



# PUC Minas

# Covering

Alunos:

- Arthur Amaral
- Guilherme Antônio

Orientadores:

- José Laerte Xavier
- Marco Rodrigo Costa

# Categorização

## CBsoft

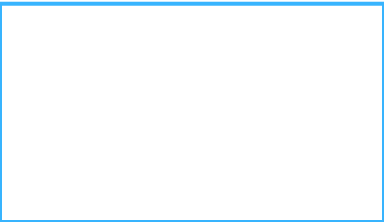
Área de Teste de  
Software e  
Engenharia de  
Software

## Desenvolvedores

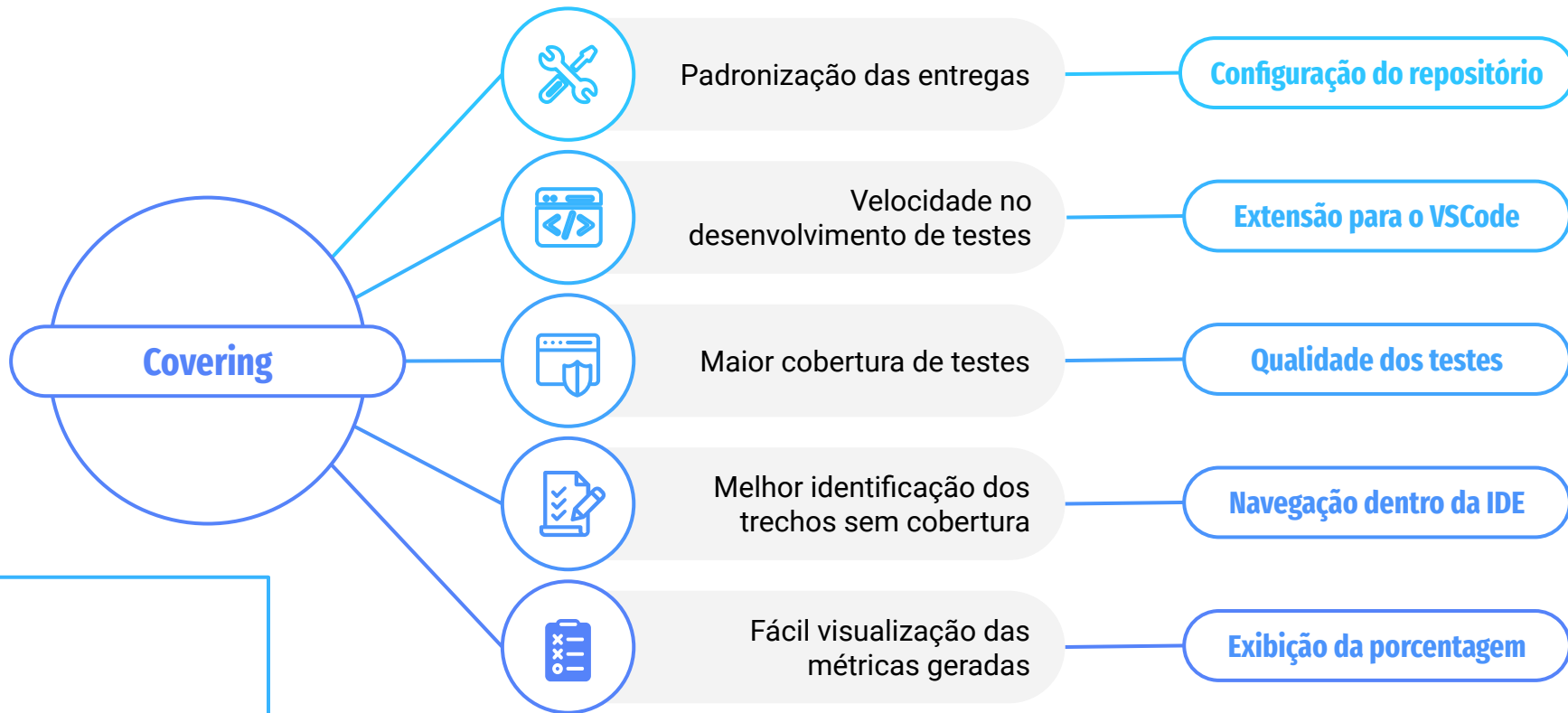
Benefícios aos  
desenvolvedores de  
software durante o  
processo de  
desenvolvimento e  
software

## Características

Qualidade da  
construção de testes  
medindo a cobertura  
do projeto e exibindo  
em tempo real para o  
desenvolvedor no  
próprio ambiente de  
desenvolvimento



# Objetivos



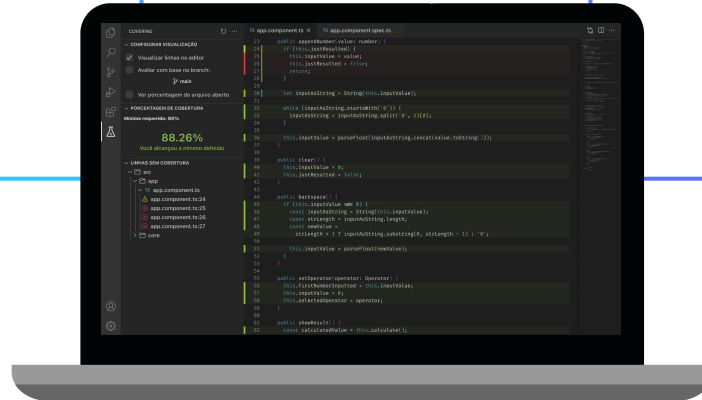
# Escopo

Extensão pra o  
VSCode

Leitura de arquivo de  
teste

Destaque das linhas

Vizualização de  
cobertura



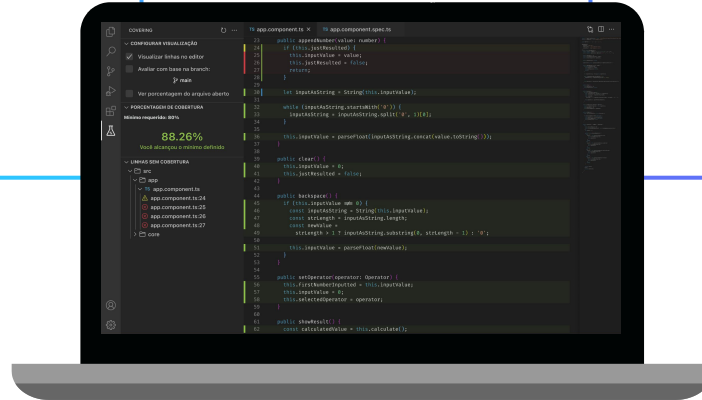
# Escopo

Comparação de  
cobertura entre  
*branches*


Bloqueio de *push*

Configuração da  
extensão


Suporte oficial às  
linguagens Javascript  
e Typescript




# Fora do Escopo



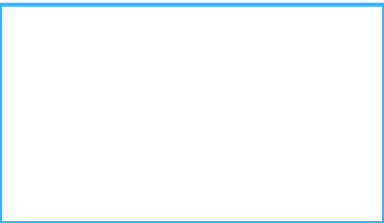
Gerar o arquivo de  
relatório de cobertura



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



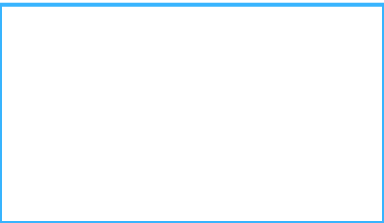
04

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



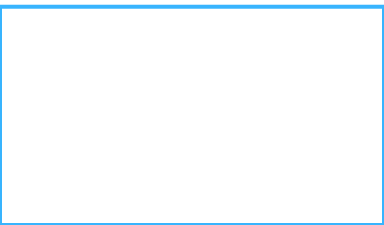
**Desenvolvedor**

Utilizador da extensão na IDE



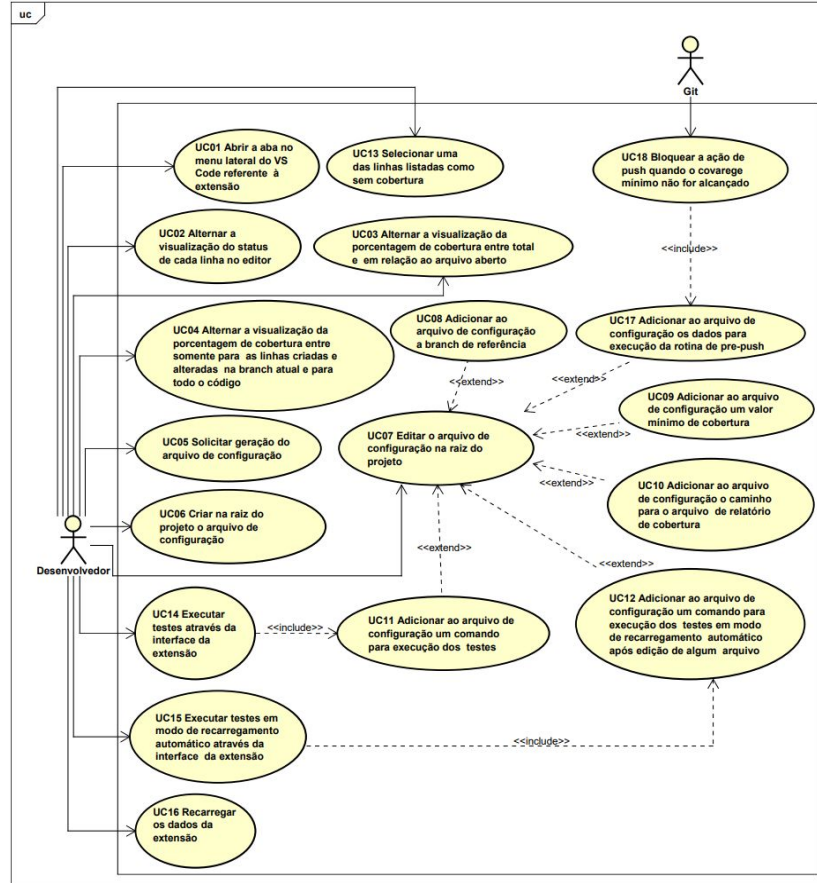
**Git**

Íntegra com a extensão, para  
que as funcionalidades sejam  
implementadas





# Diagrama de Caso de Uso



# 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

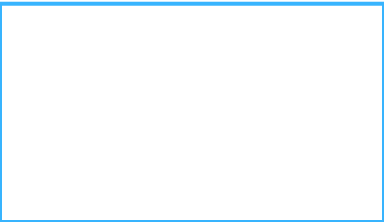
Criar na raiz do projeto o arquivo de configuração

UC08

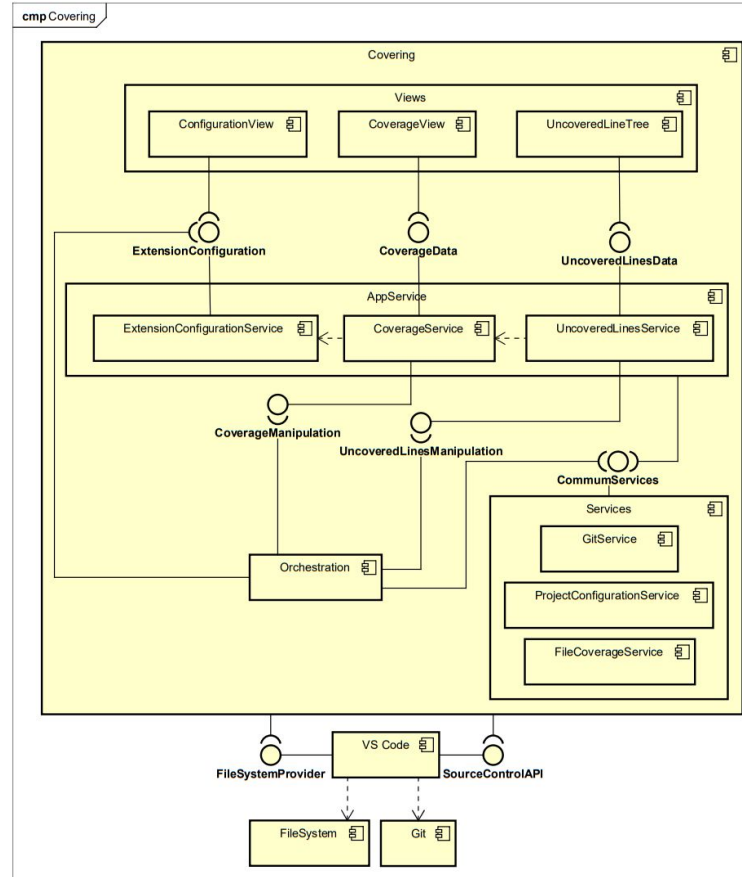
Adicionar ao arquivo de configuração a *branch* de referência

