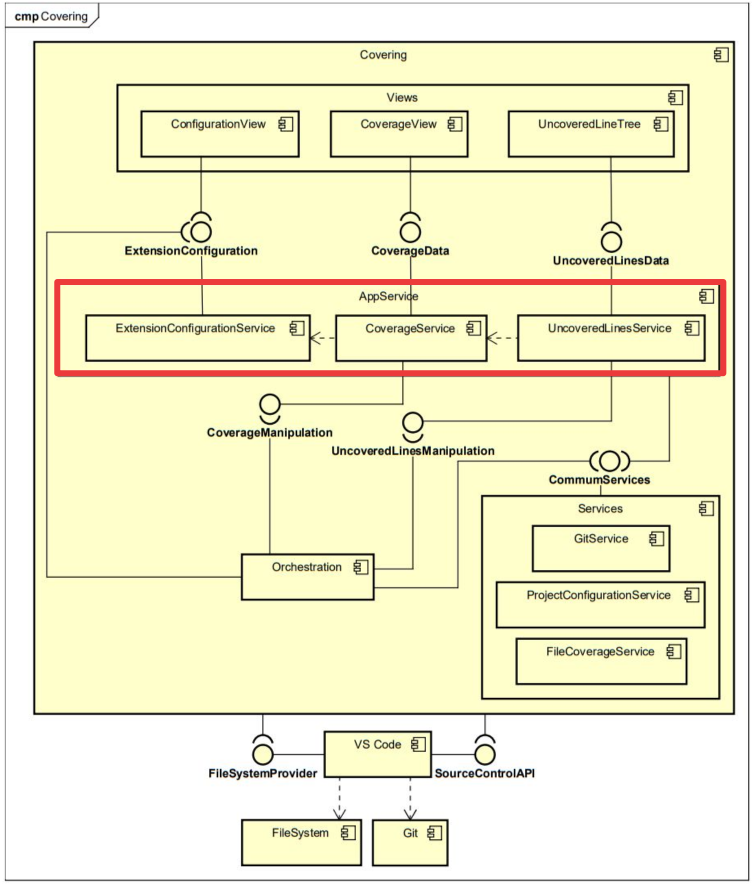


Diagrama de Componentes

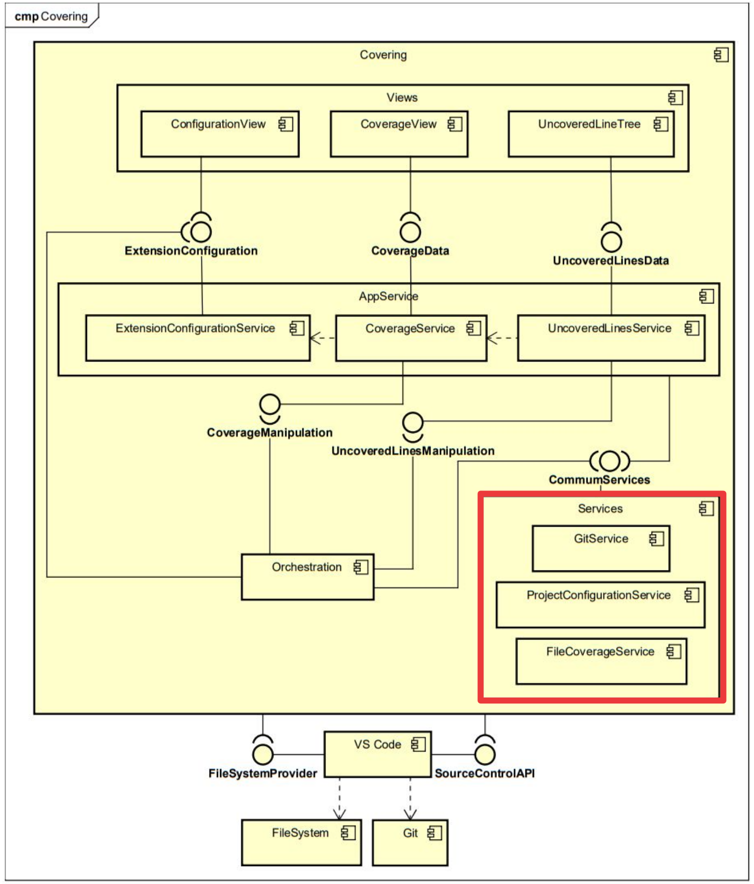
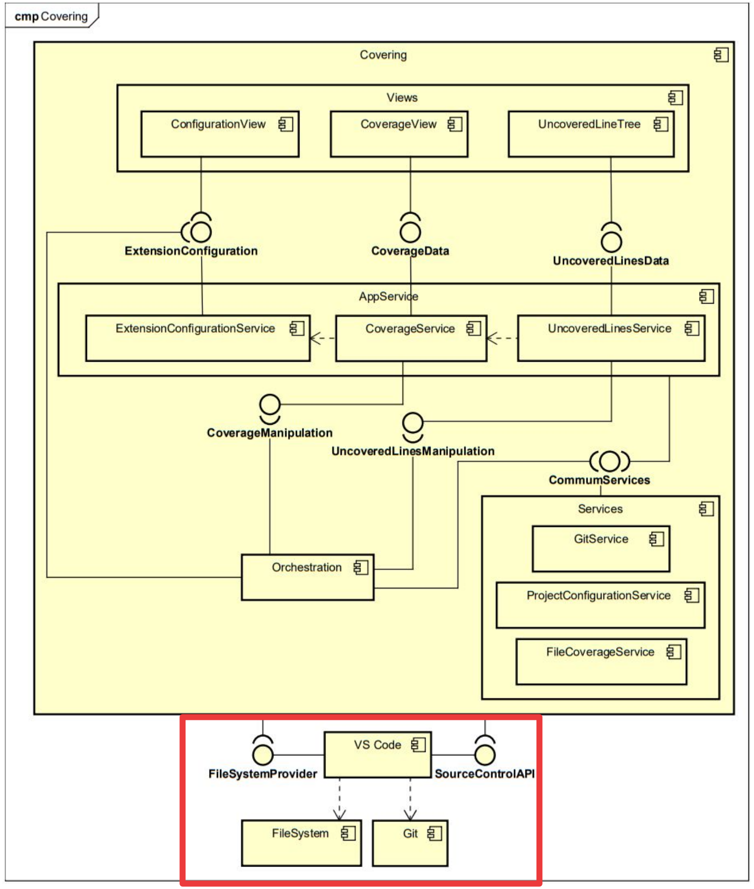


Diagrama de Componentes



Principais Componentes

Orchestration — Responsável por gerenciar as *views* e os serviços.

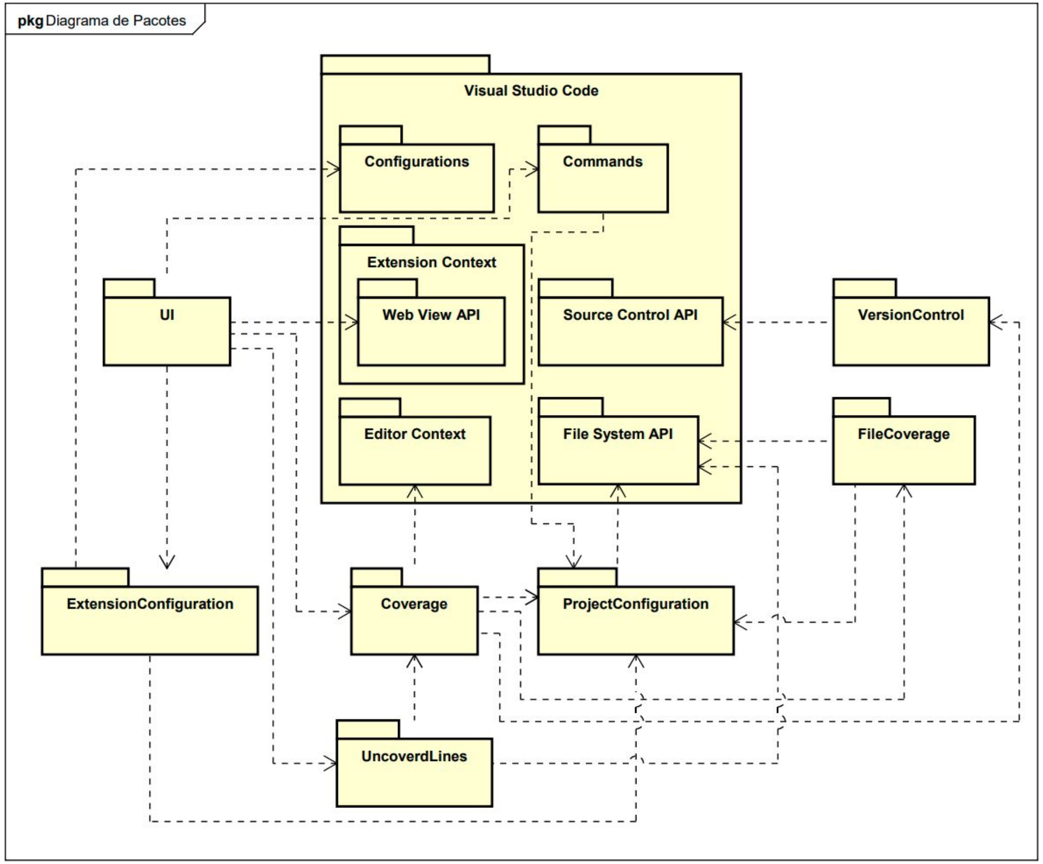
Views — Responsável pelos componentes visuais relacionados à exibição dos dados na IDE.

AppServices — Responsáveis pelos serviços que geram os dados da aplicação.

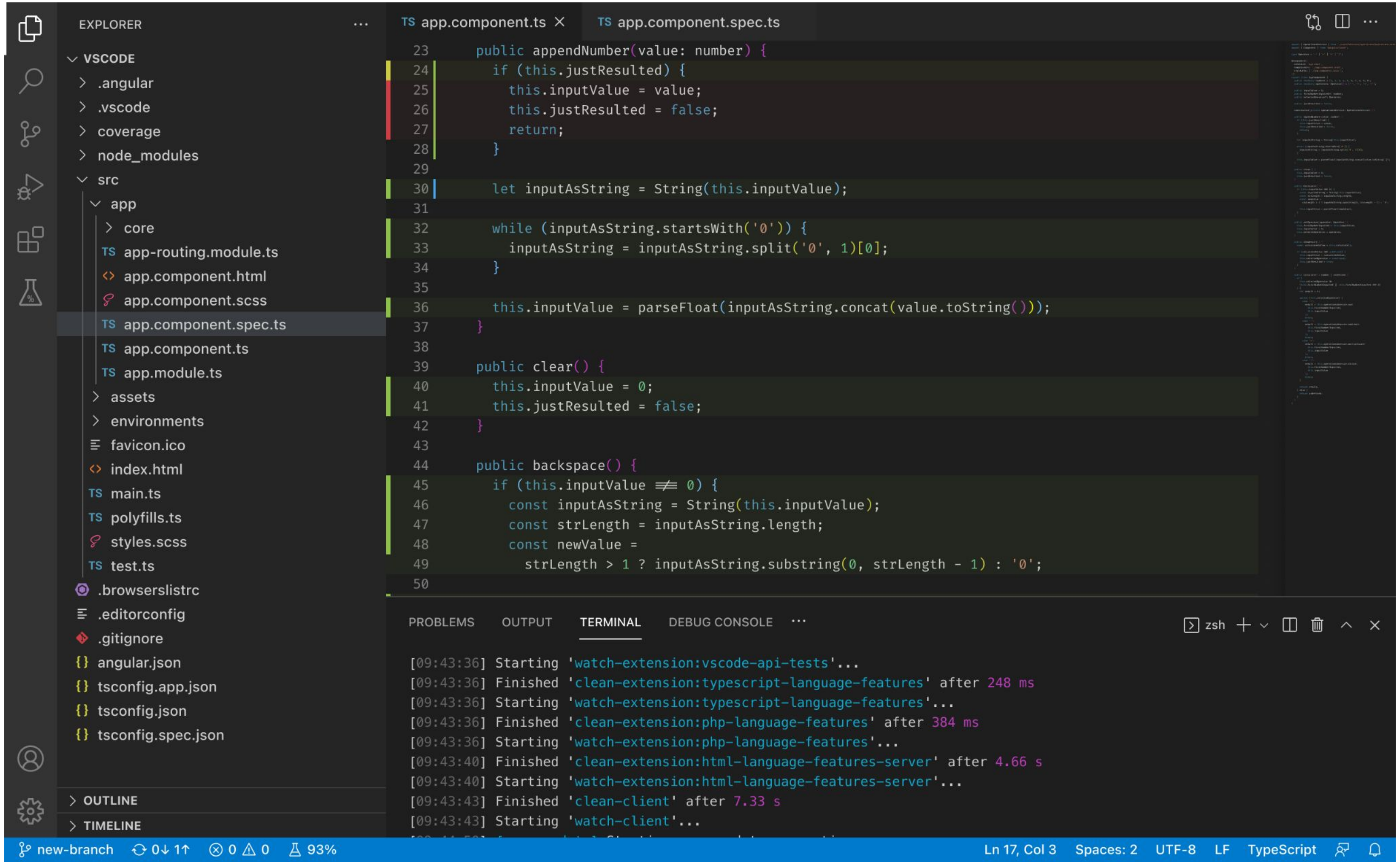
Services — Responsáveis por utilizar interfaces externas para gerar e emitir os dados base.

Externos — Aplicações que consumidas por meio de interface com o VSCode.

Diagrama de Pacotes



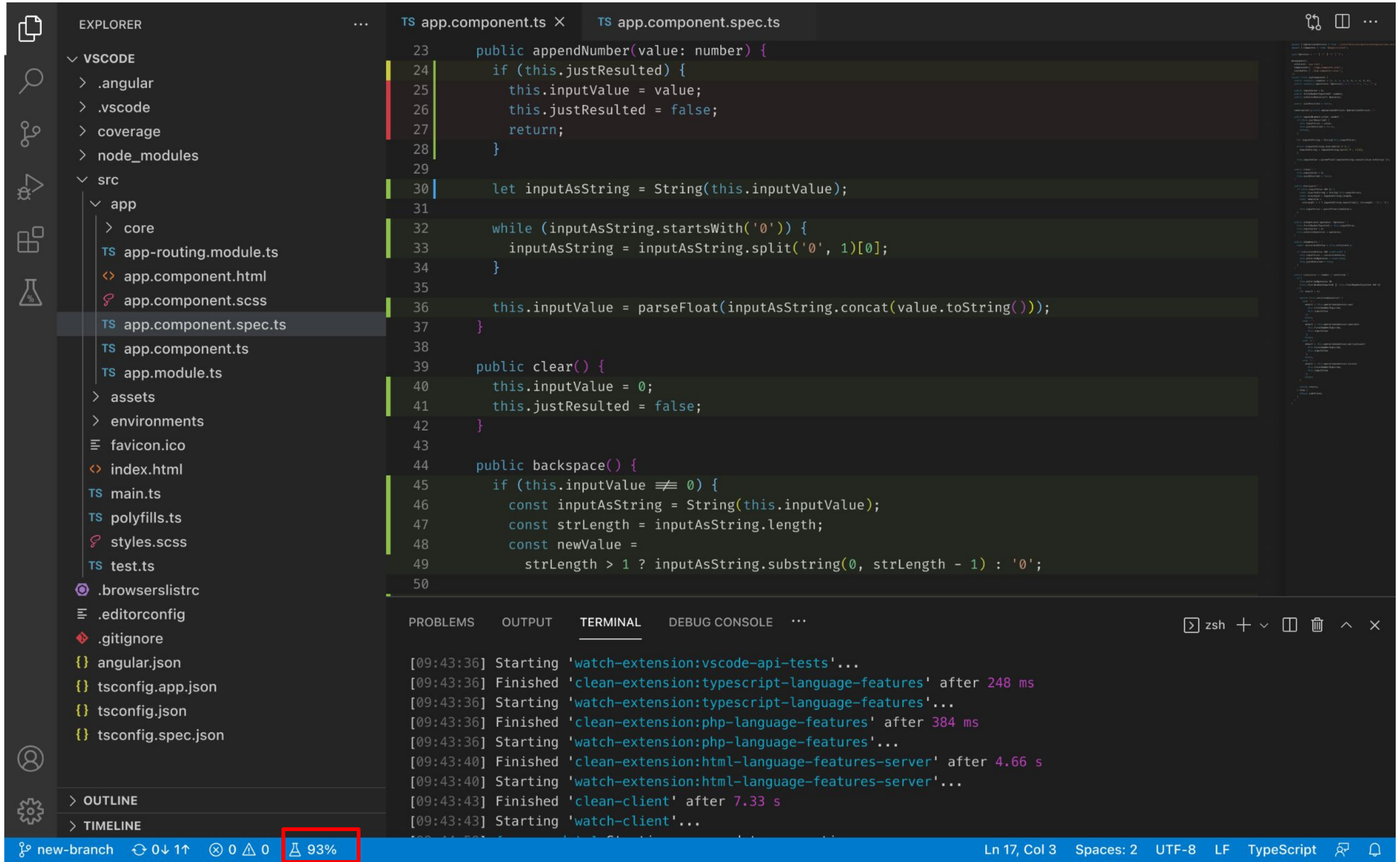
Projeto de Interfaces com o Usuário



```
  public appendNumber(value: number) {
    if (this.justResulted) {
      this.inputValue = value;
      this.justResulted = false;
      return;
    }

    let inputAsString = String(this.inputValue);

    while (inputAsString.startsWith('0')) {
      inputAsString = inputAsString.split('0', 1)[0];
    }

    this.inputValue = parseFloat(inputAsString.concat(value.toString()));
  }

  public clear() {
    this.inputValue = 0;
    this.justResulted = false;
  }

  public backspace() {
    if (this.inputValue !== 0) {
      const inputAsString = String(this.inputValue);
      const strLength = inputAsString.length;
      const newValue =
        strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';

```

```
[09:43:36] Starting 'watch-extension:vscode-api-tests'...
[09:43:36] Finished 'clean-extension:typescript-language-features' after 248 ms
[09:43:36] Starting 'watch-extension:typescript-language-features'...
[09:43:36] Finished 'clean-extension:php-language-features' after 384 ms
[09:43:36] Starting 'watch-extension:php-language-features'...
[09:43:40] Finished 'clean-extension:html-language-features-server' after 4.66 s
[09:43:40] Starting 'watch-extension:html-language-features-server'...
[09:43:43] Finished 'clean-client' after 7.33 s
[09:43:43] Starting 'watch-client'...
```

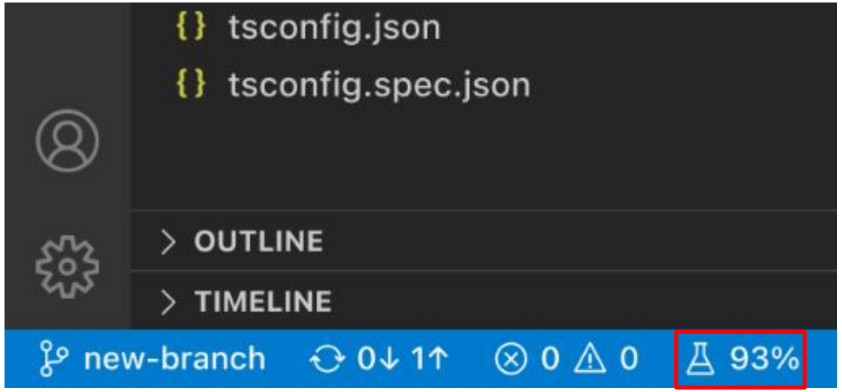
Projeto de Interfaces com o Usuário



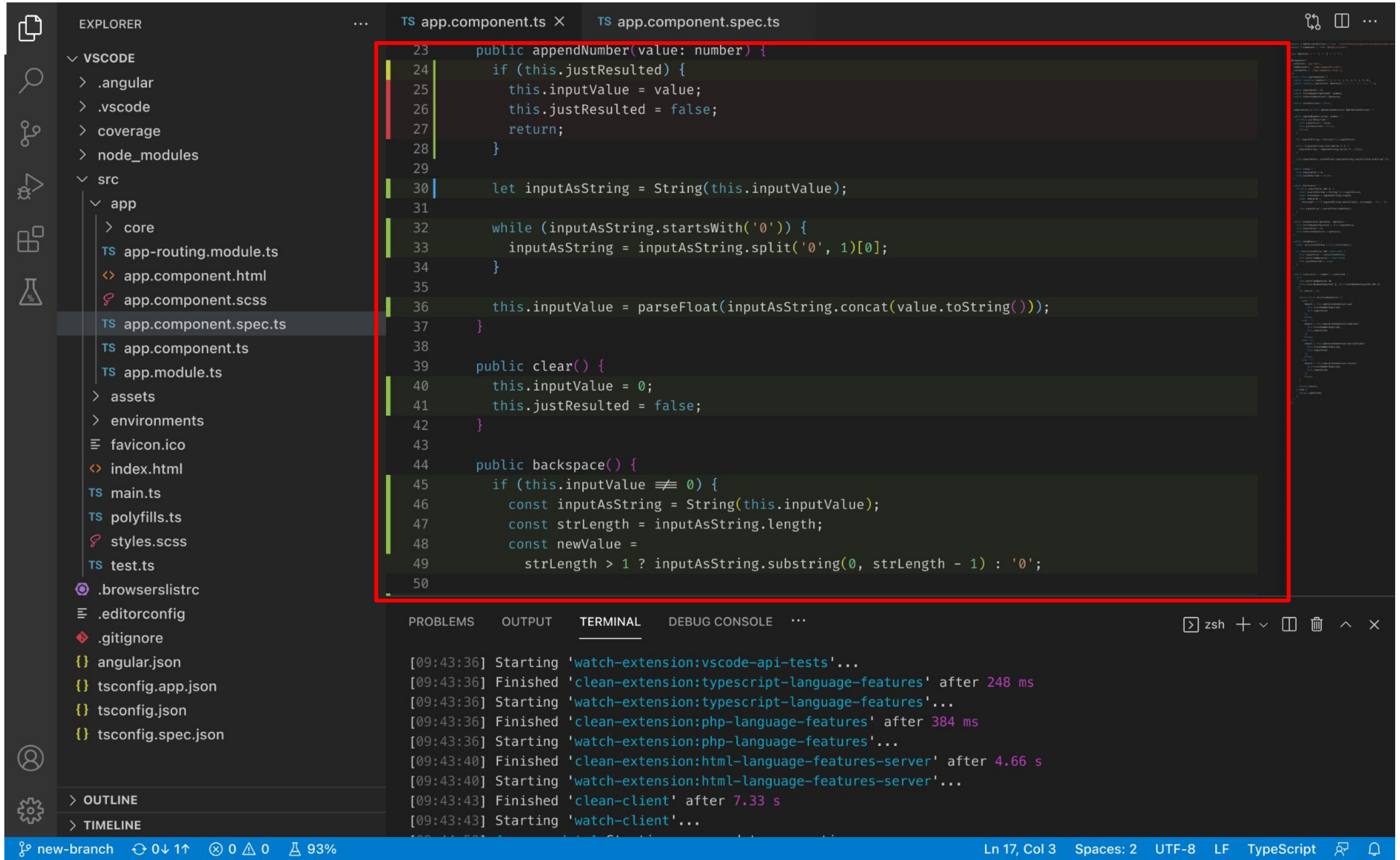
```typescript
  public appendNumber(value: number) {
    if (this.justResulted) {
      this.inputValue = value;
      this.justResulted = false;
      return;
    }

    let inputAsString = String(this.inputValue);

    while (inputAsString.startsWith('0')) {
      inputAsString = inputAsString.split('0', 1)[0];
    }

    this.inputValue = parseFloat(inputAsString.concat(value.toString()));
  }

  public clear() {
    this.inputValue = 0;
    this.justResulted = false;
  }

  public backspace() {
    if (this.inputValue !== 0) {
      const inputAsString = String(this.inputValue);
      const strLength = inputAsString.length;
      const newValue =
        strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';

```

```
[09:43:36] Starting 'watch-extension:vscode-api-tests'...
[09:43:36] Finished 'clean-extension:typescript-language-features' after 248 ms
[09:43:36] Starting 'watch-extension:typescript-language-features'...
[09:43:36] Finished 'clean-extension:php-language-features' after 384 ms
[09:43:36] Starting 'watch-extension:php-language-features'...
[09:43:40] Finished 'clean-extension:html-language-features-server' after 4.66 s
[09:43:40] Starting 'watch-extension:html-language-features-server'...
[09:43:43] Finished 'clean-client' after 7.33 s
[09:43:43] Starting 'watch-client'...
```

Projeto de Interfaces com o Usuário



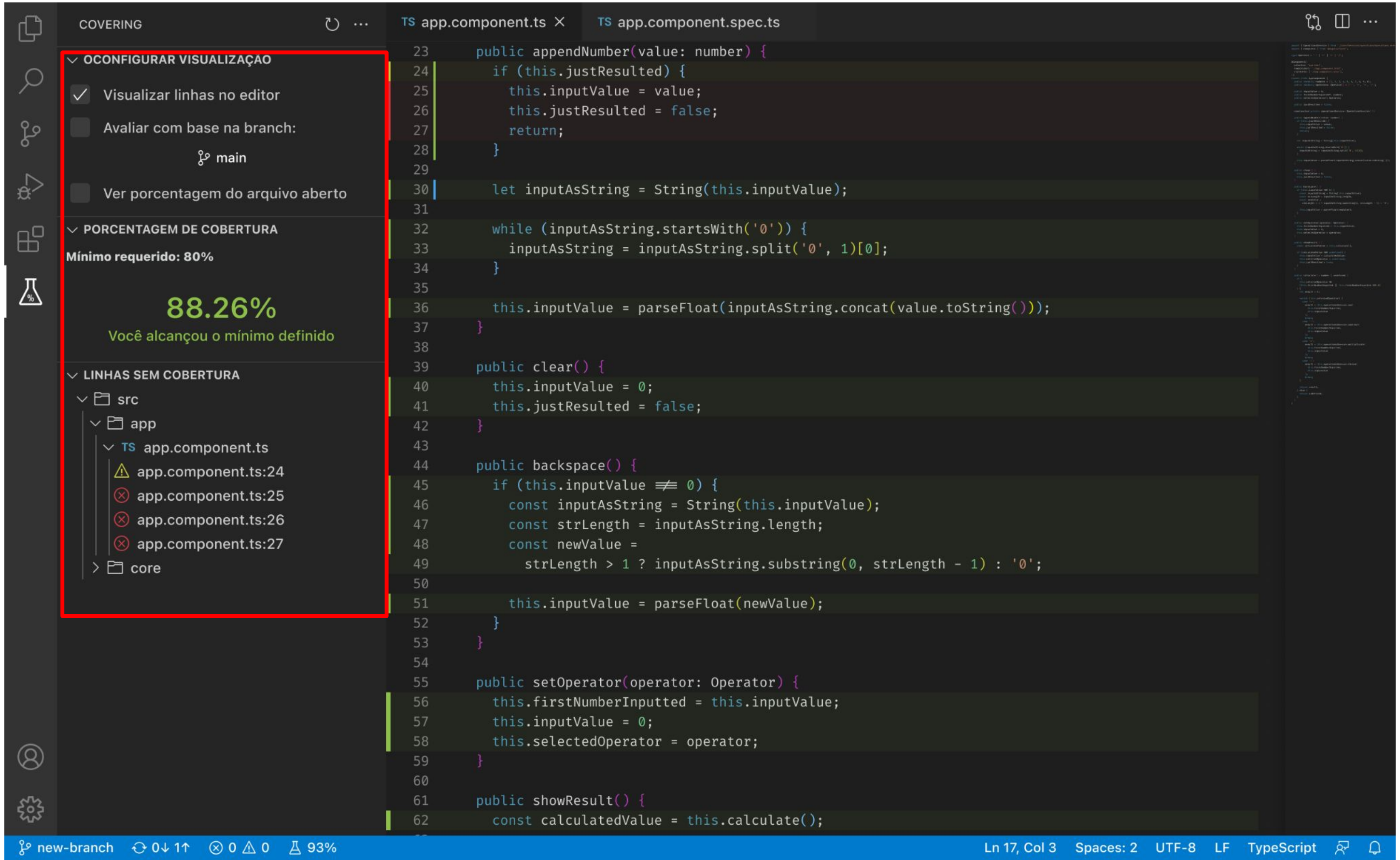
Projeto de Interfaces com o Usuário



```
  public appendNumber(value: number) {
    if (this.justResulted) {
      this.inputValue = value;
      this.justResulted = false;
      return;
    }

    let inputAsString = String(this.inputValue);

    while (inputAsString.startsWith('0')) {
      inputAsString = inputAsString.split('0', 1)[0];
    }

    this.inputValue = parseFloat(inputAsString.concat(value.toString()));
  }

  public clear() {
    this.inputValue = 0;
    this.justResulted = false;
  }

  public backspace() {
    if (this.inputValue !== 0) {
      const inputAsString = String(this.inputValue);
      const strLength = inputAsString.length;
      const newValue =
        strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';

```

```
[09:43:36] Starting 'watch-extension:vscode-api-tests'...
[09:43:36] Finished 'clean-extension:typescript-language-features' after 248 ms
[09:43:36] Starting 'watch-extension:typescript-language-features'...
[09:43:36] Finished 'clean-extension:php-language-features' after 384 ms
[09:43:36] Starting 'watch-extension:php-language-features'...
[09:43:40] Finished 'clean-extension:html-language-features-server' after 4.66 s
[09:43:40] Starting 'watch-extension:html-language-features-server'...
[09:43:43] Finished 'clean-client' after 7.33 s
[09:43:43] Starting 'watch-client'...
```

Projeto de Interfaces com o Usuário



COVERING

OCONFIGURAR VISUALIZAÇAO

- [✓] Visualizar linhas no editor
- [ ] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

88.26%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

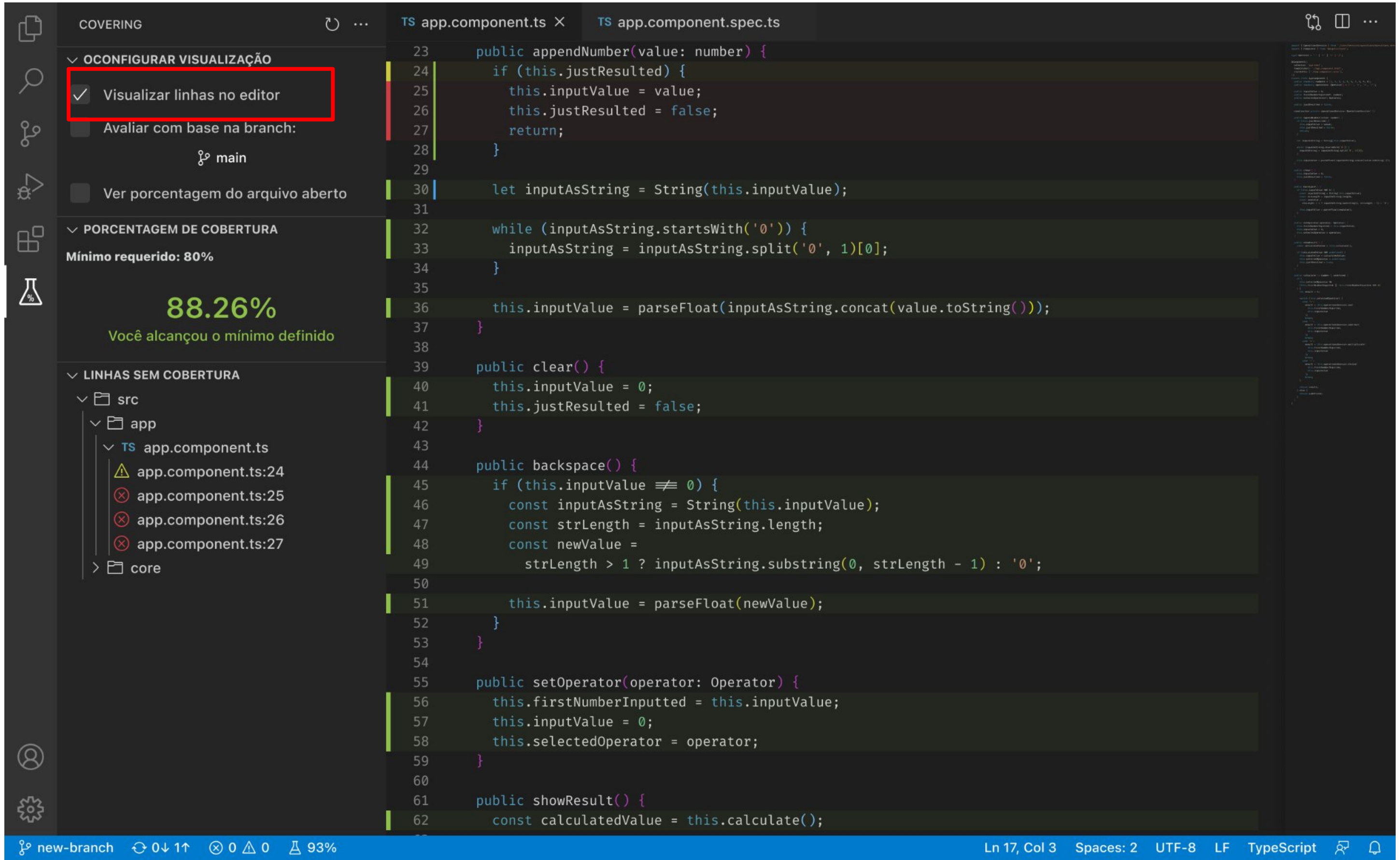
- src
  - app
    - app.component.ts
      - ⚠ app.component.ts:24
      - ⊗ app.component.ts:25
      - ⊗ app.component.ts:26
      - ⊗ app.component.ts:27
  - core