UC09

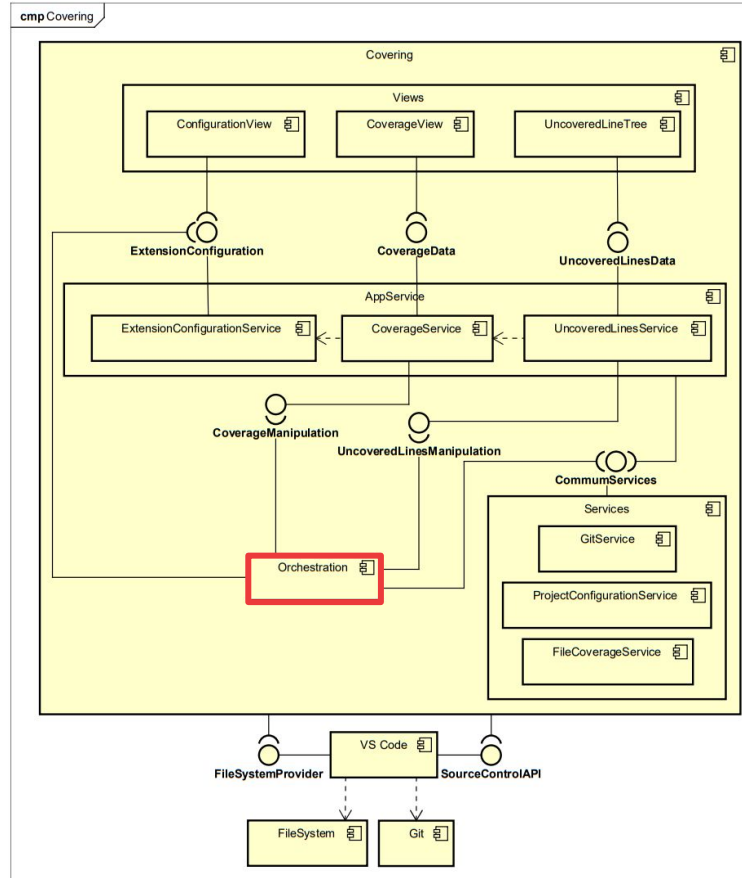
Adicionar ao arquivo de configuração um valor mínimo de cobertura



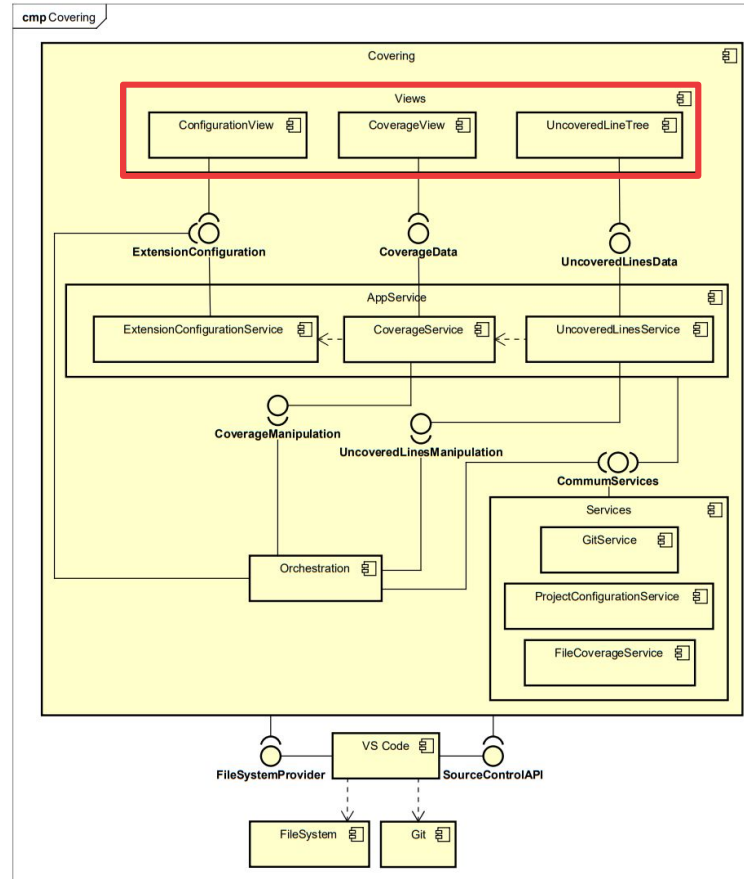
# Diagrama de Componentes



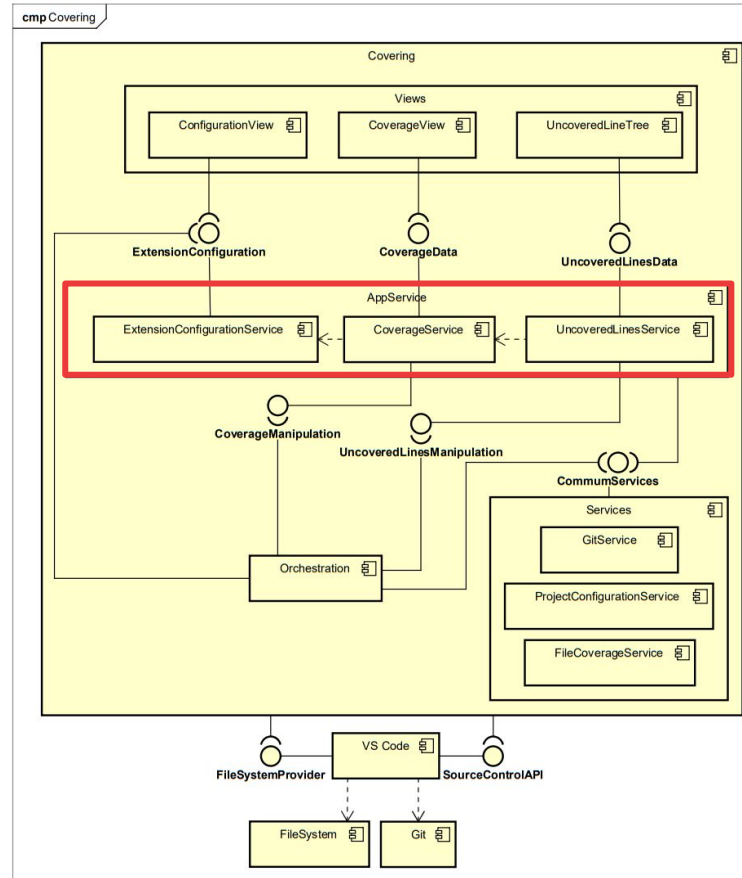
# Diagrama de Componentes



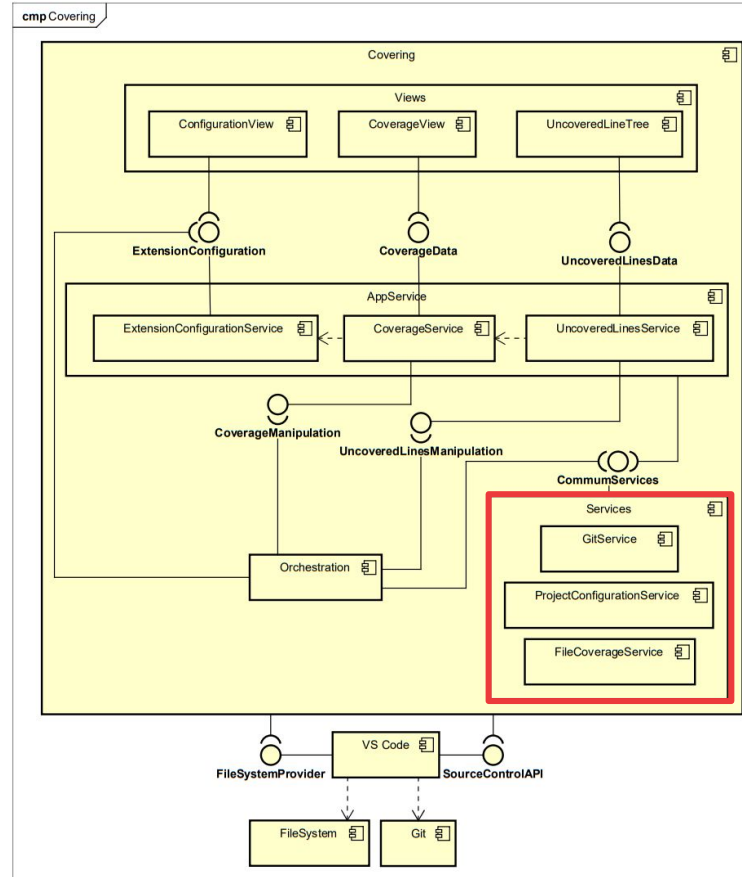
# Diagrama de Componentes



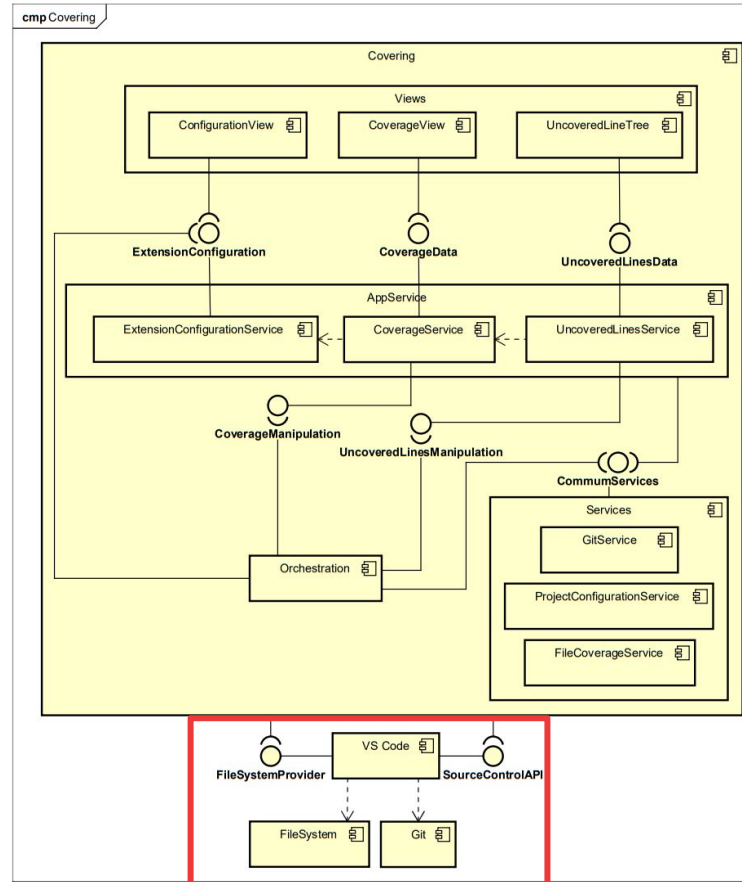
# Diagrama de Componentes



# 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.

## *AppService*

Responsáveis pelos serviços que geram os dados da aplicação.