app.component.ts | app.component.spec.ts

```
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

new-branch 0↓ 1↑ ⊗ 0 ⚠ 0 93% Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



COVERING

OCONFIGURAR VISUALIZAÇÃO

- [✓] Visualizar linhas no editor
- [ ] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

88.26%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

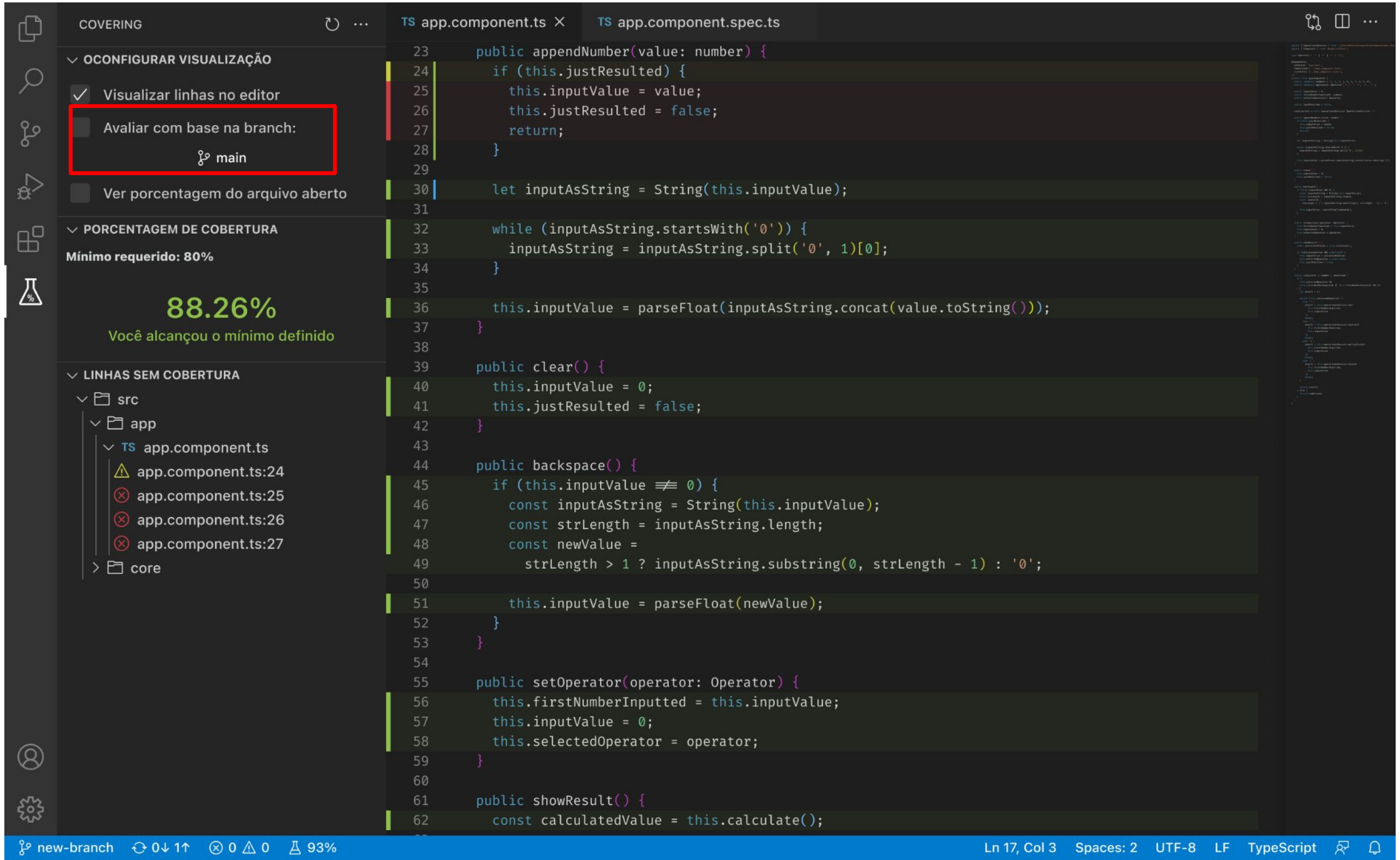
- src
  - app
    - app.component.ts
      - ⚠ app.component.ts:24
      - ⊗ app.component.ts:25
      - ⊗ app.component.ts:26
      - ⊗ app.component.ts:27
  - core

app.component.ts | app.component.spec.ts

```
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

new-branch ⟳ 0↓ 1↑ ⊗ 0 ⚠ 0 93% Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



COVERING

OCONFIGURAR VISUALIZAÇÃO

- [✓] Visualizar linhas no editor
- [ ] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

88.26%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

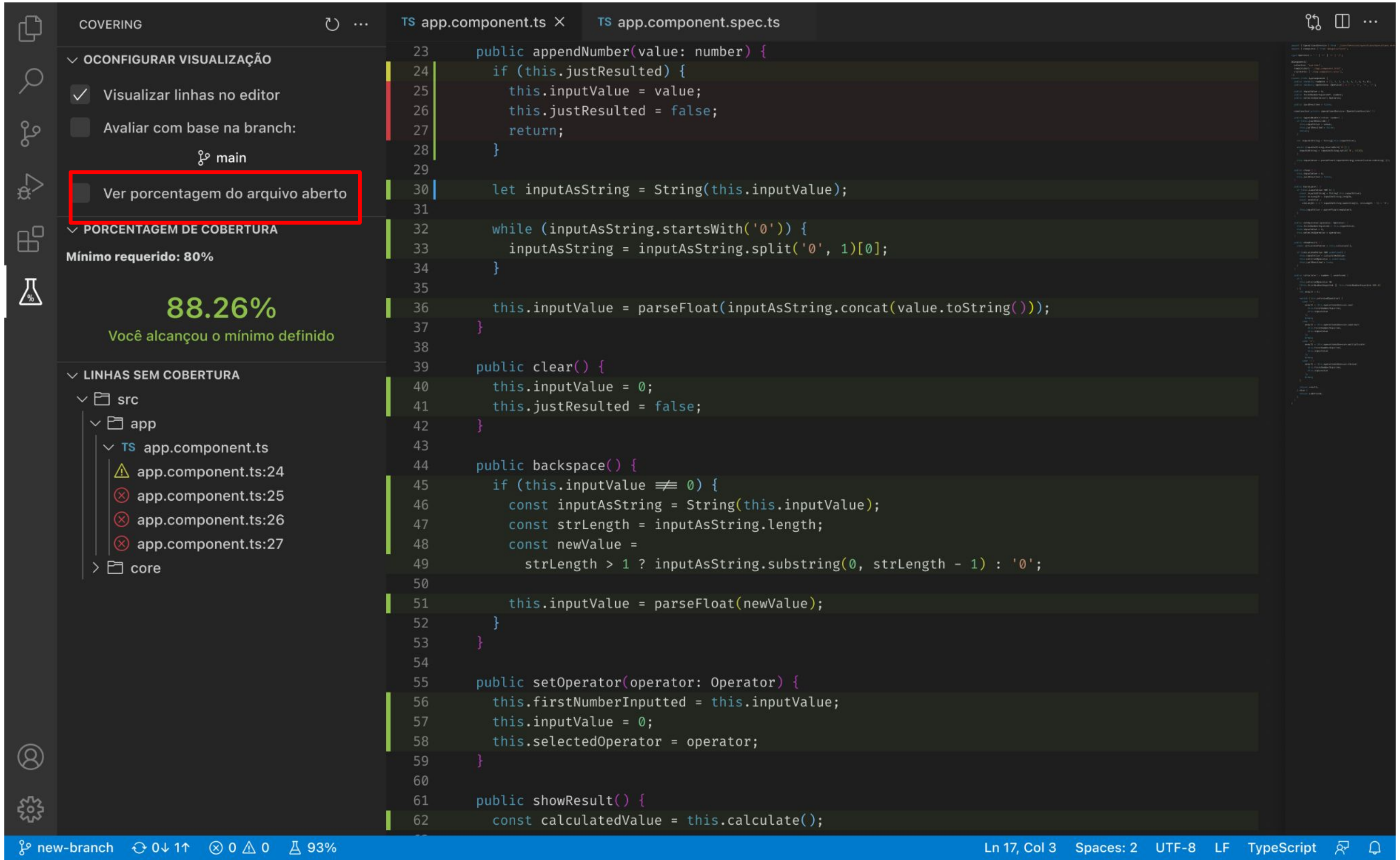
- src
  - app
    - TS app.component.ts
      - ⚠ app.component.ts:24
      - ⓧ app.component.ts:25
      - ⓧ app.component.ts:26
      - ⓧ app.component.ts:27
  - core

TS app.component.ts × | TS app.component.spec.ts

```
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

new-branch ⟳ 0↓ 1↑ ⓧ 0 ⚠ 0 93% | Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



COVERING

OCONFIGURAR VISUALIZAÇÃO

- [x] Visualizar linhas no editor
- [ ] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

88.26%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

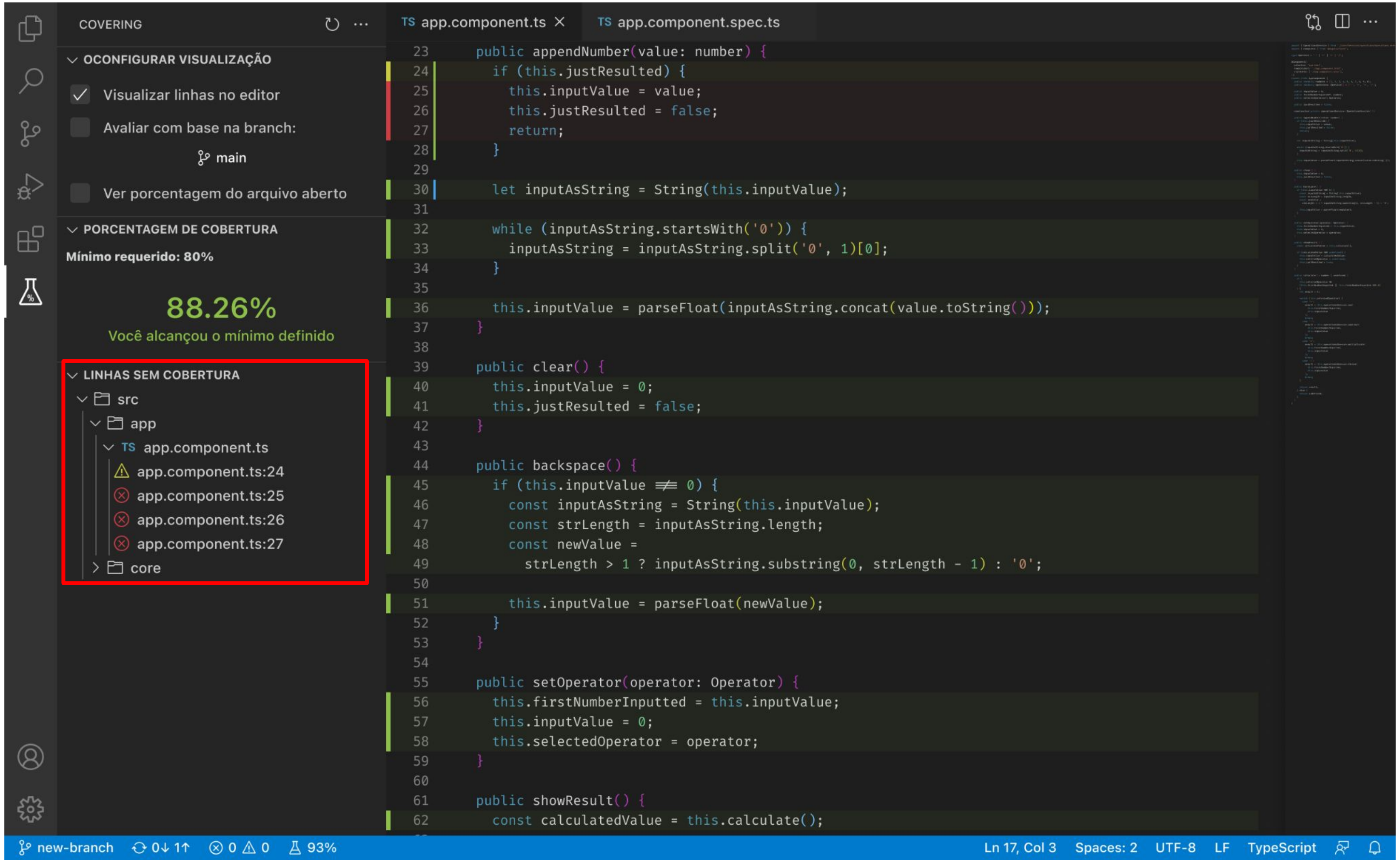
- src
  - app
    - app.component.ts
      - app.component.ts:24
      - app.component.ts:25
      - app.component.ts:26
      - app.component.ts:27
  - core

app.component.ts | app.component.spec.ts

```typescript
  public appendNumber(value: number) {
    if (this.justResulted) {
      this.inputValue = value;
      this.justResulted = false;
      return;
    }

    let inputAsString = String(this.inputValue);

    while (inputAsString.startsWith('0')) {
      inputAsString = inputAsString.split('0', 1)[0];
    }

    this.inputValue = parseFloat(inputAsString.concat(value.toString()));
  }

  public clear() {
    this.inputValue = 0;
    this.justResulted = false;
  }

  public backspace() {
    if (this.inputValue !== 0) {
      const inputAsString = String(this.inputValue);
      const strLength = inputAsString.length;
      const newValue =
        strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';

      this.inputValue = parseFloat(newValue);
    }
  }

  public setOperator(operator: Operator) {
    this.firstNumberInputted = this.inputValue;
    this.inputValue = 0;
    this.selectedOperator = operator;
  }

  public showResult() {
    const calculatedValue = this.calculate();
```

new-branch 0↓ 1↑ ⊗ 0 ⚠ 0 93% Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



COVERING

OCONFIGURAR VISUALIZAÇÃO

- [✓] Visualizar linhas no editor
- [ ] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

88.26%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

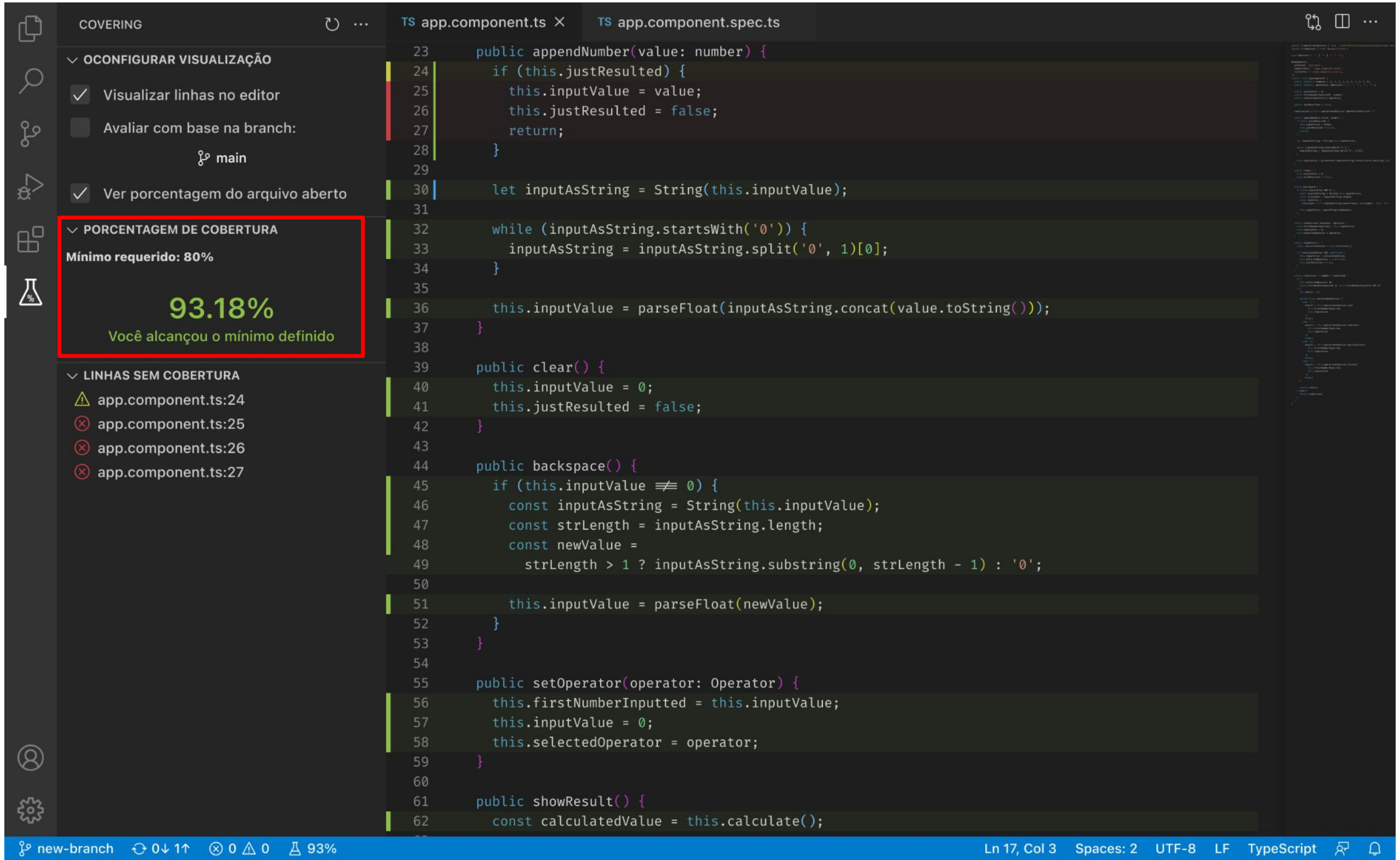
- src
  - app
    - app.component.ts
      - app.component.ts:24
      - app.component.ts:25
      - app.component.ts:26
      - app.component.ts:27
  - core

app.component.ts | app.component.spec.ts

```typescript
  public appendNumber(value: number) {
    if (this.justResulted) {
      this.inputValue = value;
      this.justResulted = false;
      return;
    }

    let inputAsString = String(this.inputValue);

    while (inputAsString.startsWith('0')) {
      inputAsString = inputAsString.split('0', 1)[0];
    }

    this.inputValue = parseFloat(inputAsString.concat(value.toString()));
  }

  public clear() {
    this.inputValue = 0;
    this.justResulted = false;
  }

  public backspace() {
    if (this.inputValue !== 0) {
      const inputAsString = String(this.inputValue);
      const strLength = inputAsString.length;
      const newValue =
        strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';

      this.inputValue = parseFloat(newValue);
    }
  }

  public setOperator(operator: Operator) {
    this.firstNumberInputted = this.inputValue;
    this.inputValue = 0;
    this.selectedOperator = operator;
  }

  public showResult() {
    const calculatedValue = this.calculate();
```

new-branch 0↓ 1↑ 0 0 93% Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



COVERING

OCONFIGURAR VISUALIZAÇÃO

- [✓] Visualizar linhas no editor
- [ ] Avaliar com base na branch:

  main

- [✓] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

93.18%

Você alcançou o mínimo definido

LINHAS SEM COBERTURA

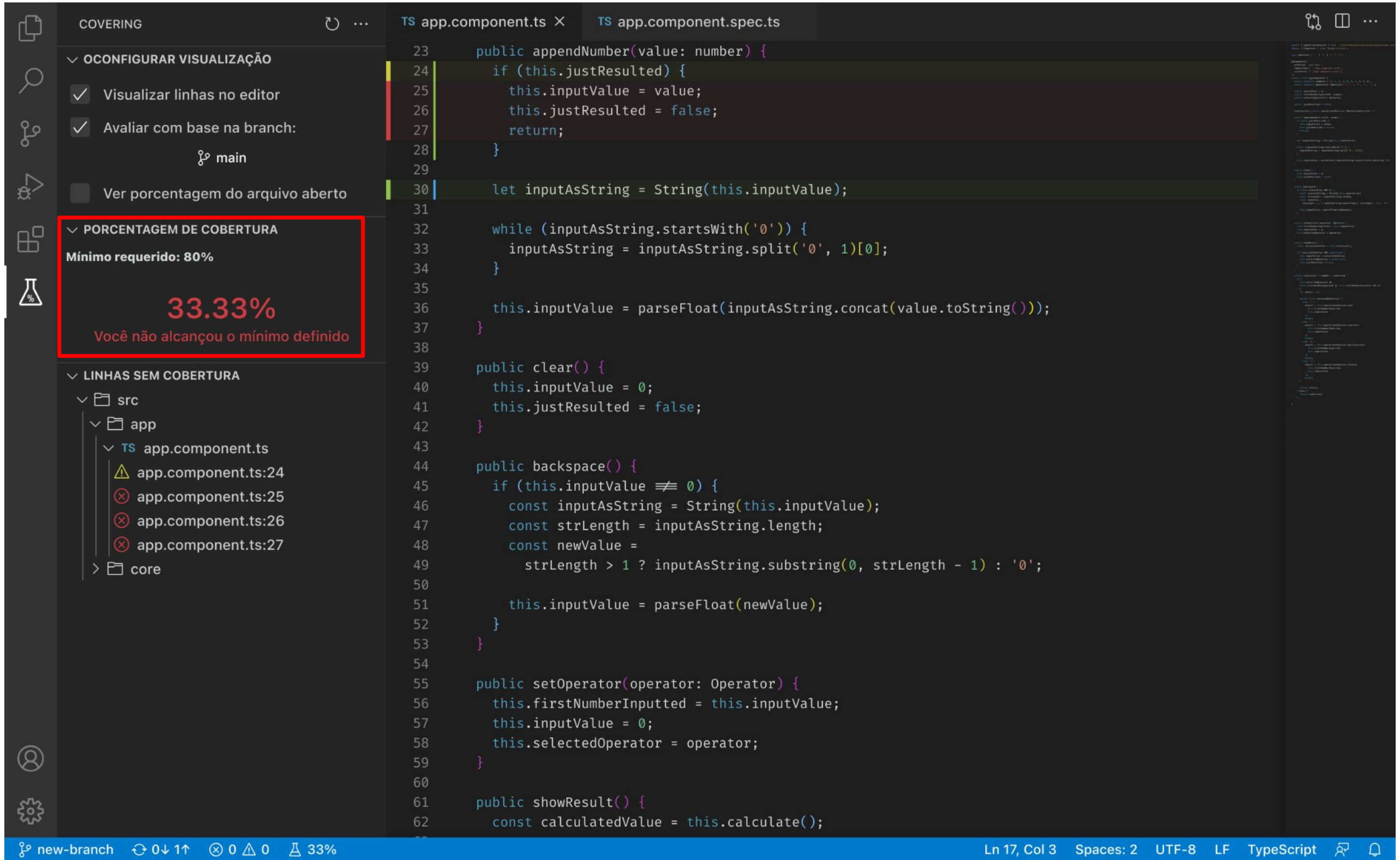
- ⚠ app.component.ts:24
- ⊗ app.component.ts:25
- ⊗ app.component.ts:26
- ⊗ app.component.ts:27

TS app.component.ts × | TS app.component.spec.ts

```
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

new-branch ⟳ 0↓ 1↑ ⊗ 0 ⚠ 0 93% | Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



OCONFIGURAR VISUALIZAÇÃO

- [✓] Visualizar linhas no editor
- [✓] Avaliar com base na branch: main
- [ ] Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 80%

33.33%

Você não alcançou o mínimo definido

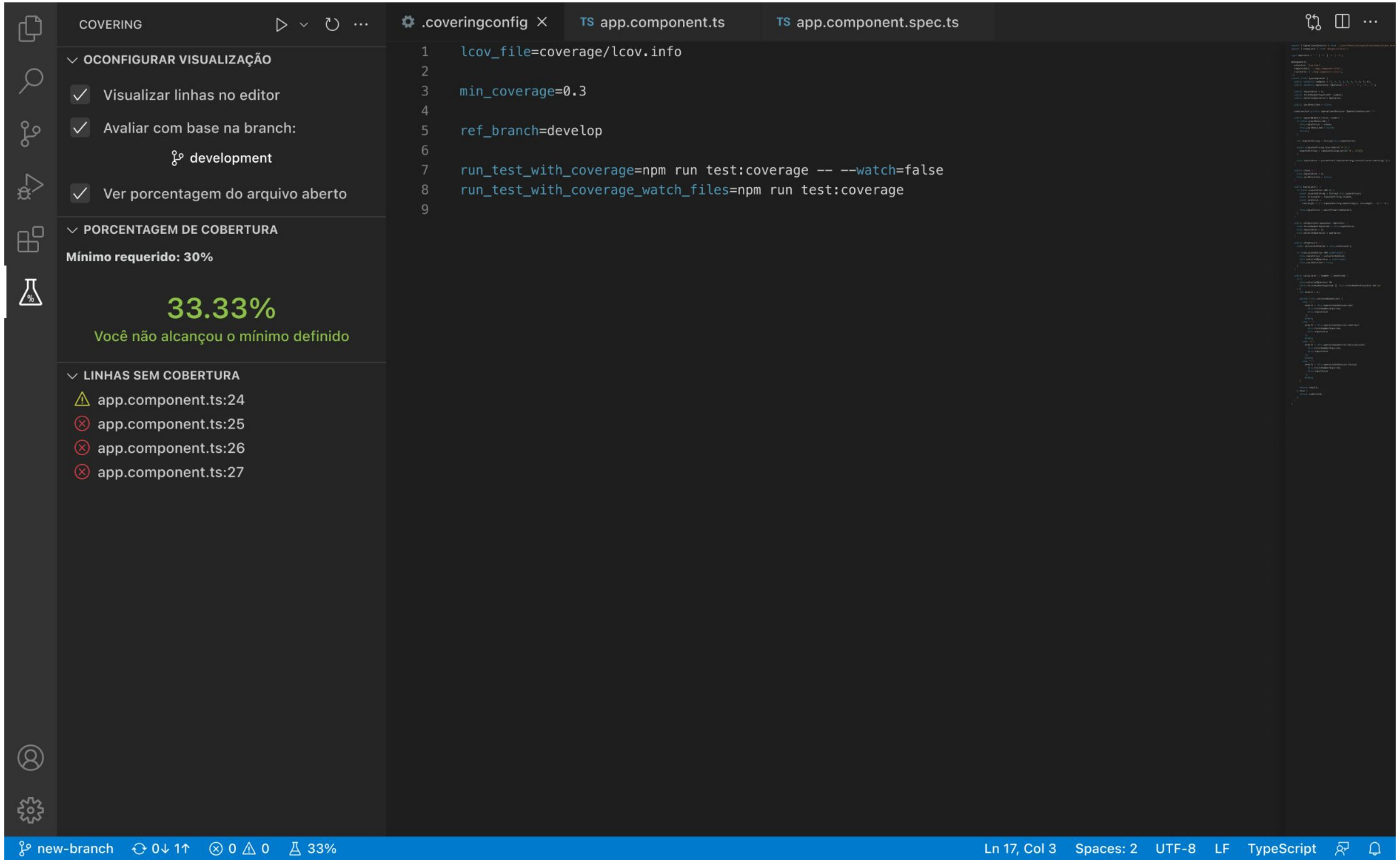
LINHAS SEM COBERTURA

- src
  - app
    - TS app.component.ts
      - ⚠ app.component.ts:24
      - ⊗ app.component.ts:25
      - ⊗ app.component.ts:26
      - ⊗ app.component.ts:27
  - core

```
23    public appendNumber(value: number) {
24      if (this.justResulted) {
25        this.inputValue = value;
26        this.justResulted = false;
27        return;
28      }
29
30      let inputAsString = String(this.inputValue);
31
32      while (inputAsString.startsWith('0')) {
33        inputAsString = inputAsString.split('0', 1)[0];
34      }
35
36      this.inputValue = parseFloat(inputAsString.concat(value.toString()));
37    }
38
39    public clear() {
40      this.inputValue = 0;
41      this.justResulted = false;
42    }
43
44    public backspace() {
45      if (this.inputValue !== 0) {
46        const inputAsString = String(this.inputValue);
47        const strLength = inputAsString.length;
48        const newValue =
49          strLength > 1 ? inputAsString.substring(0, strLength - 1) : '0';
50
51        this.inputValue = parseFloat(newValue);
52      }
53    }
54
55    public setOperator(operator: Operator) {
56      this.firstNumberInputted = this.inputValue;
57      this.inputValue = 0;
58      this.selectedOperator = operator;
59    }
60
61    public showResult() {
62      const calculatedValue = this.calculate();
```

new-branch 0↓ 1↑ ⊗ 0 ⚠ 0 33%   Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

Projeto de Interfaces com o Usuário



OCONFIGURAR VISUALIZAÇÃO

✓ Visualizar linhas no editor

✓ Avaliar com base na branch:

development

✓ Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 30%

33.33%

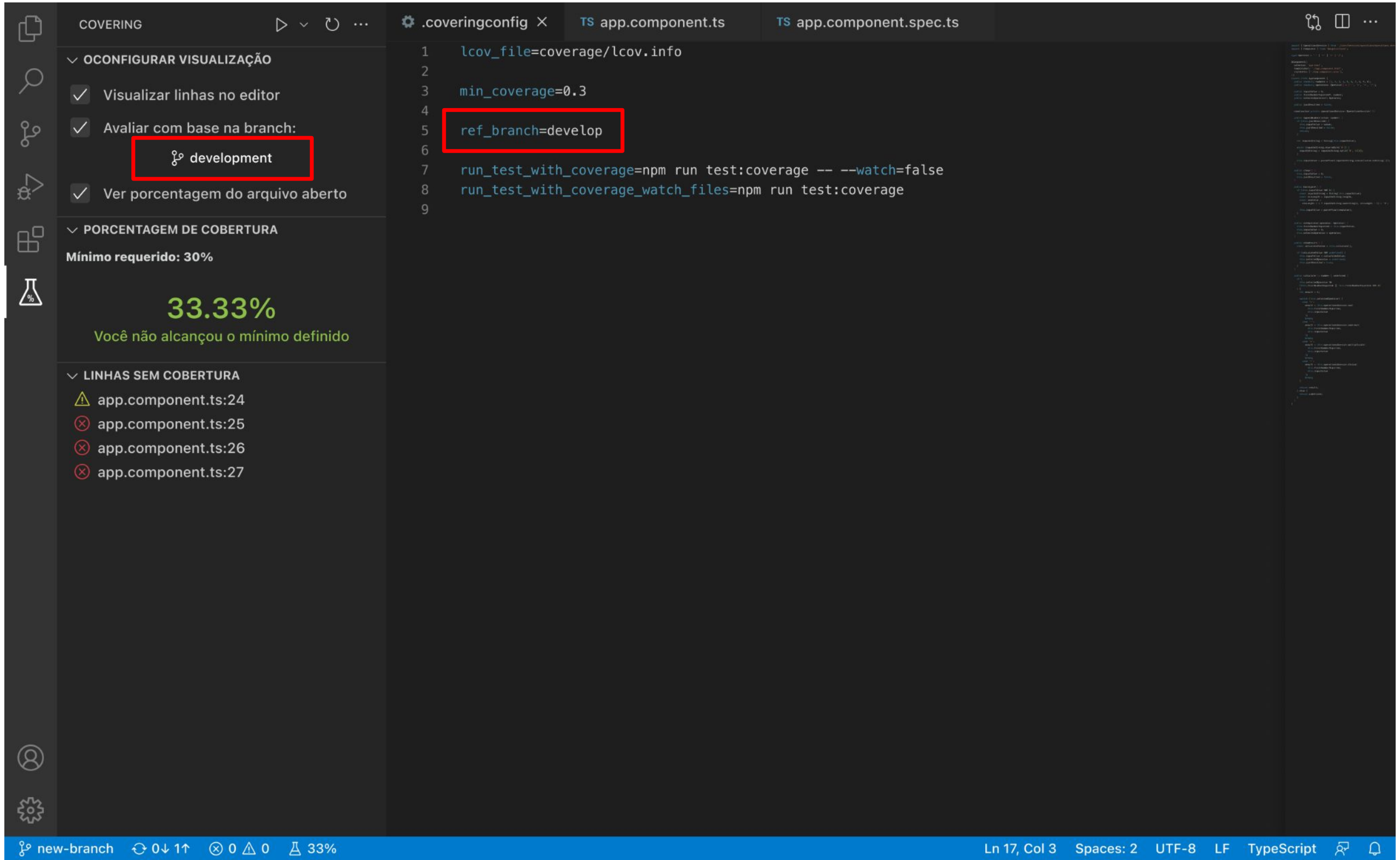
Você não alcançou o mínimo definido

LINHAS SEM COBERTURA

- app.component.ts:24
- app.component.ts:25
- app.component.ts:26
- app.component.ts:27

```
lcov_file=coverage/lcov.info

min_coverage=0.3

ref_branch=develop

run_test_with_coverage=npm run test:coverage -- --watch=false
run_test_with_coverage_watch_files=npm run test:coverage
```

Projeto de Interfaces com o Usuário



✓ Visualizar linhas no editor

✓ Avaliar com base na branch:

development

✓ Ver porcentagem do arquivo aberto

Mínimo requerido: 30%

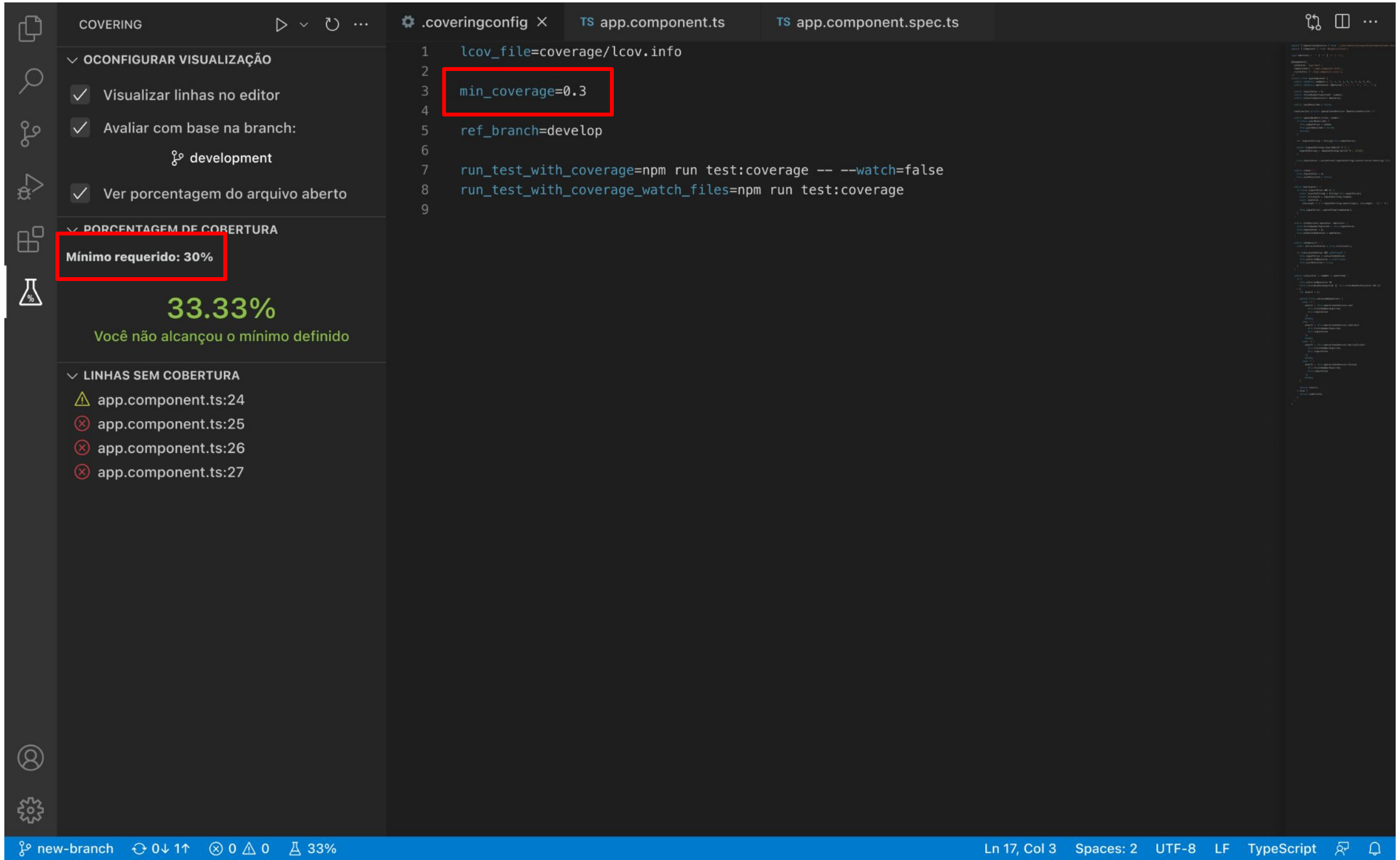
33.33%

Você não alcançou o mínimo definido

- app.component.ts:24
- app.component.ts:25
- app.component.ts:26
- app.component.ts:27

```
lcov_file=coverage/lcov.info

min_coverage=0.3

ref_branch=develop

run_test_with_coverage=npm run test:coverage -- --watch=false
run_test_with_coverage_watch_files=npm run test:coverage
```