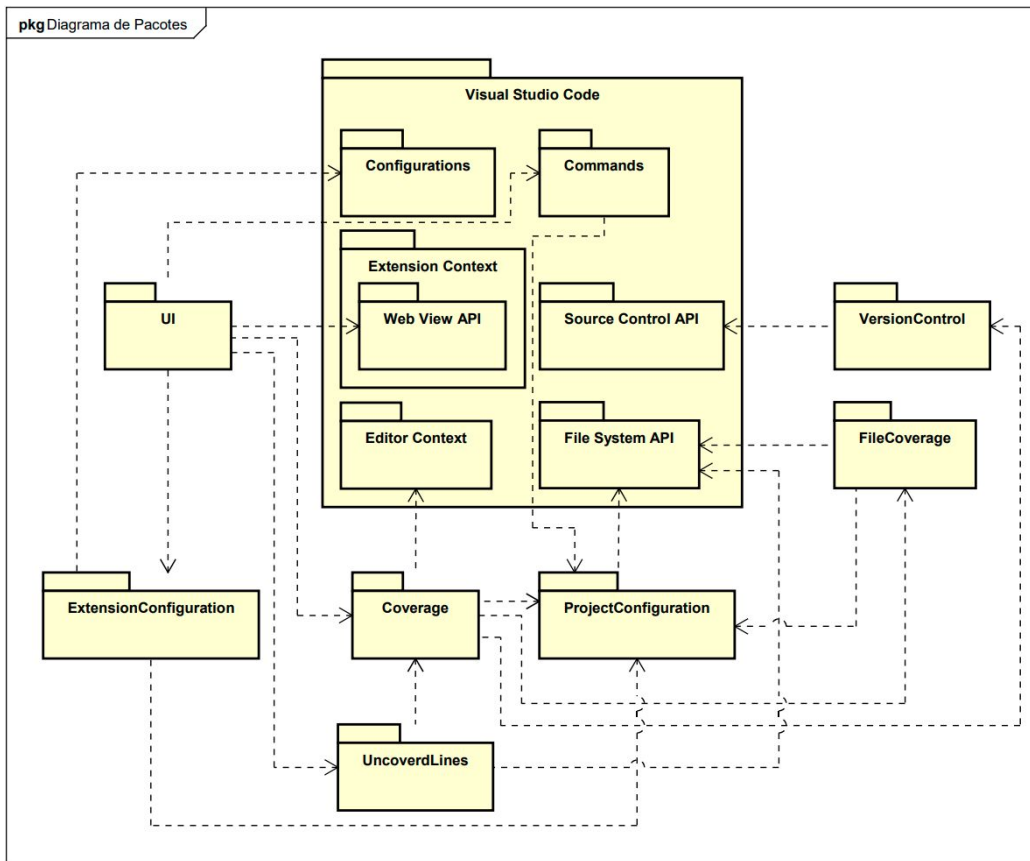
## *Service*

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

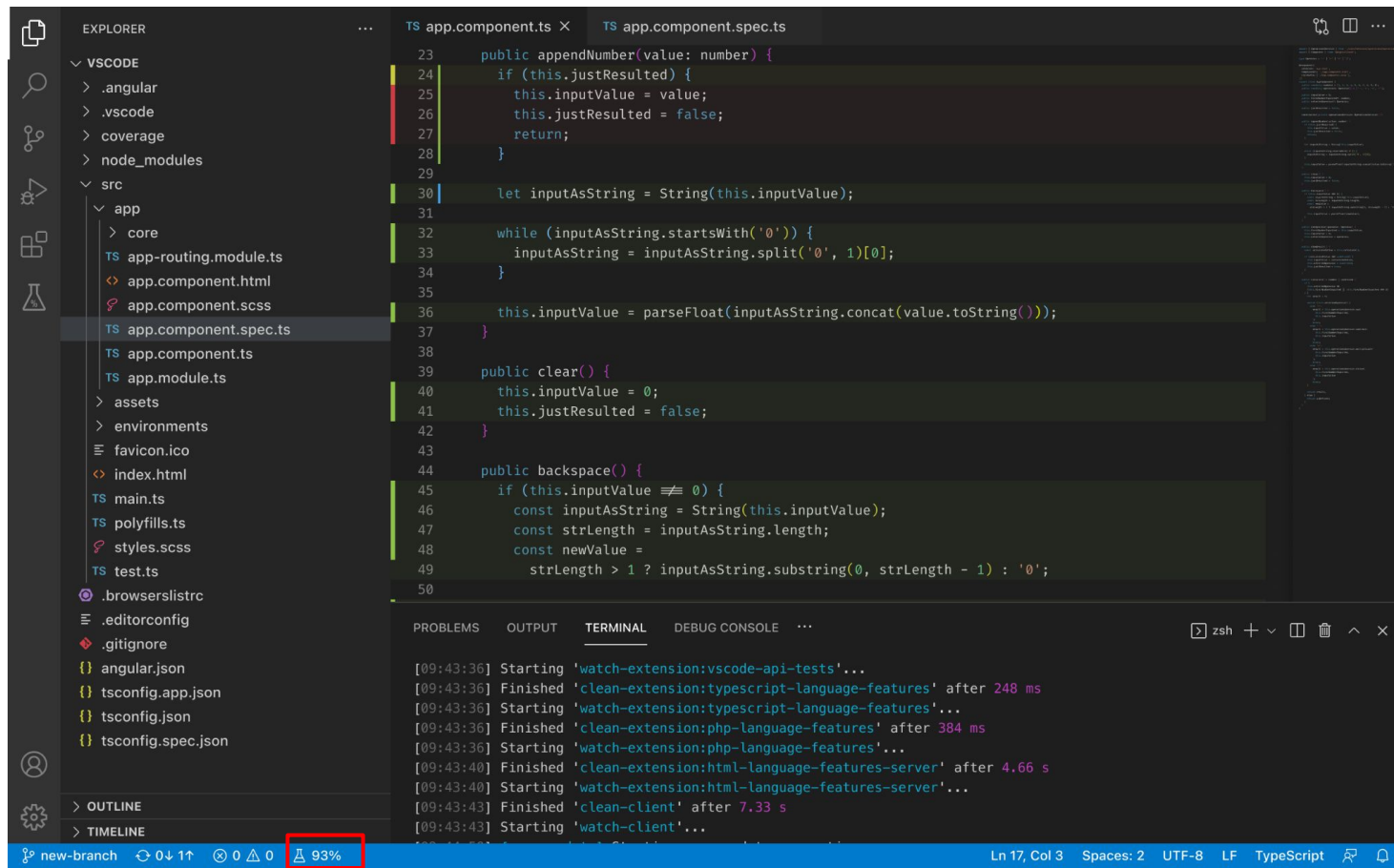
The image displays a Visual Studio Code editor interface for a TypeScript project. The Explorer sidebar on the left shows the project structure, with the file `app.component.spec.ts` selected under the `src/app` directory. The main editor window shows the implementation of the `app.component.ts` file, which includes the following code:

```
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     }
```

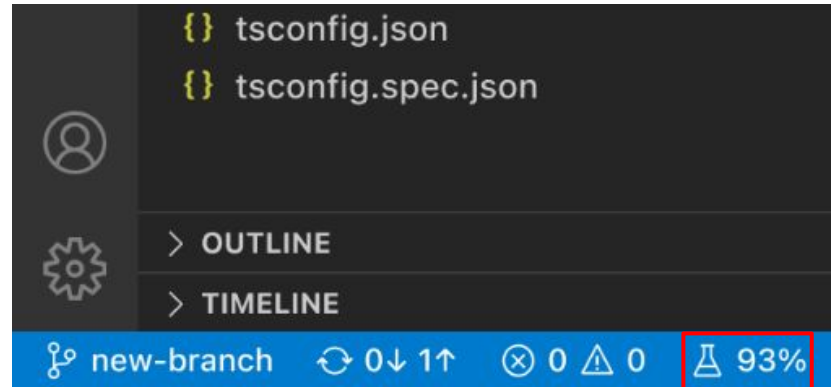
The bottom panel of the editor shows the TERMINAL output, which includes logs for extension installations and server startups:

```
[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'...
```