Projeto de Interfaces com o Usuário



✓ Visualizar linhas no editor

✓ Avaliar com base na branch:

development

✓ Ver porcentagem do arquivo aberto

Mínimo requerido: 30%

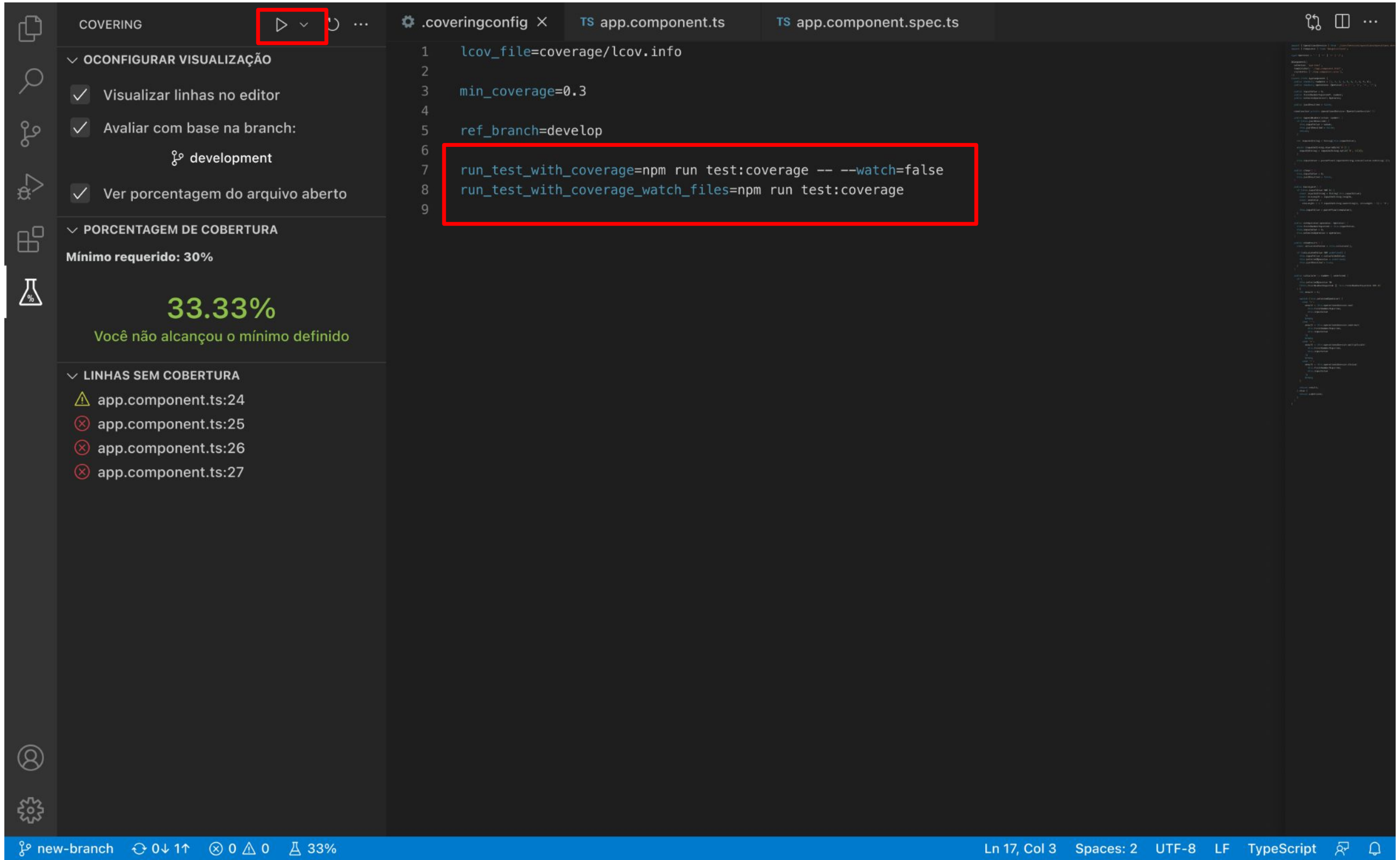
33.33%

Você não alcançou o mínimo definido

- ⚠ app.component.ts:24
- ⊗ app.component.ts:25
- ⊗ app.component.ts:26
- ⊗ app.component.ts:27

```
lcov_file=coverage/lcov.info

min_coverage=0.3

ref_branch=develop

run_test_with_coverage=npm run test:coverage -- --watch=false
run_test_with_coverage_watch_files=npm run test:coverage
```

Projeto de Interfaces com o Usuário



OCONFIGURAR VISUALIZAÇÃO

- ✓ Visualizar linhas no editor
- ✓ Avaliar com base na branch: development
- ✓ Ver porcentagem do arquivo aberto

PORCENTAGEM DE COBERTURA

Mínimo requerido: 30%

33.33%

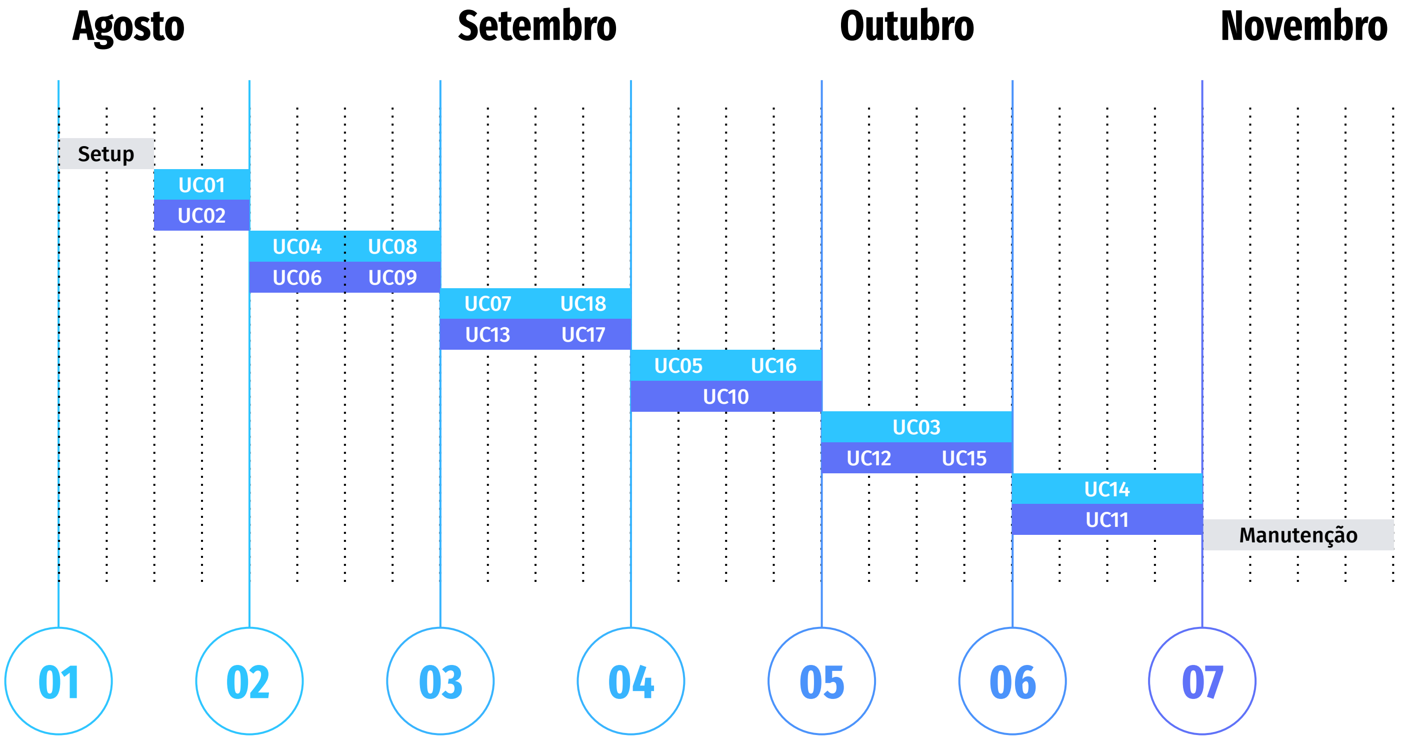
Você não alcançou o mínimo definido

LINHAS SEM COBERTURA

- app.component.ts:24
- app.component.ts:25
- app.component.ts:26
- app.component.ts:27

```
lcov_file=coverage/lcov.info

min_coverage=0.3

ref_branch=develop

run_test_with_coverage=npm run test:coverage -- --watch=false
run_test_with_coverage_watch_files=npm run test:coverage
```

Cronograma do TCC II



Legenda:

Todos integrantes

Arthur Amaral

Guilherme Oliveira

PUC Minas

Obrigado

Arthur Amaral - Guilherme Antônio

A/B Testing Infographics

	A	B
Visitors	8343	7849
Conversions	189	173
Conversion rate	4%	3%
Results	Jupiter is a gas giant and also the biggest planet	Venus is the second planet from the Sun

A/B Testing Infographics

A/B Testing overview

Despite being red, Mars is a cold place full of iron oxide dust

Step 1

Mercury is the smallest of them all

Step 2

Saturn is a gas giant and has several rings

Step 3

Jupiter is the biggest planet of them all

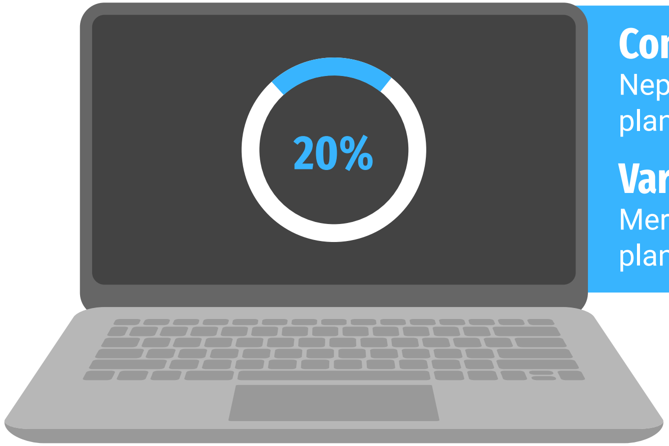
Step 4

Neptune is very far away from Earth

Step 5

Earth is the planet where we live on

A/B Testing Infographics



Control
Neptune is the farthest planet from the Sun

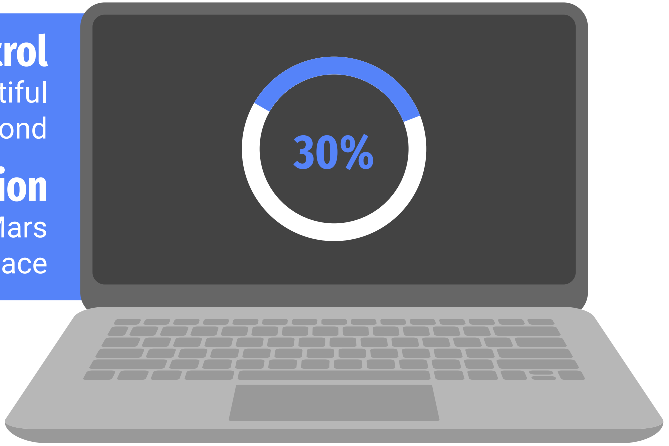
Variation
Mercury is the closest planet to the Sun

A

B

Control
Venus has a beautiful name and is the second

Variation
Despite being red, Mars is a very cold place



A/B Testing Infographics

01

02

03

04

Analyze data

Mercury is the closest planet to the Sun and the smallest one of them all

Form an hypothesis

It's a gas giant and the biggest planet in the entire Solar System

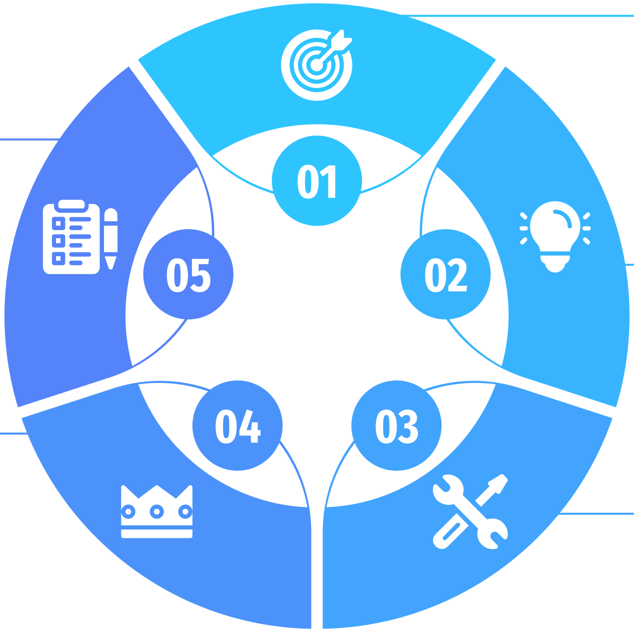
Experiment

Despite being red, Mars is a very cold planet full of iron oxide dust

Evaluate results

Venus is the second planet from the Sun and is terribly hot

A/B Testing Infographics



Define goal

It's a gas giant and the biggest planet

Generate idea

Venus is the second planet from the Sun

Implement changes

Despite being red, Mars is a cold place

Test until significant

Despite being red, Mars is a cold place

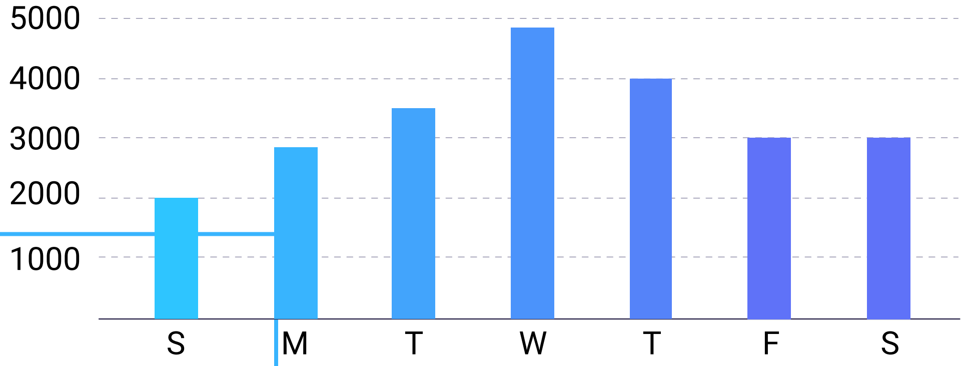
Crown the champion

Mercury is the closest planet to the Sun

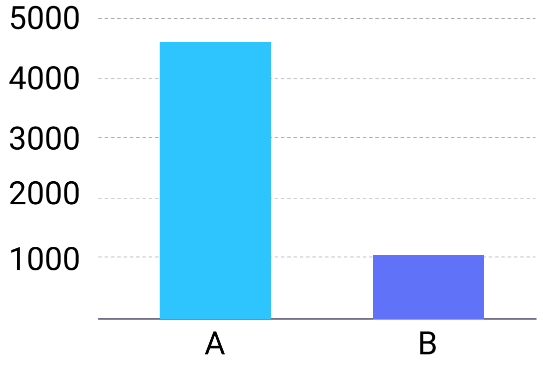
A/B Testing Infographics

Variation	Users	Conversions	CR	Uplift	Change
A Default	30,000	1,200	4.00%	—	—
B Variation	30,000	1,280	4.25%	5.00%	89.1%

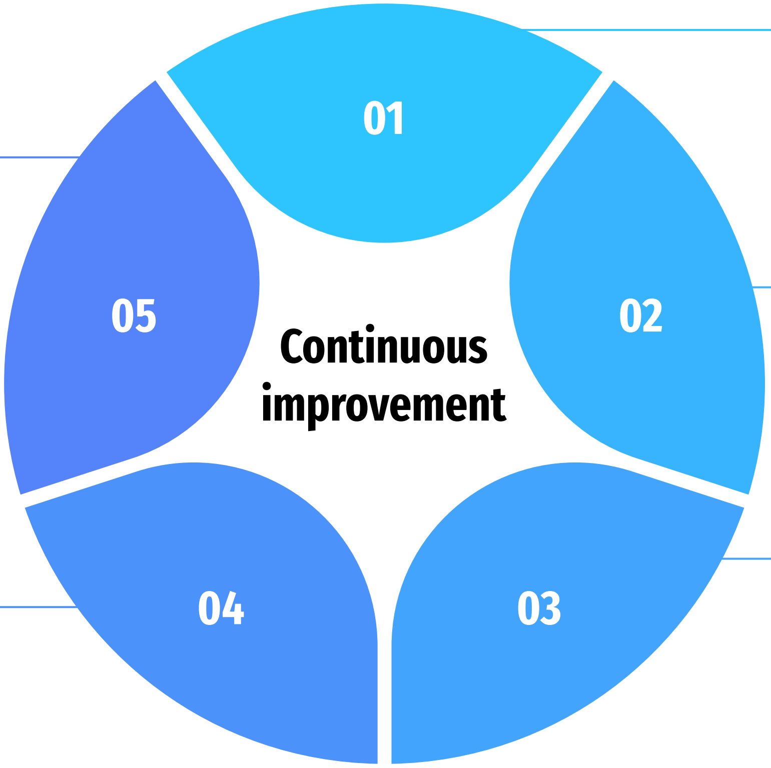
Sum of newsletter sign ups



Chance of B outperforming A



A/B Testing Infographics



01
Define goal
It's a gas giant and the biggest planet

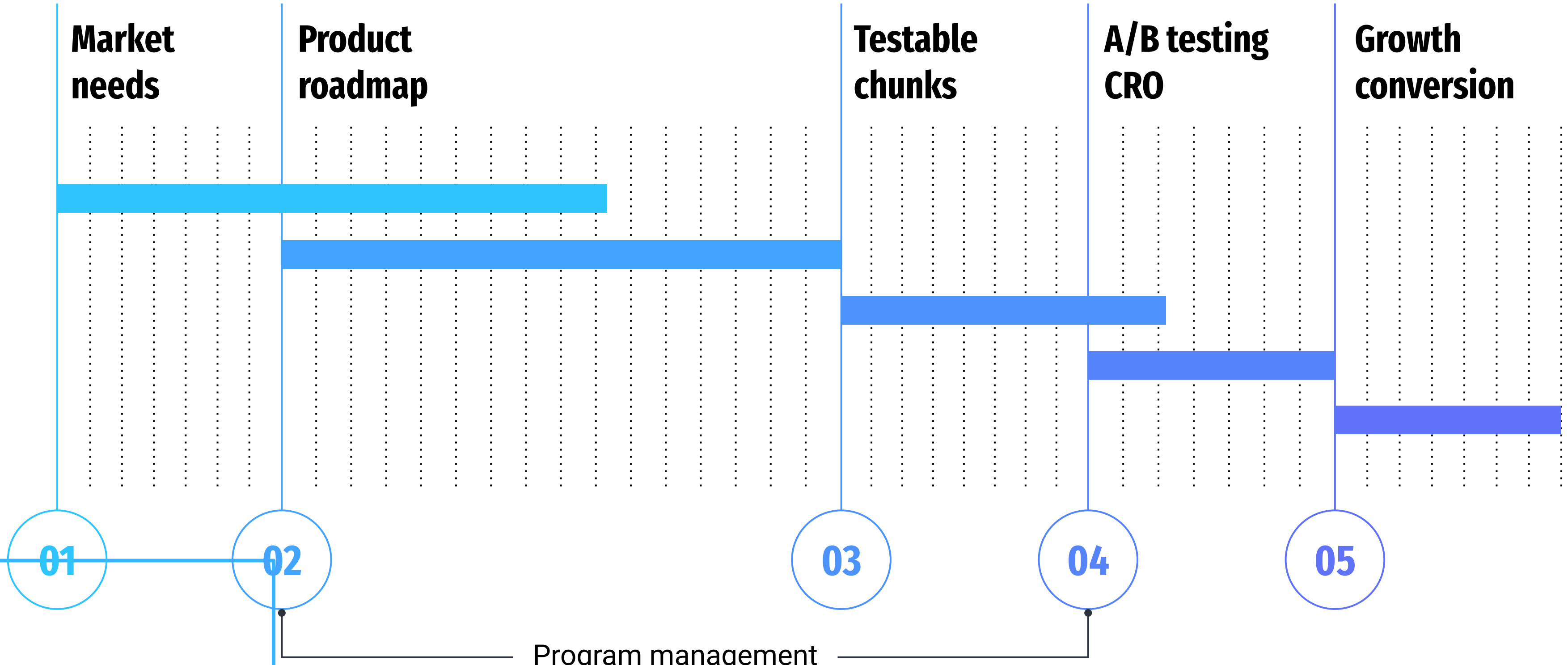
02
Generate idea
Venus is the second planet from the Sun

03
Implement changes
Despite being red, Mars is a cold place

04
Crown the champion
Despite being red, Mars is a cold place

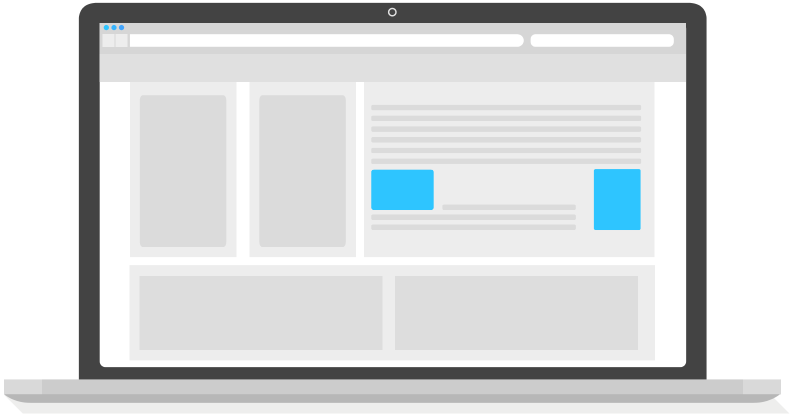
05
Test until significant
Mercury is the closest planet to the Sun

A/B Testing Infographics



A/B Testing Infographics

VS



Variation A

15% conversion

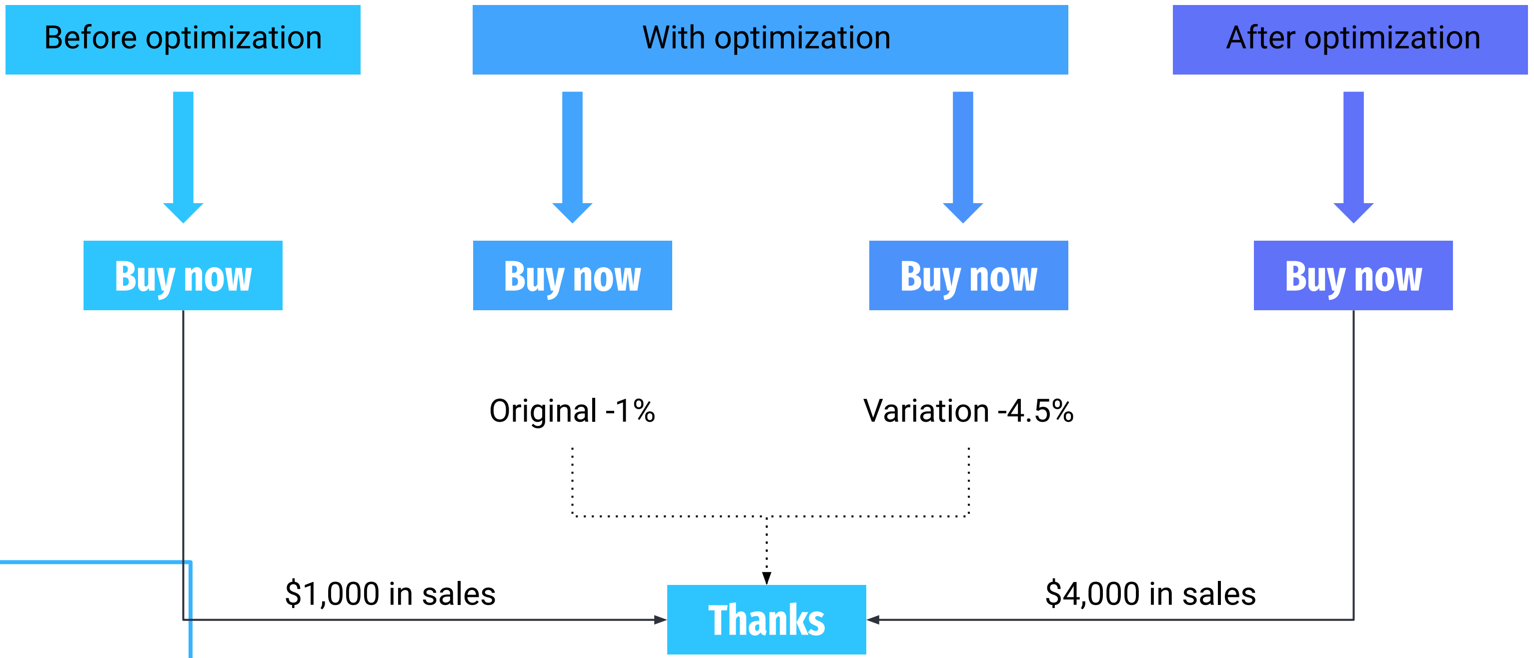
Venus has a beautiful name and is the second planet from the Sun in the Solar System

Variation B

50% conversion

Saturn is the ringed planet. It's composed mostly of hydrogen and helium

A/B Testing Infographics



A/B Testing Infographics

A/B Testing overview

Despite being red, Mars is a cold place full of iron oxide dust

Step 1

Mercury is the smallest of them all

Step 2

Saturn is a gas giant and has several rings

Step 3

Jupiter is the biggest planet of them all

Step 4

Neptune is very far away from Earth

Step 5

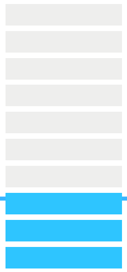
Earth is the planet where we live on

A/B Testing Infographics

A

Control

It has a beautiful name and is the second planet from the Sun



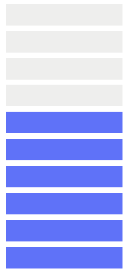
Conversion

30%

B

Variation

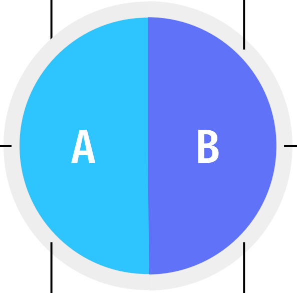
It's made of hydrogen and helium. It has rings. It's a gas giant



Conversion

60%

A/B Testing Infographics



8,543 Conversions

Jupiter is the biggest of them all

5,786 Clicks

Mercury is the smallest planet

4,765 Sign up

Saturn is a gas giant and has rings

Conversions 7,876

It’s the farthest planet from the Sun

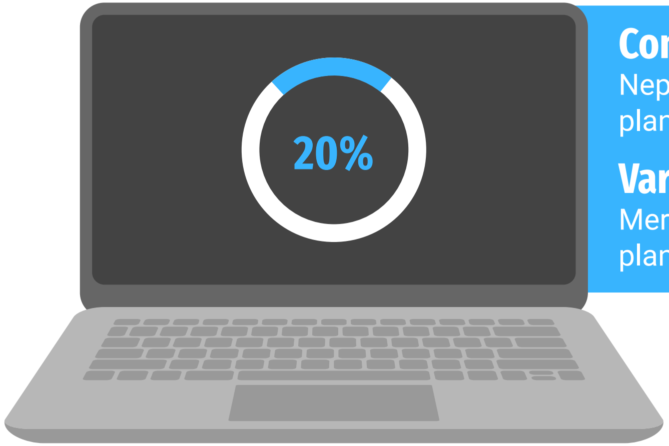
Clicks 3,543

Venus is the second planet from the Sun

Sign up 4,342

Earth is the third planet from the Sun

A/B Testing Infographics



Control
Neptune is the farthest planet from the Sun

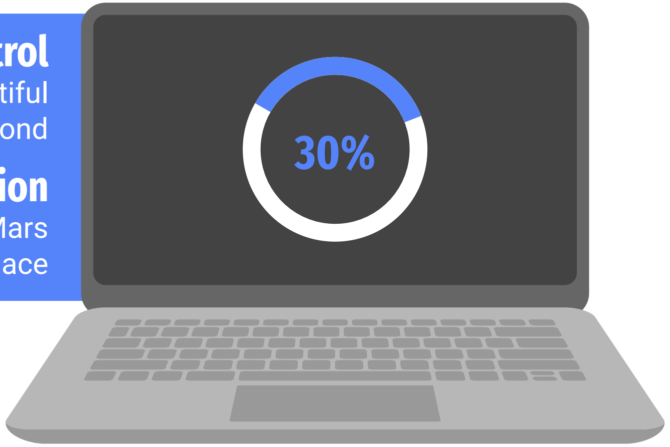
Variation
Mercury is the closest planet to the Sun