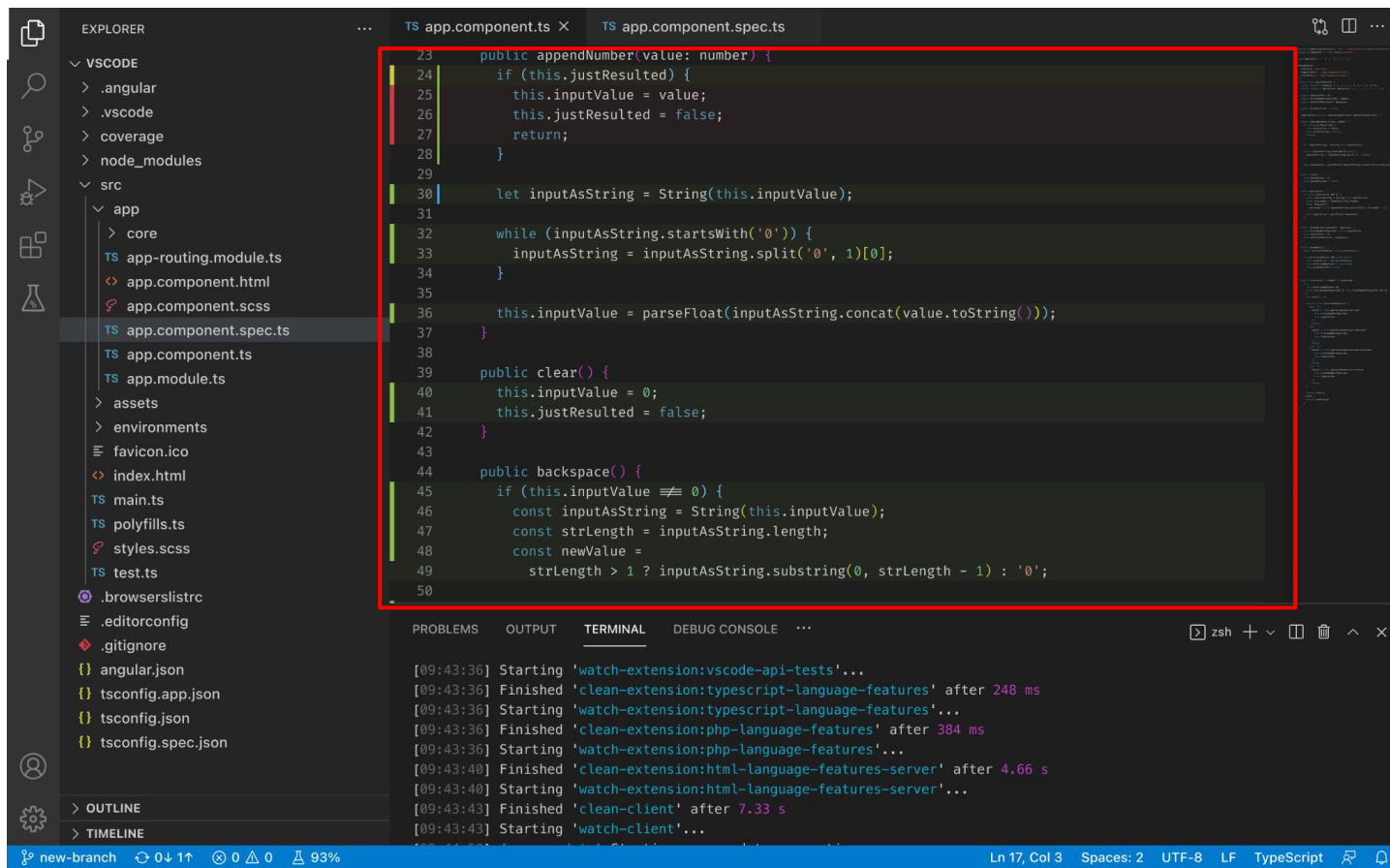
# Projeto de Interfaces com o Usuário



# Projeto de Interfaces com o Usuário



# Projeto de Interfaces com o Usuário



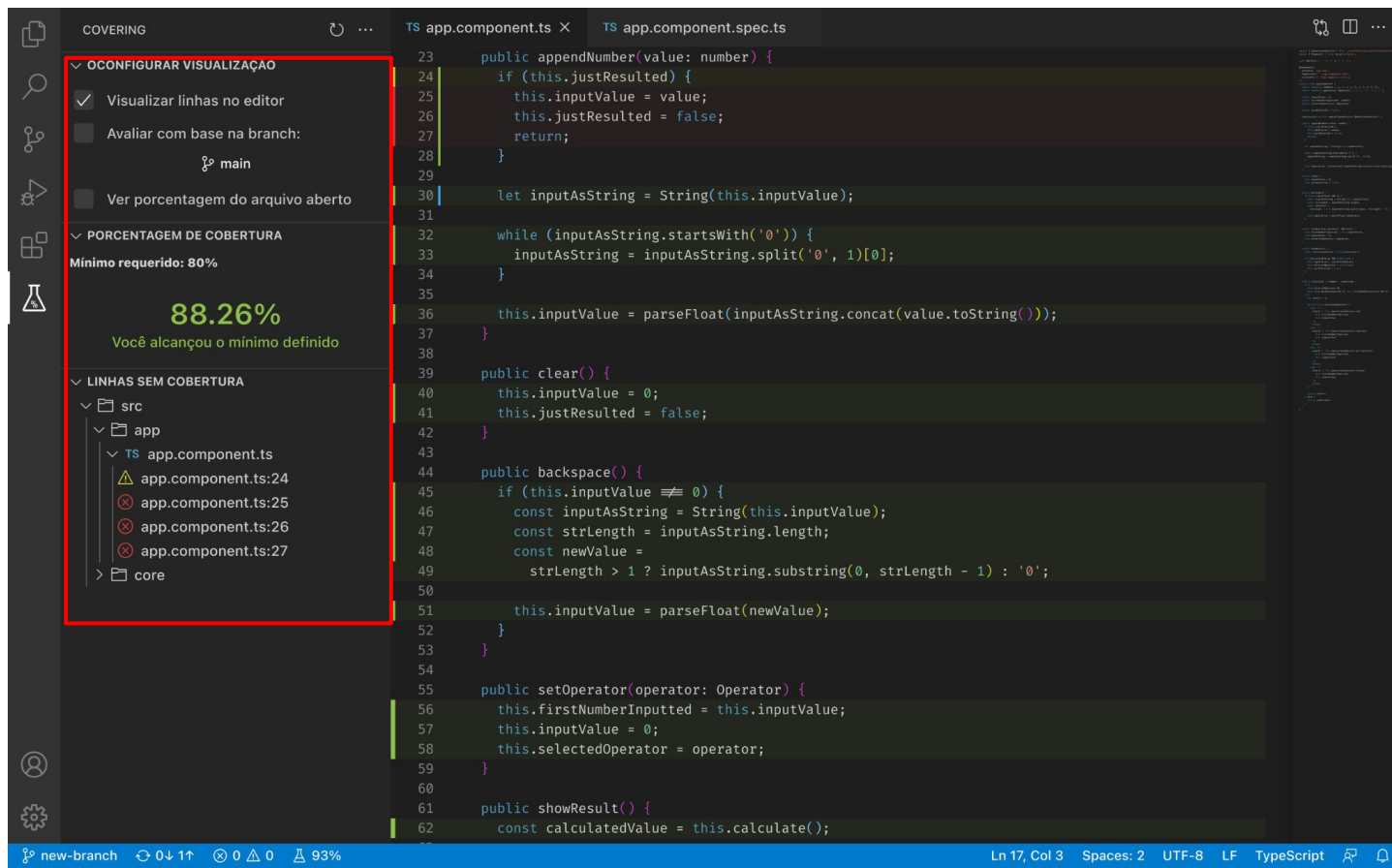
The screenshot displays a VS Code editor interface with a TypeScript file named `app.component.spec.ts` open. The Explorer sidebar on the left shows the project structure, including files like `app.component.html`, `app.component.spec.ts`, `app.component.ts`, `app.module.ts`, `assets`, `environments`, `favicon.ico`, `index.html`, `main.ts`, `polyfills.ts`, `styles.scss`, `test.ts`, `.browserslistrc`, `.editorconfig`, `.gitignore`, `angular.json`, `tsconfig.app.json`, `tsconfig.json`, and `tsconfig.spec.json`.

The main editor area shows the following TypeScript code:

```
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     }
```

The Terminal at the bottom shows the output of running the application, including messages from the watch-extension and clean-extension tasks.

# Projeto de Interfaces com o Usuário



The screenshot displays the Visual Studio Code interface with the 'COVERING' view on the left sidebar. A red box highlights the 'OCONFIGURAR VISUALIZAÇÃO' (Configure Visualization) section, which includes options for 'Visualizar linhas no editor' (checked), 'Avaliar com base na branch:' (set to 'main'), and 'Ver porcentagem do arquivo aberto' (unchecked). Below this, the 'PORCENTAGEM DE COBERTURA' (Coverage Percentage) section shows a 'Mínimo requerido: 80%' (Minimum required: 80%) and a current coverage of '88.26%', with a note 'Você alcançou o mínimo definido' (You reached the defined minimum). The 'LINHAS SEM COBERTURA' (Lines without coverage) section lists the following files:

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

The main editor shows the TypeScript code for 'app.component.ts' with line numbers 23 through 62. The code includes methods for appending numbers, clearing the input, backspacing, setting operators, and showing results. The status bar at the bottom indicates 'Ln 17, Col 3', 'Spaces: 2', 'UTF-8', 'LF', 'TypeScript', and a 93% coverage percentage.

# Projeto de Interfaces com o Usuário

**COVERING**

✓ **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**

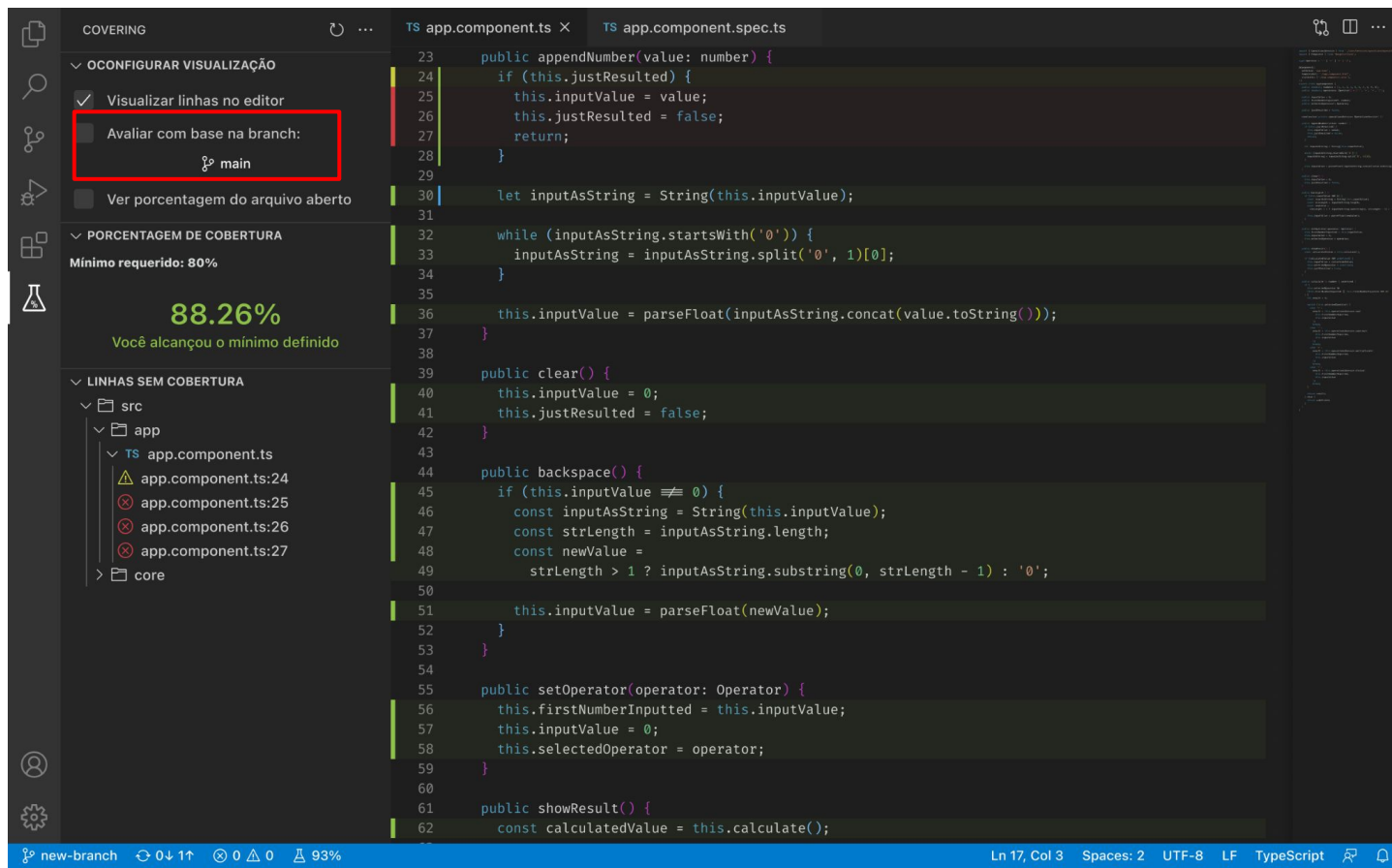
- src
  - app
    - TS app.component.ts
      - app.component.ts:24
      - app.component.ts:25
      - app.component.ts:26
      - app.component.ts:27

```
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();
```

Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript



# Projeto de Interfaces com o Usuário

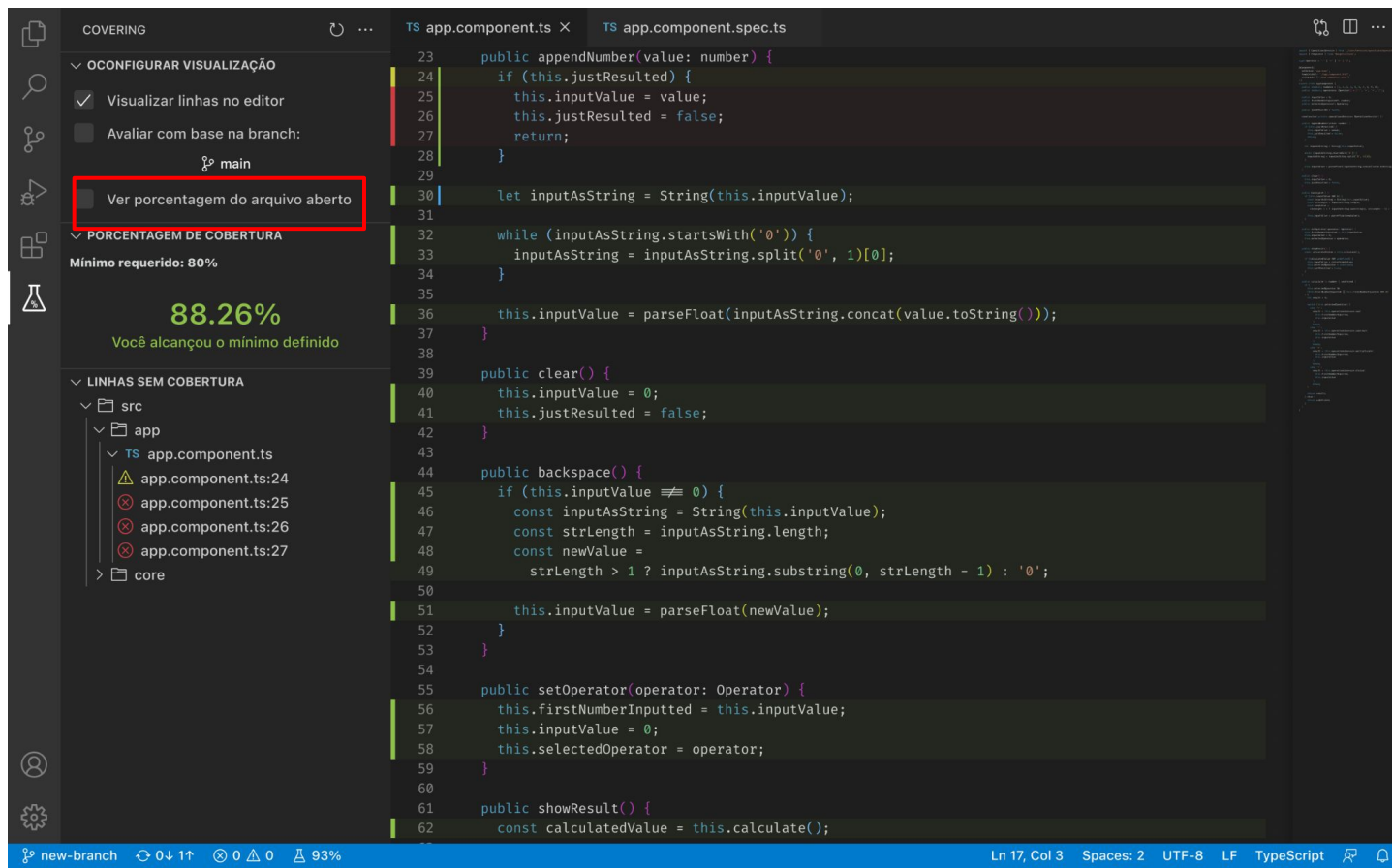


The screenshot displays the Visual Studio Code interface with a focus on code coverage analysis. The left sidebar contains a 'COVERING' panel with the following sections:

- 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

The main editor area shows the TypeScript code for `app.component.ts`. The code includes methods for appending numbers, clearing the input, backspacing, setting operators, and showing results. Line numbers 23 through 62 are visible. The status bar at the bottom indicates 'Ln 17, Col 3', 'Spaces: 2', 'UTF-8', 'LF', and 'TypeScript'.