A

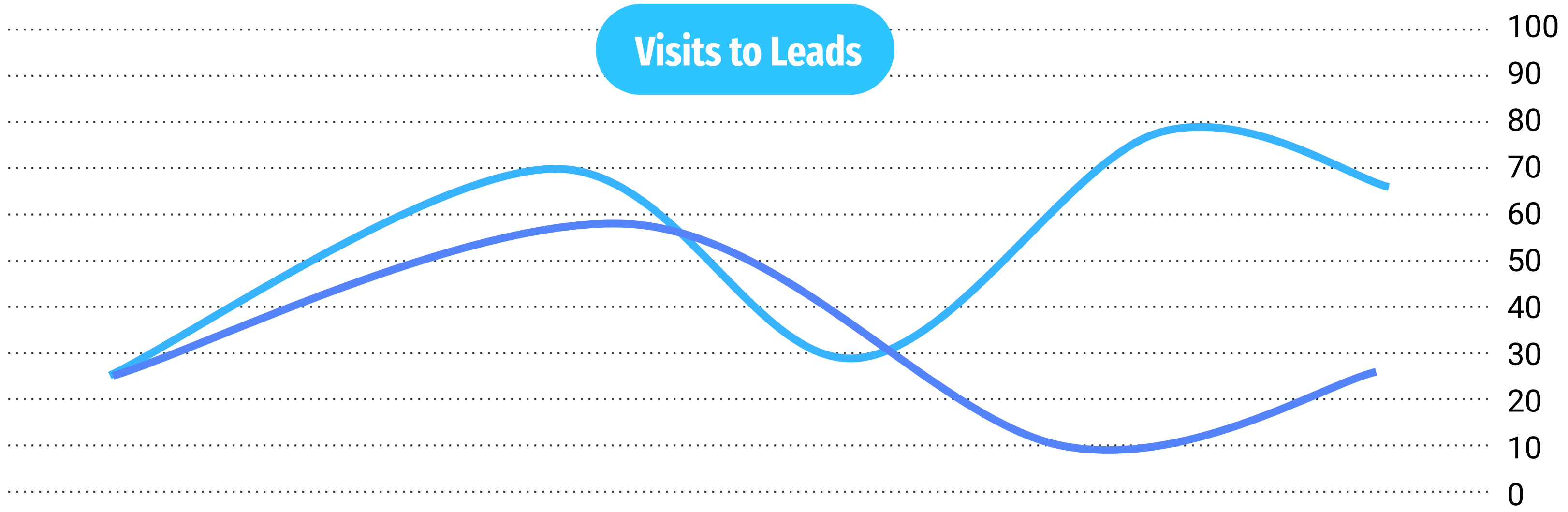
B

Control
Venus has a beautiful name and is the second

Variation
Despite being red, Mars is a very cold place



A/B Testing Infographics



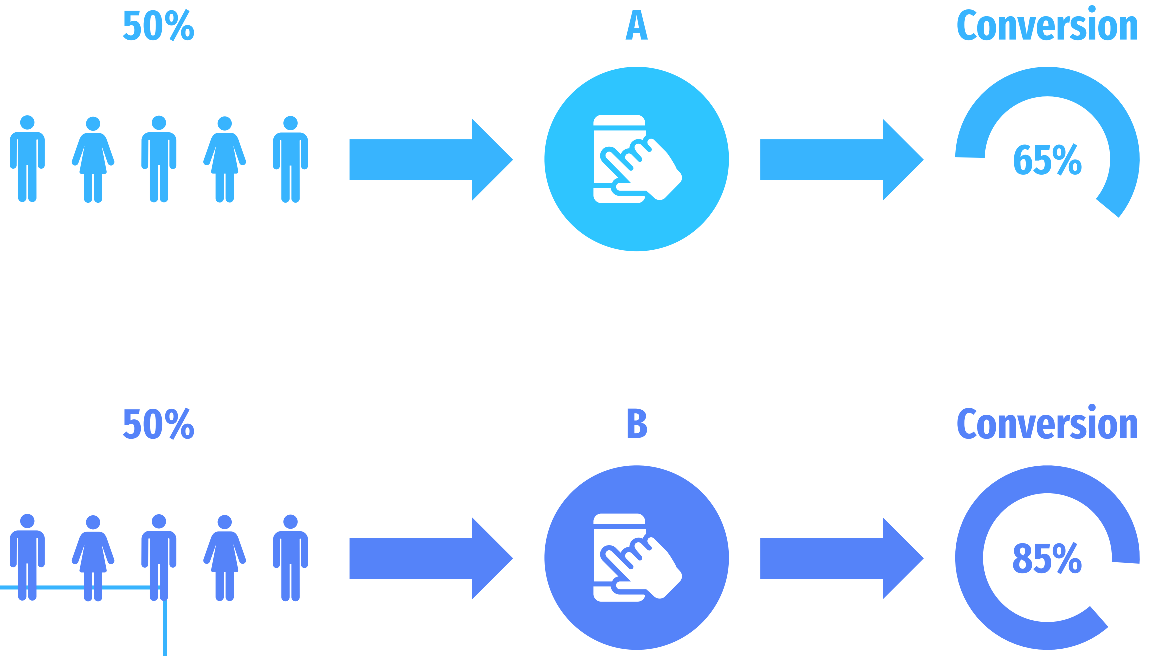
Variation A

Mercury is the closest planet to the Sun and the smallest one

Variation B

Despite being red, Mars is cold and full of iron oxide dust

A/B Testing Infographics



Jupiter is the biggest planet of them all

Saturn is a gas giant and has several rings

A/B Testing Infographics

01
Design base layout

Jupiter is a gas giant and also the biggest planet

02
Create variations

Mercury is the closest planet to the Sun

03
Define test plan

Despite being red, Mars is a very cold place

04
Collect data

Venus is the second planet from the Sun

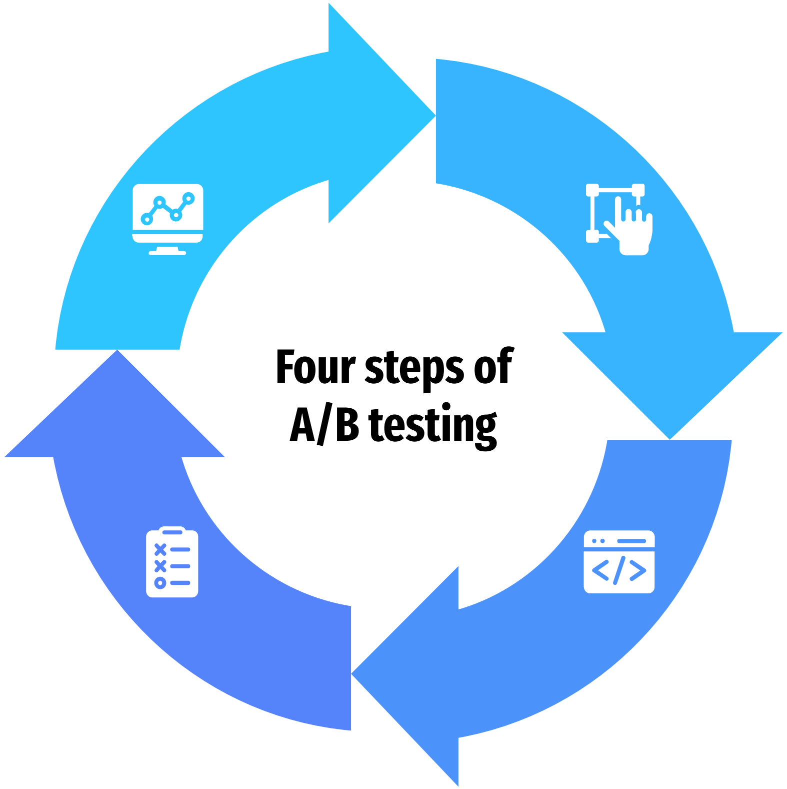
05
Run with best option

Neptune is the farthest planet from the Sun

06
Set up another A/B test

Earth is the only planet that harbors life

A/B Testing Infographics



01

Analyze data

Mercury is the closest planet to the Sun and the smallest one

02

Hypothesis

It's a gas giant and the biggest planet in the Solar System

03

Experiment

Venus is the second planet from the Sun and it's terribly hot

04

Evaluate results

Despite being red, Mars is a cold place full of iron oxide dust

A/B Testing Infographics

	Visitors	Conversions	Conversion rate	Results
A	8343	189	4%	Jupiter is a gas giant and also the biggest planet
B	7849	173	3%	Venus is the second planet from the Sun

A/B Testing Infographics

A

Mercury
Mercury is the closest planet to the Sun

Mars
Despite being red, Mars is a cold place

Jupiter
It’s a gas giant and the biggest planet

Venus
Venus is the second planet from the Sun

Saturn
Saturn is the ringed one. It’s a gas giant

B

Mercury
Mercury is the closest planet to the Sun

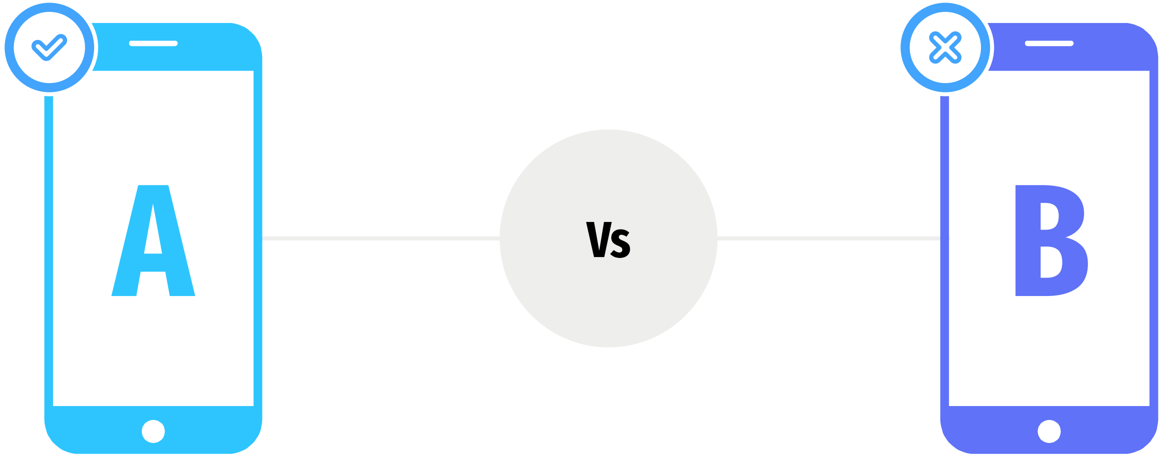
Mars
Despite being red, Mars is a cold place

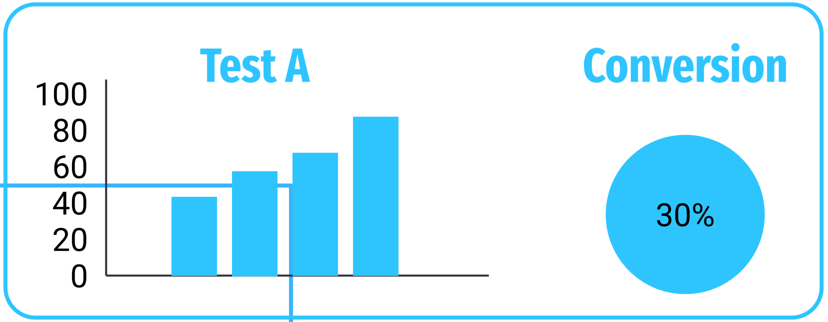
Jupiter
It’s a gas giant and the biggest planet

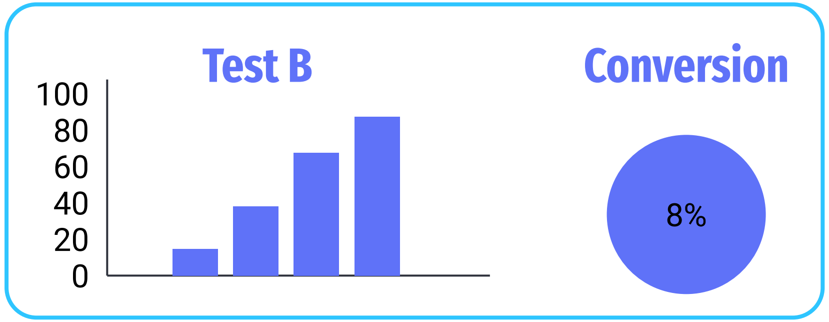
Venus
Venus is the second planet from the Sun

Saturn
Saturn is the ringed one. It’s a gas giant

A/B Testing Infographics





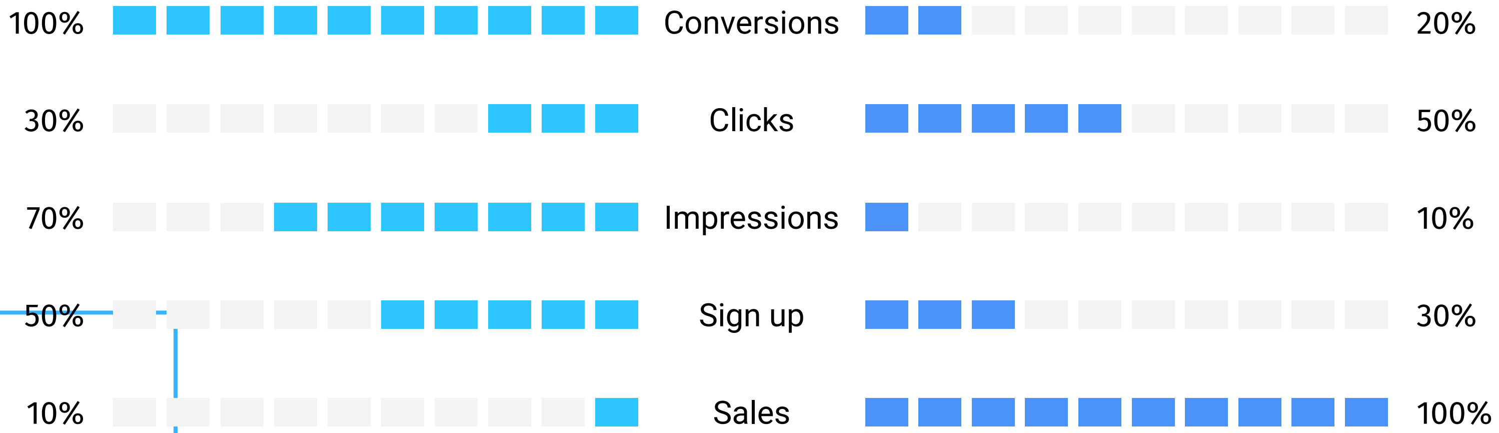


A/B Testing Infographics

Variation A vs **Variation B**

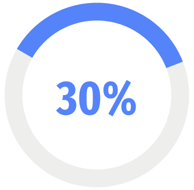
Mercury is the closest planet to the Sun

Venus is the second planet from the Sun



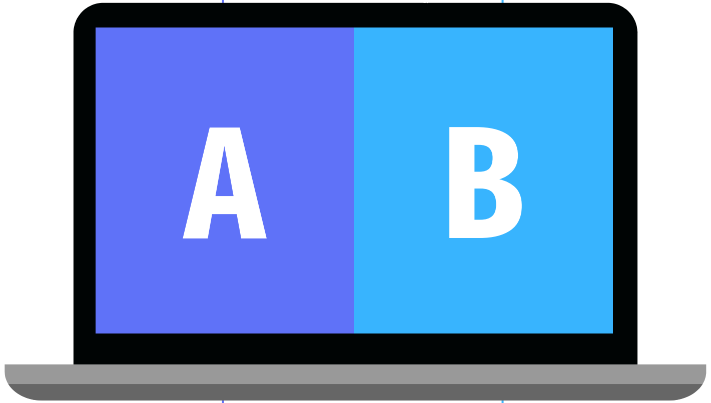
A/B Testing Infographics

Conversion

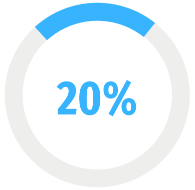


Control
Venus is the second planet from the Sun

Variation
Despite being red, Mars is a very cold place



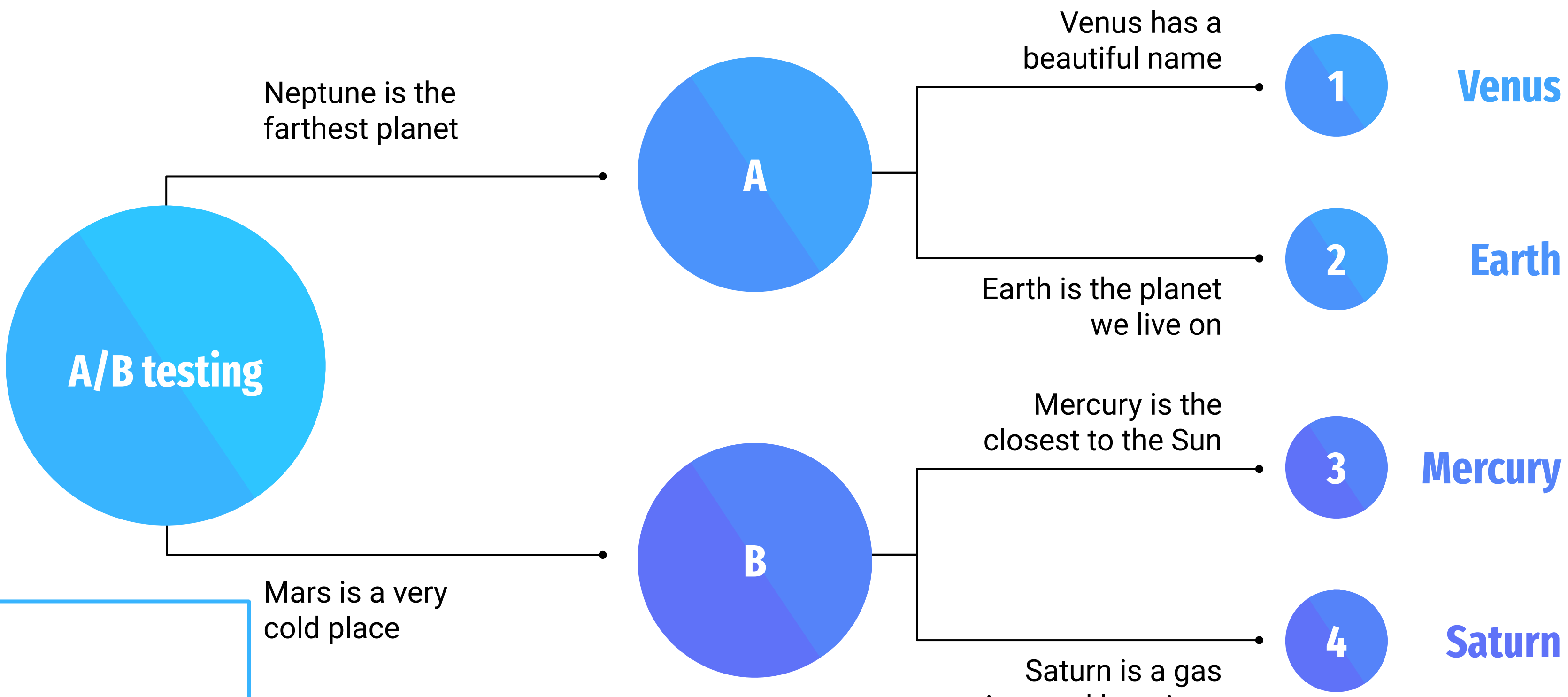
Conversion



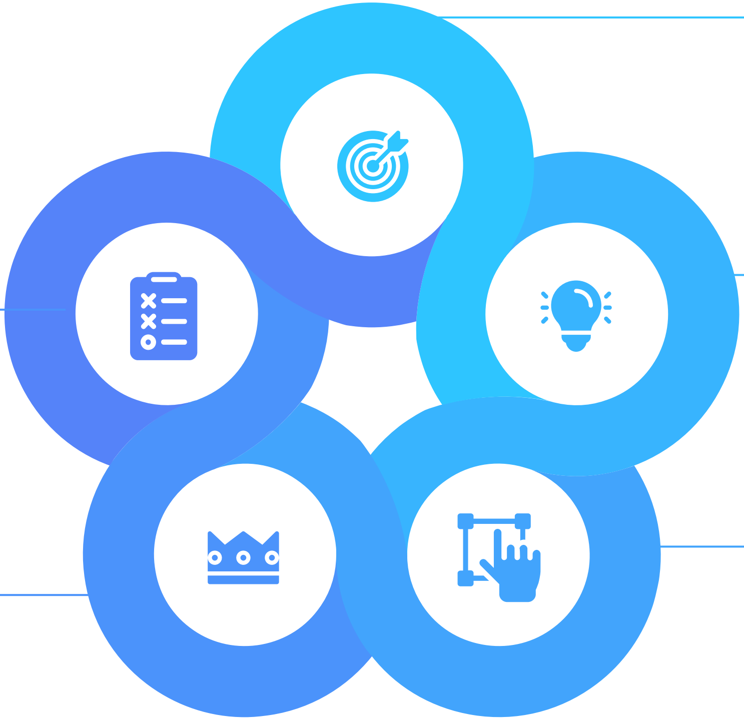
Control
Neptune is the farthest planet from the Sun

Variation
Mercury is the closest planet to the Sun

A/B Testing Infographics


Neptune is the farthest planet
A/B testing
Mars is a very cold place
A
B
Venus has a beautiful name
Earth is the planet we live on
Mercury is the closest to the Sun
Saturn is a gas giant and has rings
1
Venus
2
Earth
3
Mercury
4
Saturn

A/B Testing Infographics



01
Define goal

It's a gas giant and the biggest planet

02
Generate idea

Venus is the second planet from the Sun

03
Implement changes

Despite being red, Mars is a cold place

04
Crown the champion

Despite being red, Mars is a cold place

05
Test until significant

Mercury is the closest planet to the Sun

A/B Testing Infographics

01

Define goal

It’s a gas giant and the biggest planet

02

Generate idea

Venus is the second planet from the Sun

03

Implement changes

Despite being red, Mars is a cold place

04

Crown the champion

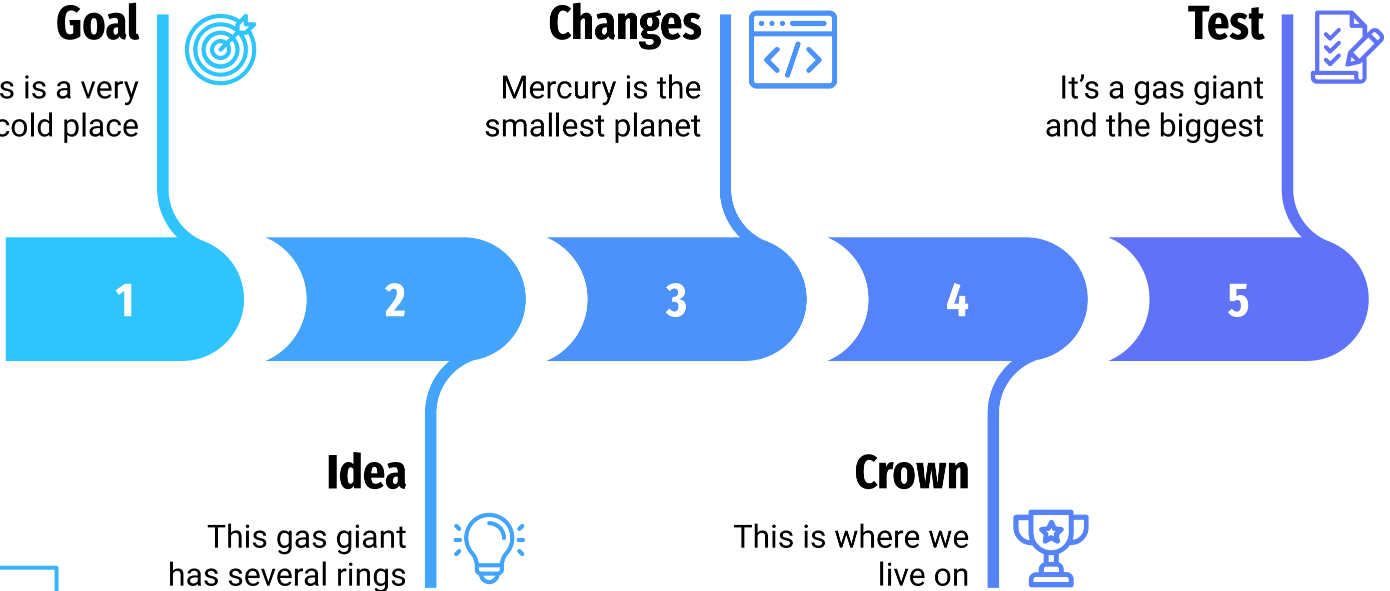
Mercury is the closest planet to the Sun

05

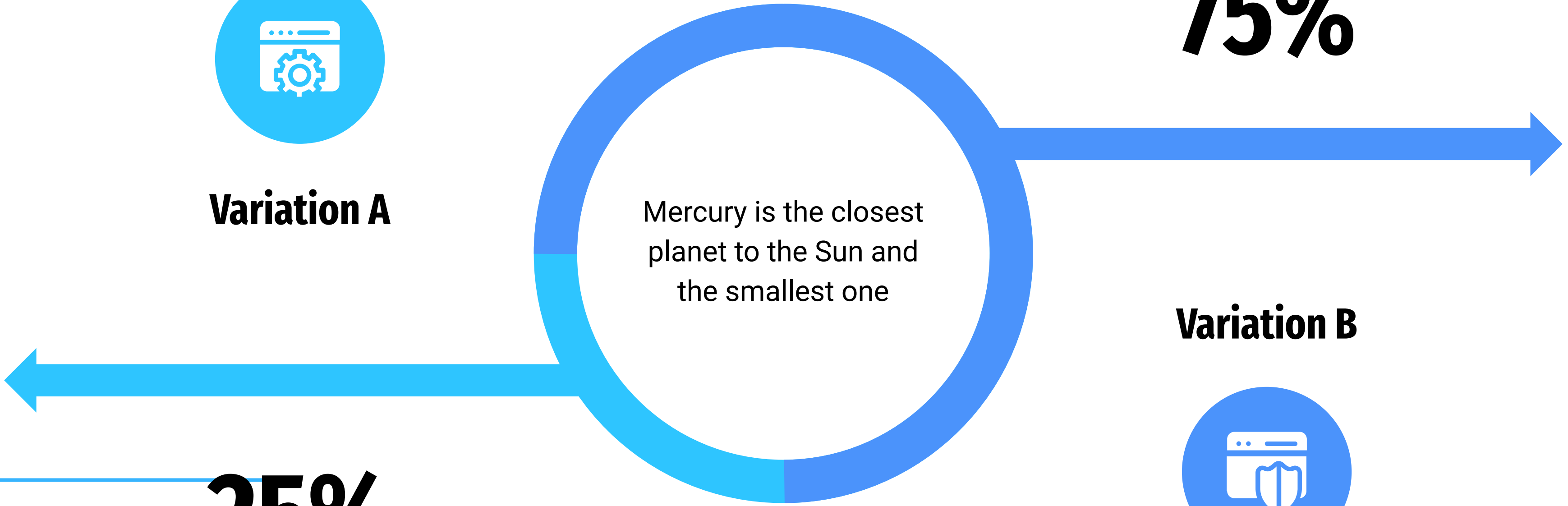
Test until significant

Despite being red, Mars is a cold place

A/B Testing Infographics


Goal
Mars is a very cold place
1
2
Idea
This gas giant has several rings
Changes
Mercury is the smallest planet
3
4
Crown
This is where we live on
Test
It’s a gas giant and the biggest
5

A/B Testing Infographics



Variation A

75%

Mercury is the closest planet to the Sun and the smallest one

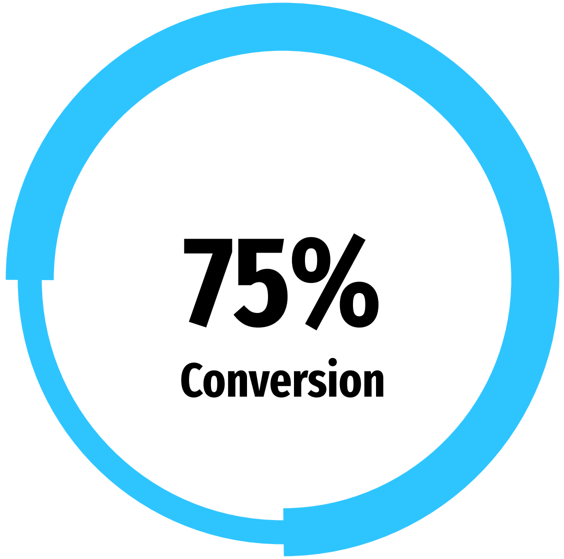
Variation B

25%

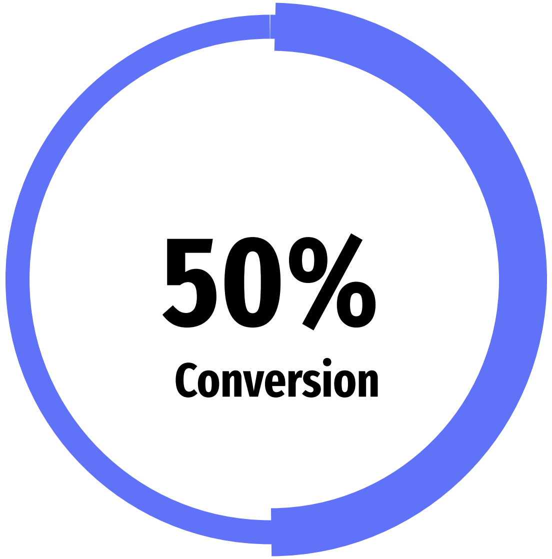
A/B Testing Infographics

	A	B
Visitors	8343	7849
Conversions	189	173
Conversion rate	4%	3%
Results	Jupiter is a gas giant and also the biggest planet	Venus is the second planet from the Sun

A/B Testing Infographics



VS



Variation A

Despite being red,
Mars is a cold place

Variation B

Venus is the second
planet from the Sun

A/B Testing Infographics

01

02

03

04

Analyze data

Mercury is the closest planet to the Sun and the smallest one of them all

Form an hypothesis

It's a gas giant and the biggest planet in the entire Solar System

Experiment

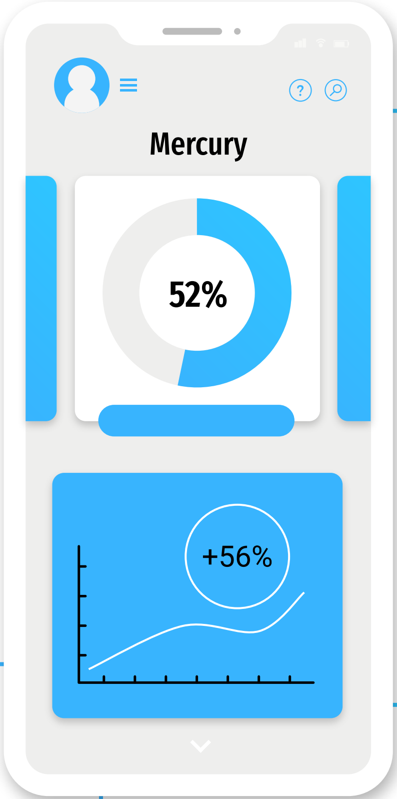
Despite being red, Mars is a very cold planet full of iron oxide dust

Evaluate results

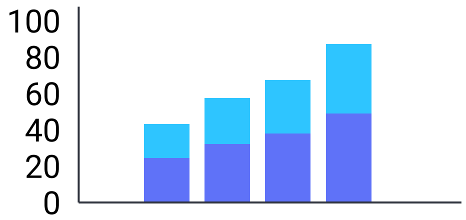
Venus is the second planet from the Sun and is terribly hot

A/B Testing Infographics

A



Test

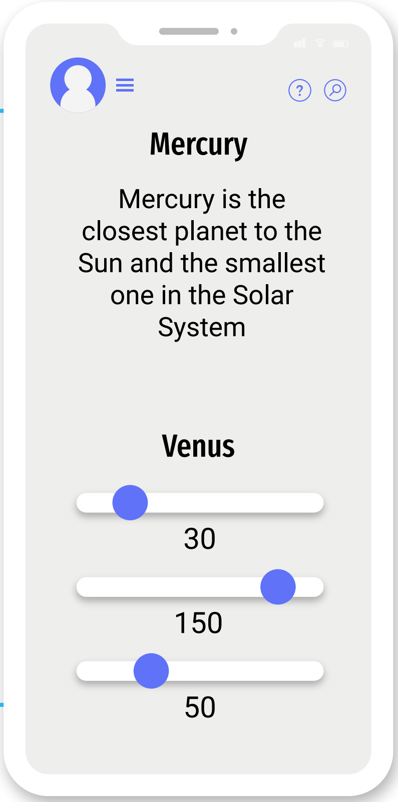


Conversion

30%

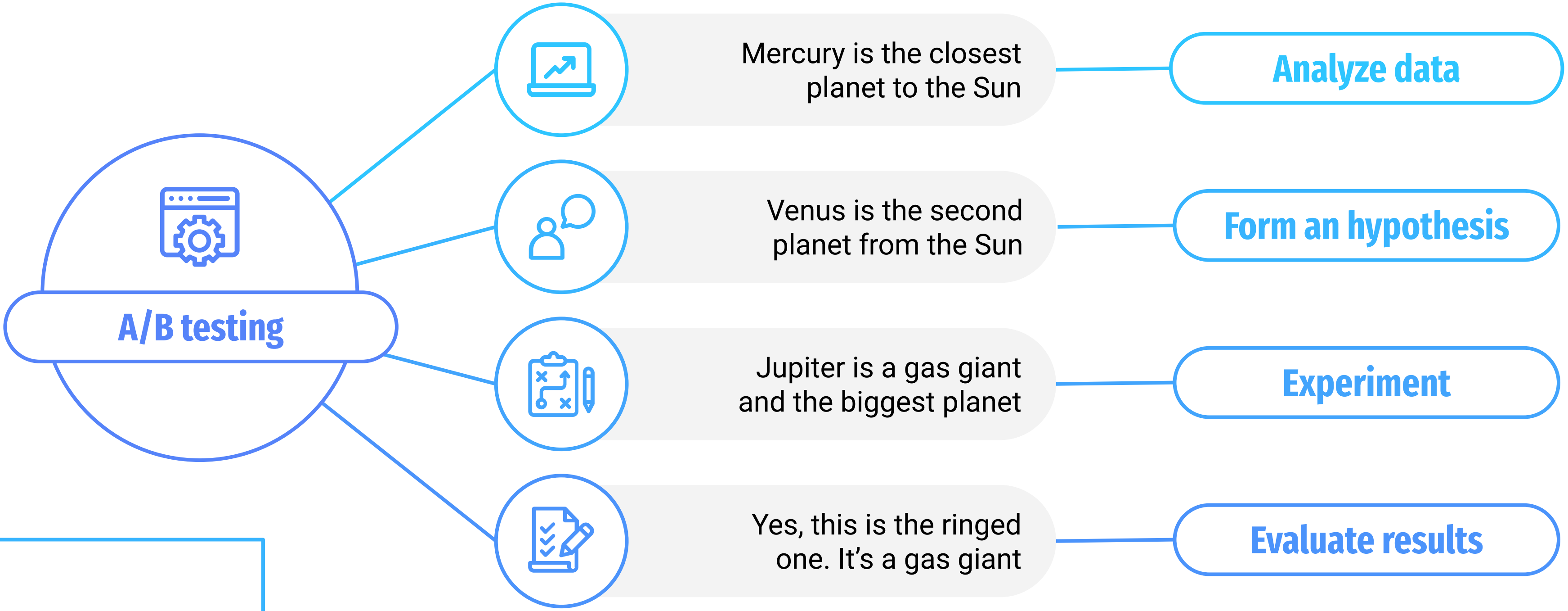
8%

B



Mercury is the closest planet to the Sun and the smallest one in the Solar System

A/B Testing Infographics



Mercury is the closest planet to the Sun

Venus is the second planet from the Sun

Jupiter is a gas giant and the biggest planet

Yes, this is the ringed one. It's a gas giant

Escopo

Data

Earth is the third planet from the Sun

Hypothesis

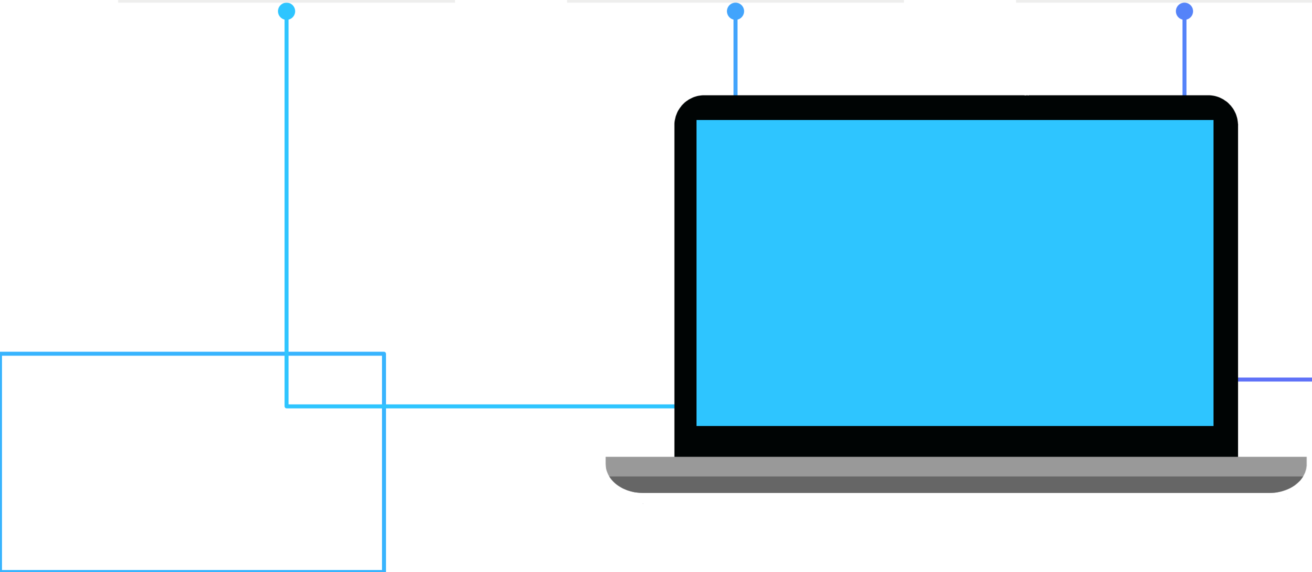
It’s the farthest planet from the Sun

Experiment

Despite being red, Mars is a cold place

Evaluation

It’s the closest object to the Sun



Infographics

You can add and edit some **infographics** to your presentation to present your data in a visual way.

- Choose your favourite infographic and insert it in your presentation using Ctrl C + Ctrl V or Cmd C + Cmd V in Mac.
- Select one of the parts and **ungroup** it by right-clicking and choosing "Ungroup".
- **Change the color** by clicking on the paint bucket.
- Then **resize** the element by clicking and dragging one of the square-shaped points of its bounding box (the cursor should look like a double-headed arrow). Remember to hold Shift while dragging to keep the proportions.
- **Group** the elements again by selecting them, right-clicking and choosing "Group".
- Repeat the steps above with the other parts and when you're done editing, copy the end result and paste it into your presentation.
- Remember to choose the **"Keep source formatting"** option so that it keeps the design. For more info, please visit **Slidesgo School**.