# Projeto de Interfaces com o Usuário



The screenshot displays the Visual Studio Code interface with the 'COVERING' view on the left sidebar. The 'COVERING' view shows the following options:

- ☒ Visualizar linhas no editor
- ☐ Avaliar com base na branch: main
- ☐ Ver porcentagem do arquivo aberto (highlighted with a red box)

Below these options, the 'PORCENTAGEM DE COBERTURA' (Coverage Percentage) is shown as 88.26%, with a note 'Você alcançou o mínimo definido' (You reached the defined minimum). The 'LINHAS SEM COBERTURA' (Lines without coverage) section lists the following files:

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

The main editor shows the TypeScript code for app.component.ts, with line numbers 23 through 62. The code includes methods for appending numbers, clearing the input, backspacing, setting operators, and showing results.

```
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();
```

# Projeto de Interfaces com o Usuário

**COVERING**

✓ 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

```
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();
```

Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

# Projeto de Interfaces com o Usuário

**COVERING**

- ✓ 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

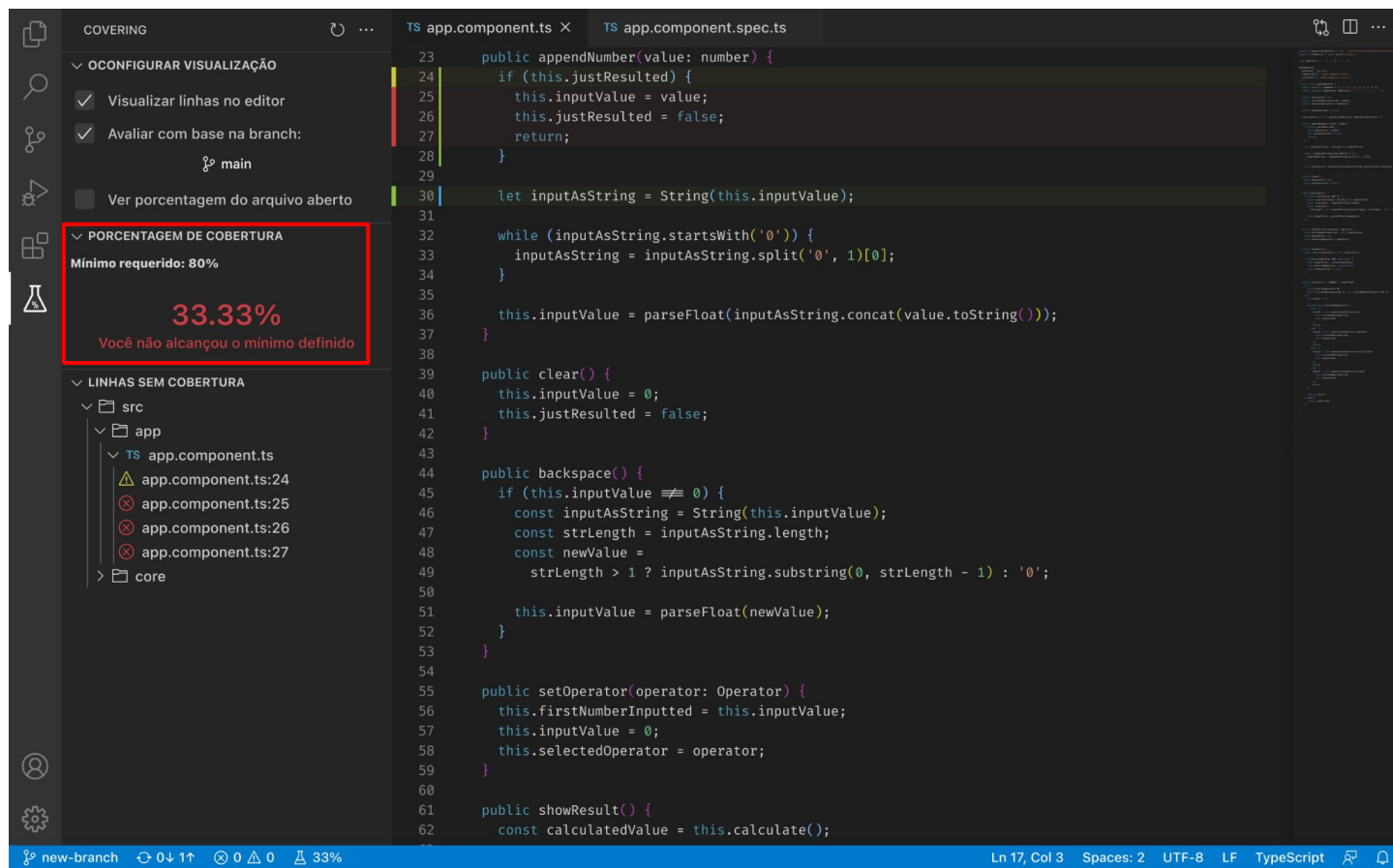
**▼ LINHAS SEM COBERTURA**

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

```
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();
```

Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

# Projeto de Interfaces com o Usuário



**COVERING**

- ✓ Visualizar linhas no editor
- ✓ Avaliar com base na branch: `main`
- Ver percentagem 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

```
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();
```

Ln 17, Col 3 Spaces: 2 UTF-8 LF TypeScript

# Projeto de Interfaces com o Usuário

The screenshot displays the Visual Studio Code interface with the following details:

- Left Sidebar (Explorer):**
  - COVERING** (expanded):
    - 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%** (highlighted in green)
      - Você não alcançou o mínimo definido
    - LINHAS SEM COBERTURA**
      - app.component.ts:24 (warning icon)
      - app.component.ts:25 (error icon)
      - app.component.ts:26 (error icon)
      - app.component.ts:27 (error icon)
- Main Editor:**
  - Files: `.coverconfig`, `TS app.component.ts`, `TS app.component.spec.ts`
  - Content of `.coverconfig`:

```
1 lcov_file=coverage/lcov.info
2
3 min_coverage=0.3
4
5 ref_branch=develop
6
7 run_test_with_coverage=npm run test:coverage -- --watch=false
8 run_test_with_coverage_watch_files=npm run test:coverage
9
```
- Status Bar:**
  - new-branch | 04 1↑ | 0 0 0 | 33%

# Projeto de Interfaces com o Usuário

The screenshot displays the VS Code interface with the COVERING extension. The left sidebar is divided into two sections: 'OCONFIGURAR VISUALIZAÇÃO' and 'PORCENTAGEM DE COBERTURA'.

**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**

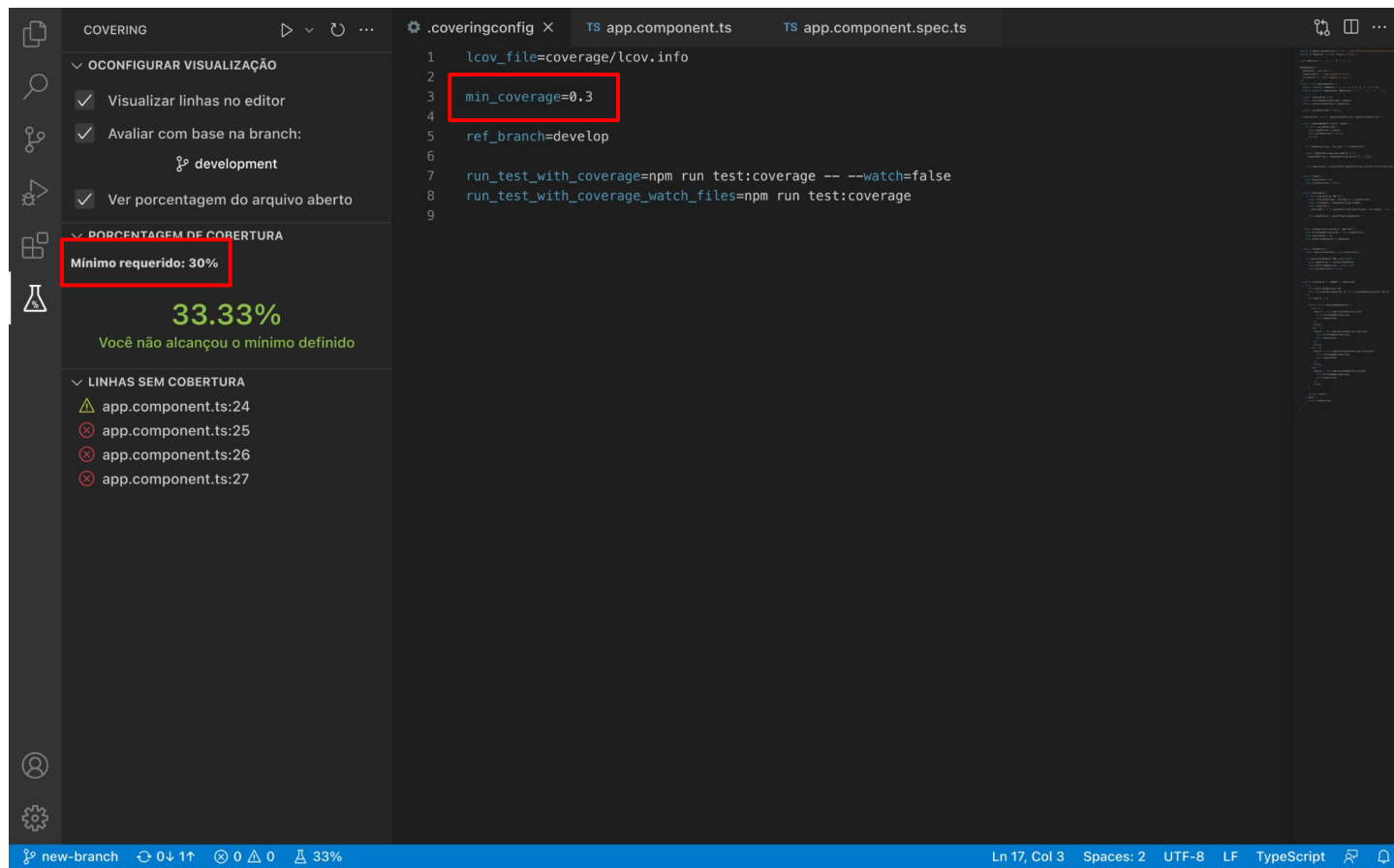
- ⚠ app.component.ts:24
- ✖ app.component.ts:25
- ✖ app.component.ts:26
- ✖ app.component.ts:27

The main editor shows the `.coveringconfig` file with the following content:

```
1 lcov_file=coverage/lcov.info
2
3 min_coverage=0.3
4
5 ref_branch=develop
6
7 run_test_with_coverage=npm run test:coverage -- --watch=false
8 run_test_with_coverage_watch_files=npm run test:coverage
9
```

The status bar at the bottom indicates the current branch is `new-branch` and the coverage is `33%`.

# Projeto de Interfaces com o Usuário



The screenshot displays the Visual Studio Code interface with a focus on code coverage. The left sidebar contains the 'COVERING' extension's settings and a summary of coverage results. The main editor shows the configuration file for the coverage tool.

**COVERING Settings:**

- OCONFIGURAR VISUALIZAÇÃO**
  - ☒ Visualizar linhas no editor
  - ☒ Avaliar com base na branch: development
  - ☒ Ver porcentagem do arquivo aberto
- PORCENTAGEM DE COBERTURA**
  - Mínimo requerido: 30%
- COBERTURA**
  - 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

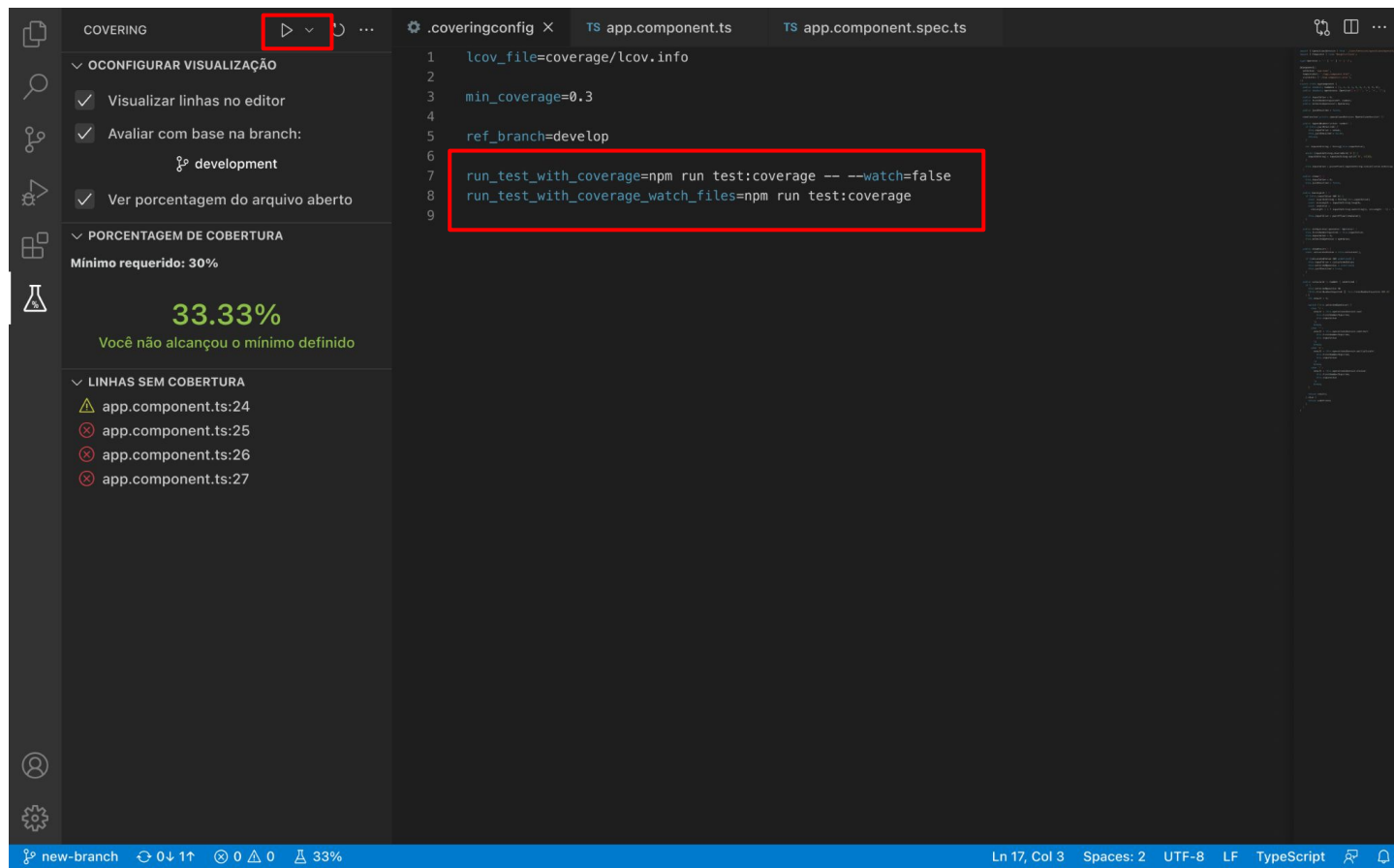
**Configuration File (.coveringconfig):**

```
1 lcov_file=coverage/lcov.info
2 min_coverage=0.3
3
4 ref_branch=develop
5
6
7 run_test_with_coverage=npm run test:coverage -- --watch=false
8 run_test_with_coverage_watch_files=npm run test:coverage
9
```

**Status Bar:** new-branch 0 0 0 33%



# Projeto de Interfaces com o Usuário



The screenshot displays the Visual Studio Code interface with the following elements:

- Left Sidebar (COVERING Panel):**
  - OCONFIGURAR VISUALIZAÇÃO:** Includes checkboxes for "Visualizar linhas no editor" (checked), "Avaliar com base na branch:" (checked, set to "development"), and "Ver porcentagem do arquivo aberto" (checked).
  - PORCENTAGEM DE COBERTURA:** Shows "Mínimo requerido: 30%" and a large green "33.33%" with the message "Você não alcançou o mínimo definido".
  - LINHAS SEM COBERTURA:** Lists four lines with red 'x' icons: "app.component.ts:24", "app.component.ts:25", "app.component.ts:26", and "app.component.ts:27".
- Top Bar:** Shows tabs for ".coveringconfig", "TS app.component.ts", and "TS app.component.spec.ts". A red box highlights the play button icon.
- Editor (app.component.spec.ts):** Contains the following configuration lines:

```
1 lcov_file=coverage/lcov.info
2
3 min_coverage=0.3
4
5 ref_branch=develop
6
7 run_test_with_coverage=npm run test:coverage -- --watch=false
8 run_test_with_coverage_watch_files=npm run test:coverage
9
```

A red box highlights lines 7 and 8.
- Bottom Bar:** Displays "new-branch", "04 1↑", "0 0", and "33%".
- Right Sidebar:** Shows a file explorer with various project files.

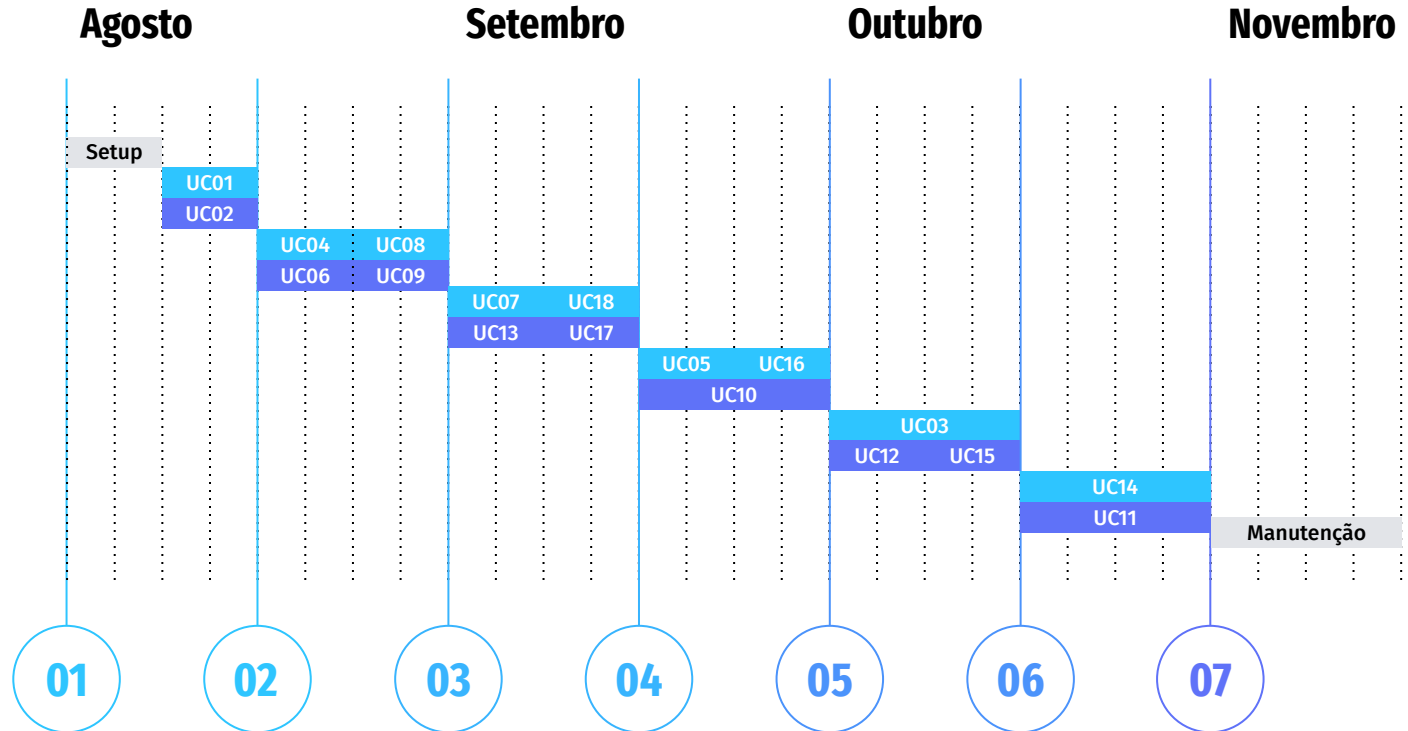
# Cronograma do TCC II

## 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

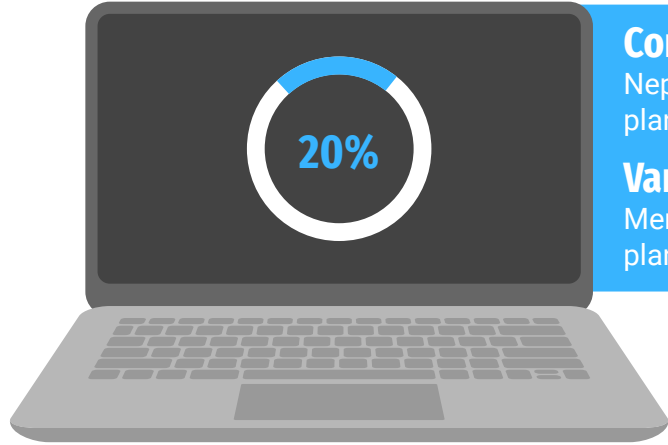
Neptune is very far away from Earth



### Step 5

Earth is the planet where we live on

# A/B Testing Infographics



## Control

Neptune is the farthest planet from the Sun

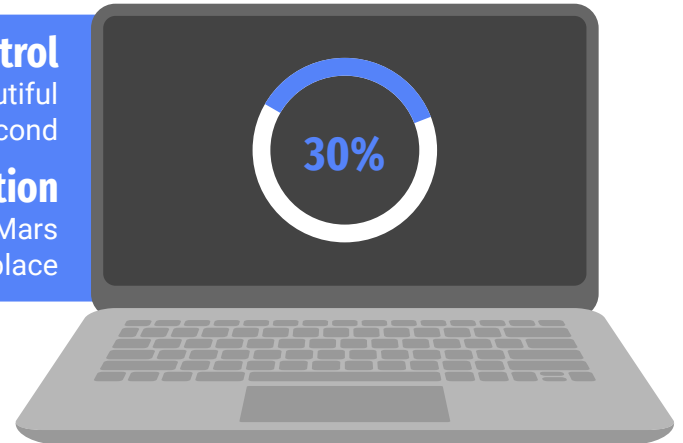
## Variation

Mercury is the closest planet to the Sun



**Control**  
Venus has a beautiful name and is the second

**Variation**  
Despite being red, Mars is a very cold place



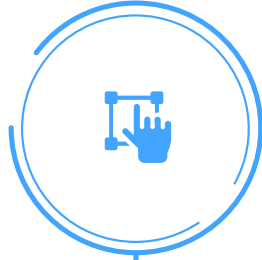
# A/B Testing Infographics



01

## Analyze data

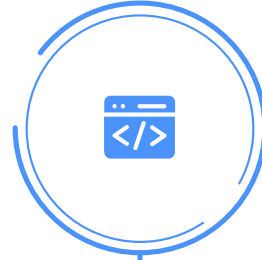
Mercury is the closest planet to the Sun and the smallest one of them all



02

## Form an hypothesis

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



03

## Experiment

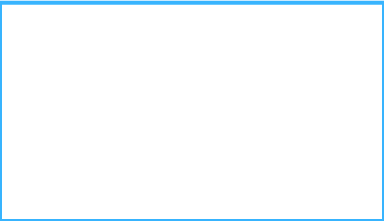
Despite being red, Mars is a very cold planet full of iron oxide dust



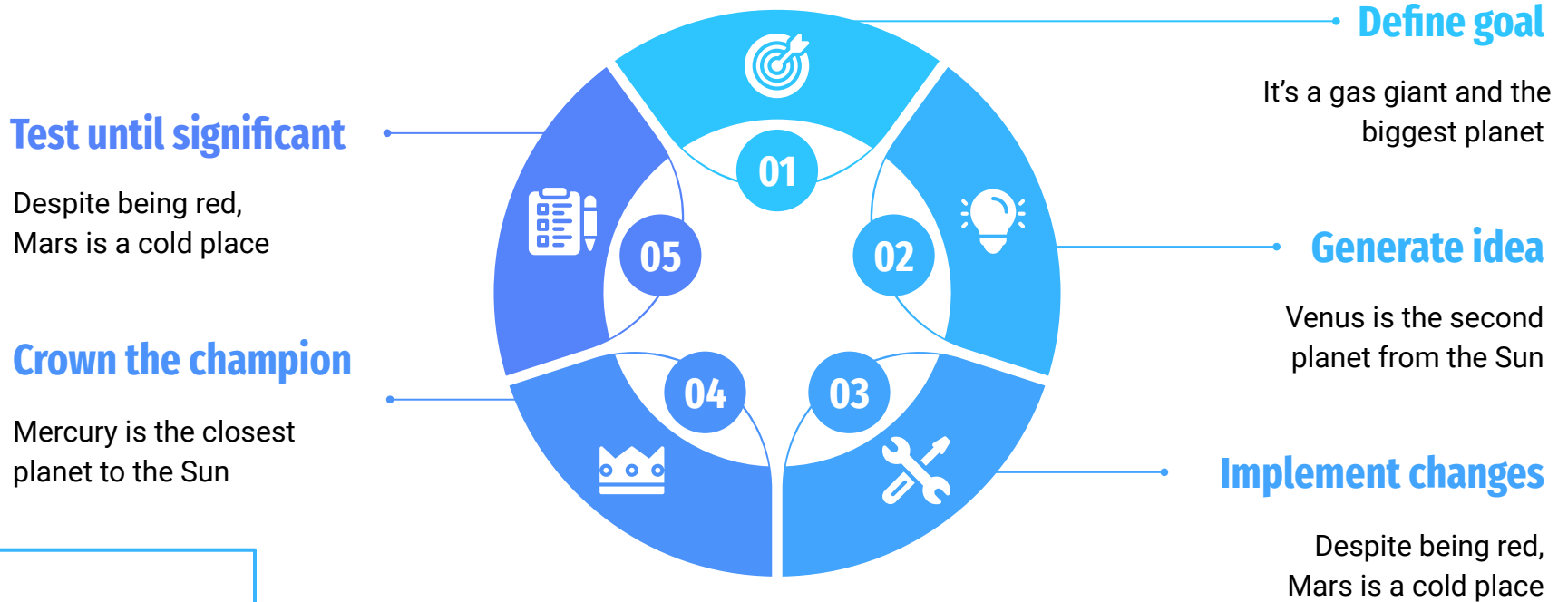
04

## Evaluate results

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



# A/B Testing Infographics

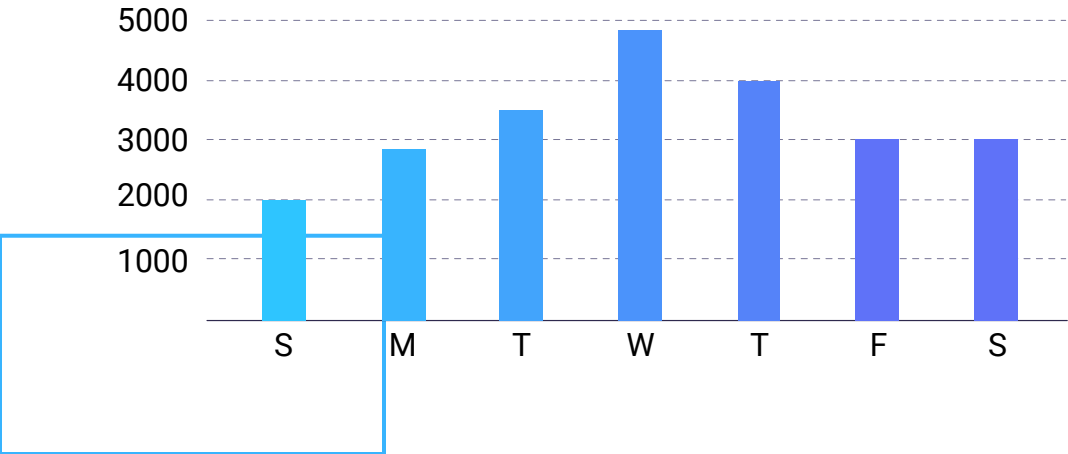




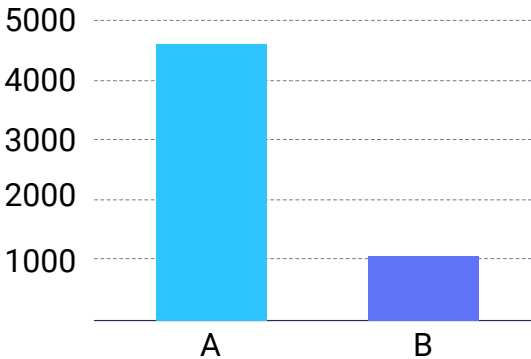
# A/B Testing Infographics

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%

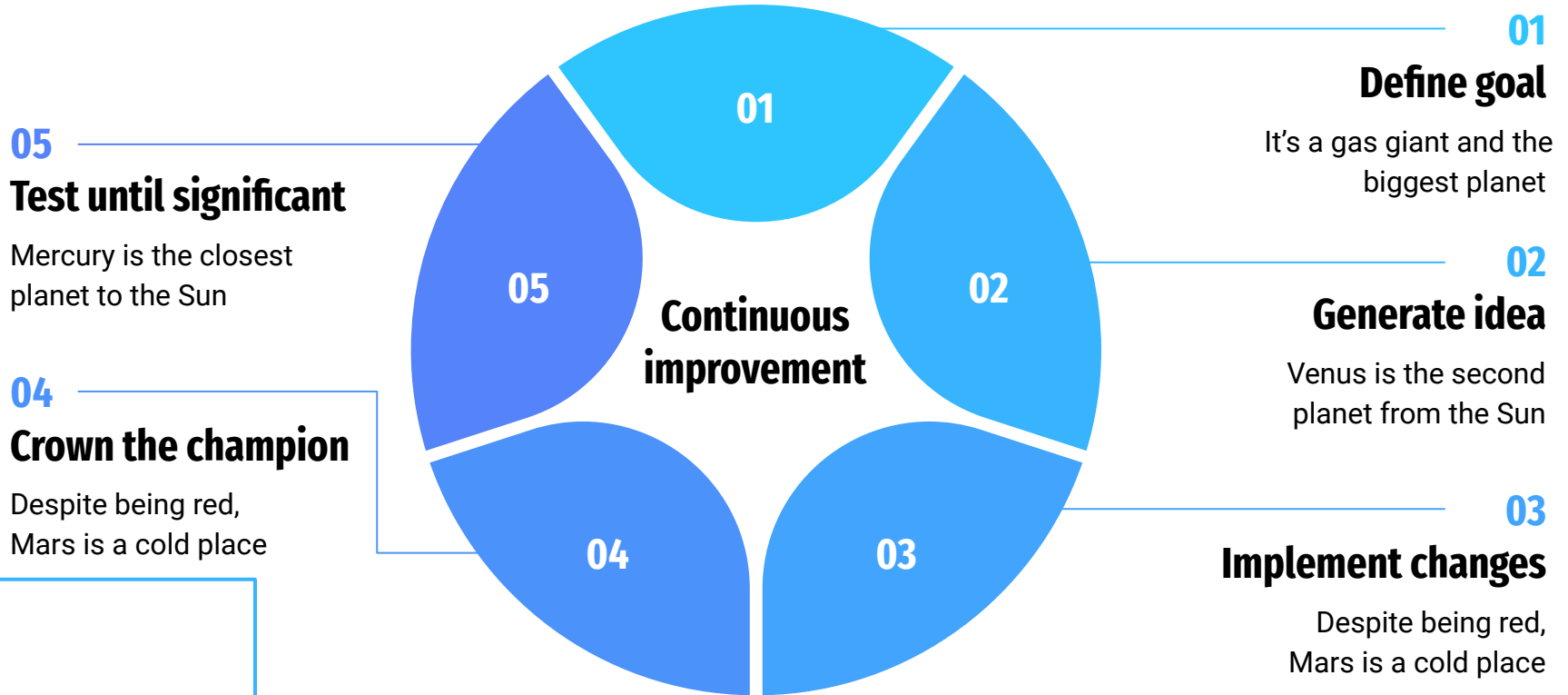
Sum of newsletter sign ups



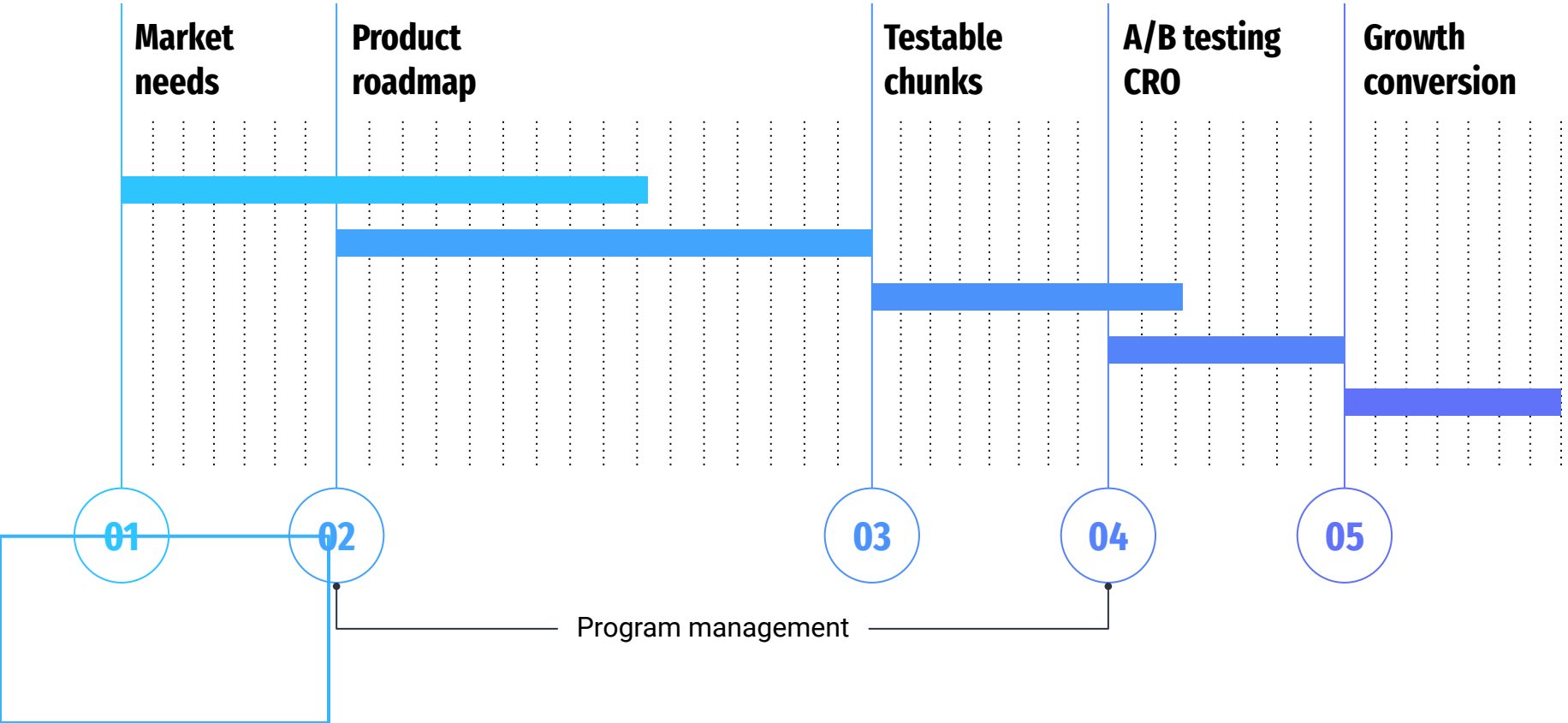
Chance of B outperforming A



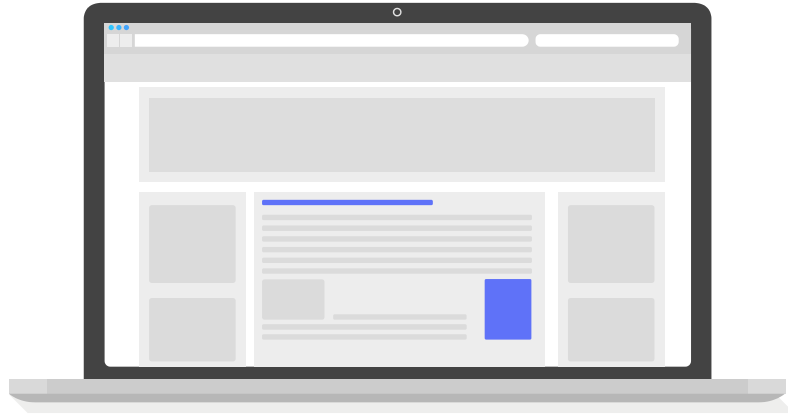
# A/B Testing Infographics



# A/B Testing Infographics



# A/B Testing Infographics

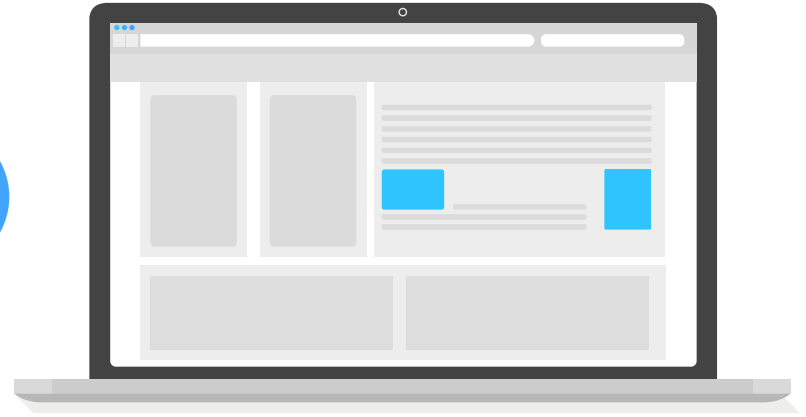


## Variation A



15% conversion

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



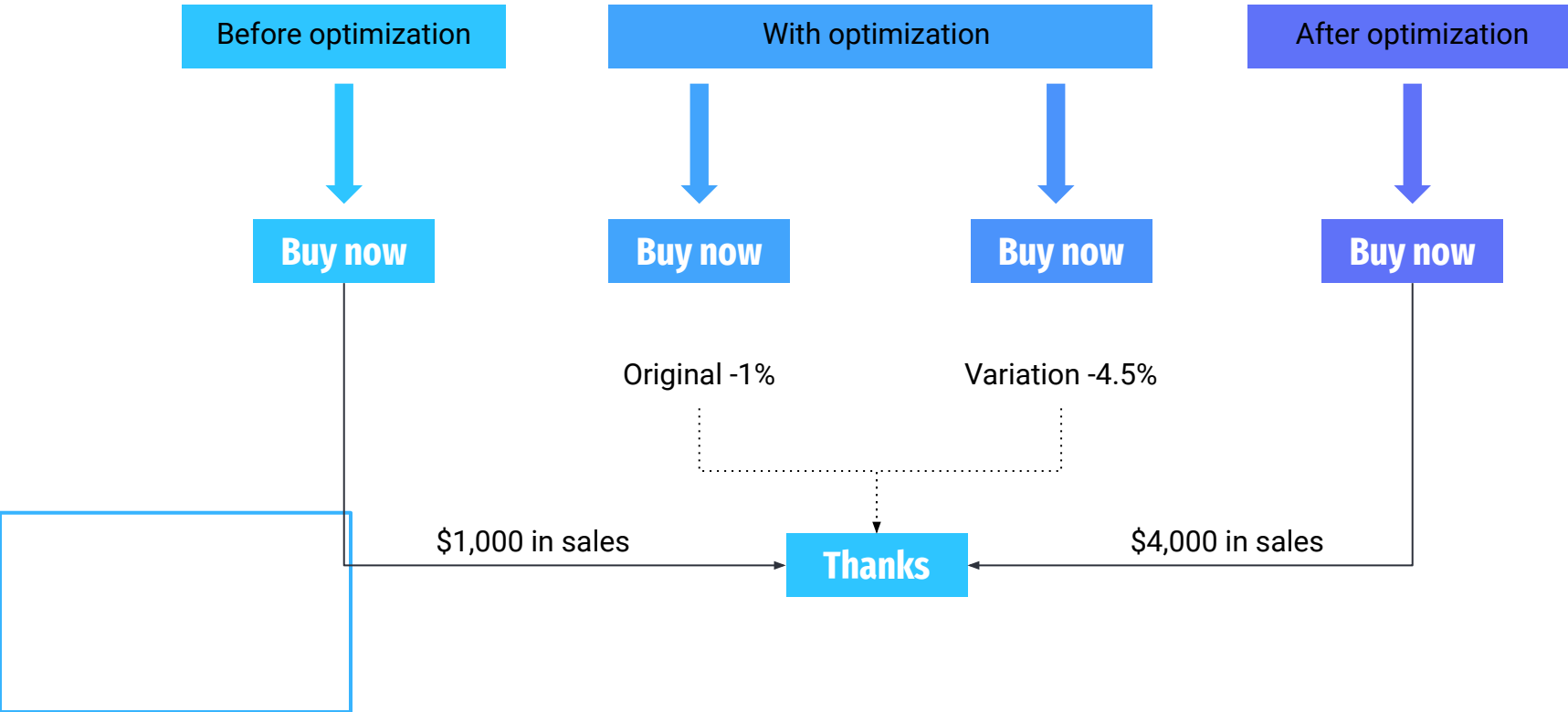
## Variation B



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



## Control

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

## Conversion

30%



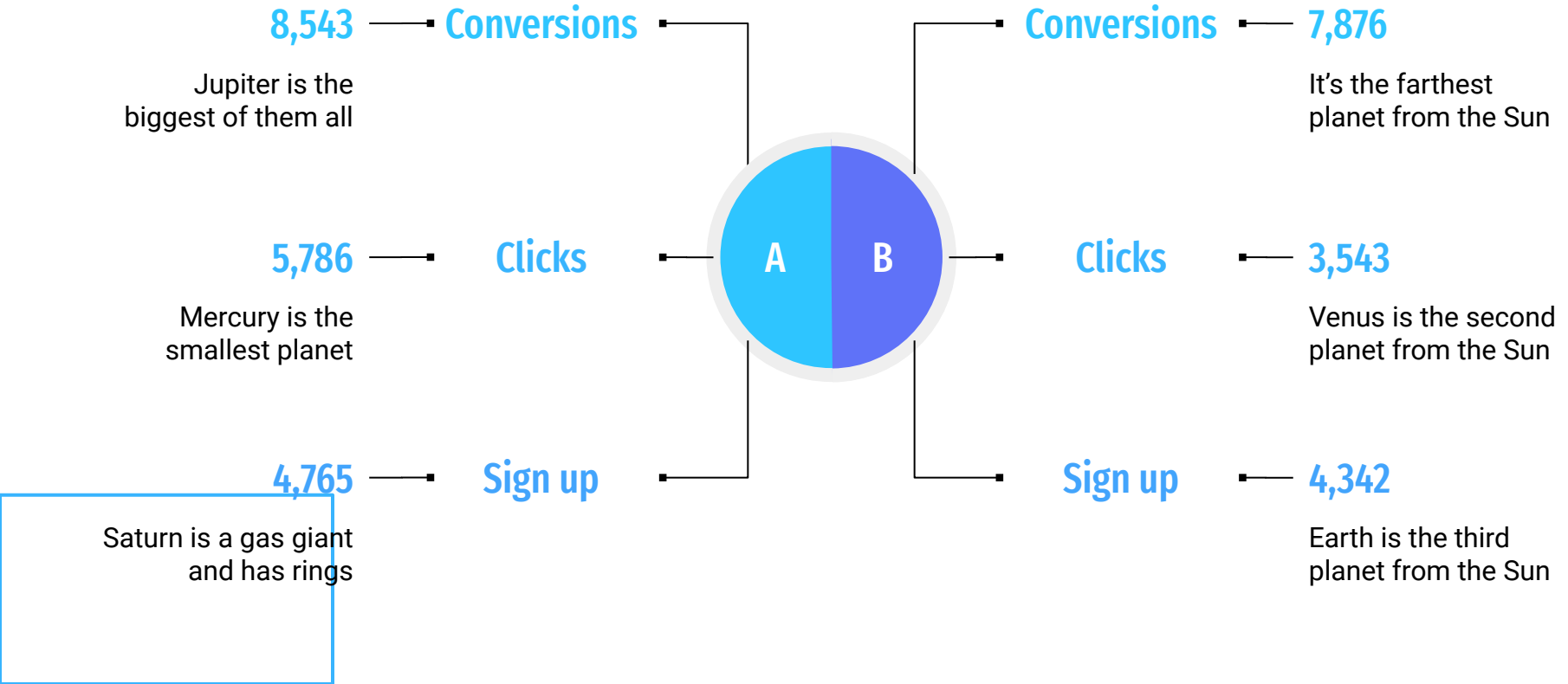
## Variation

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

## Conversion

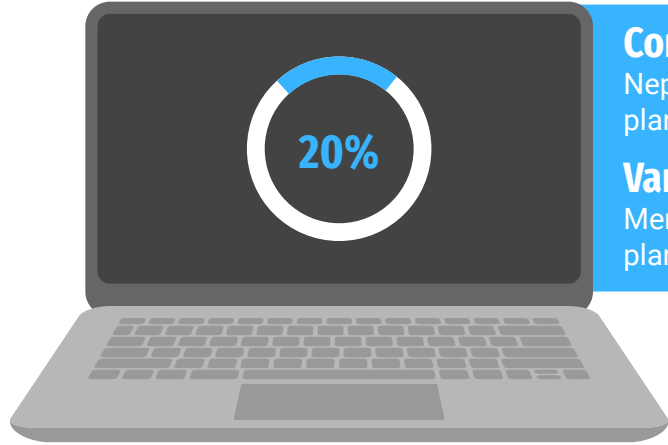
60%

# A/B Testing Infographics





# A/B Testing Infographics



## Control

Neptune is the farthest planet from the Sun

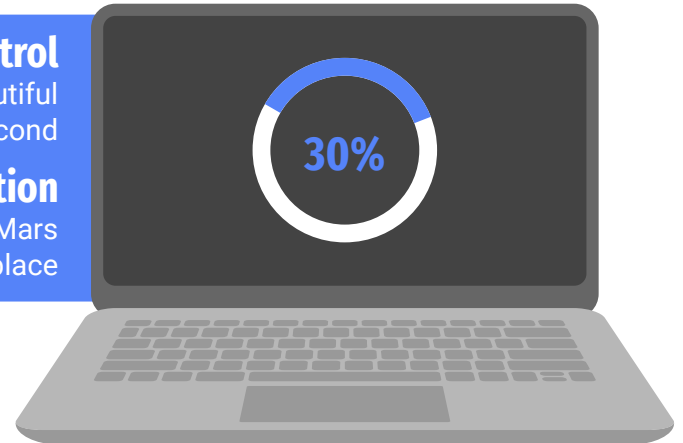
## Variation

Mercury is the closest planet to the Sun

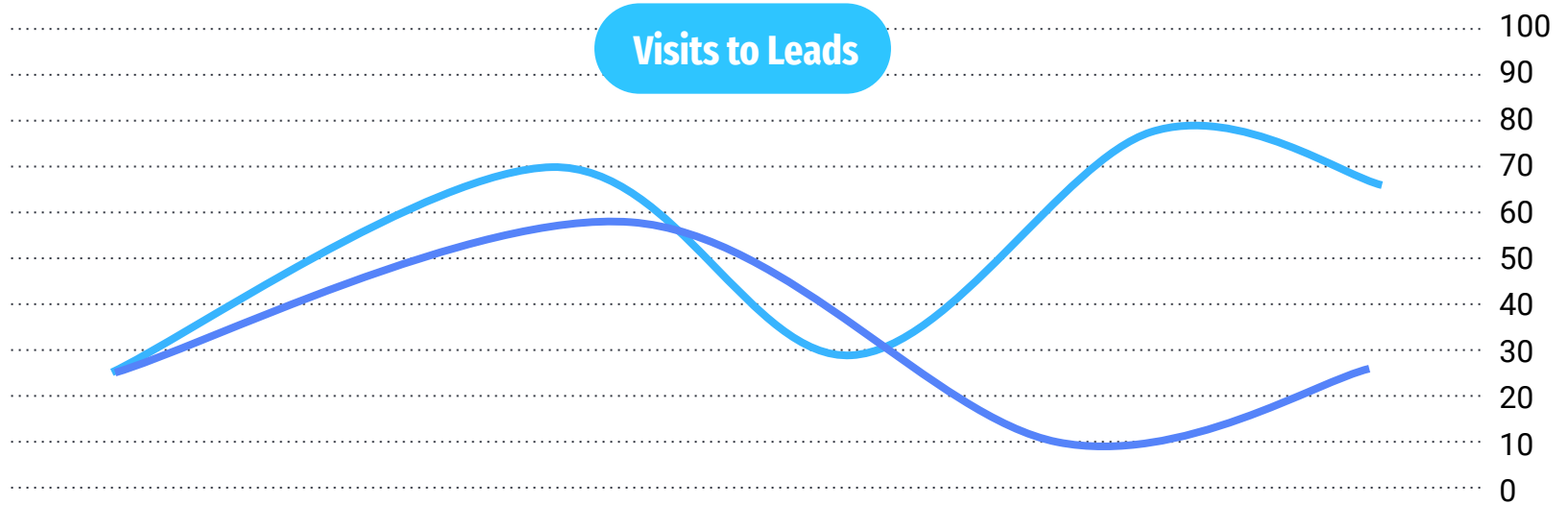


**Control**  
Venus has a beautiful name and is the second

**Variation**  
Despite being red, Mars is a very cold place



# A/B Testing Infographics



## Variation A

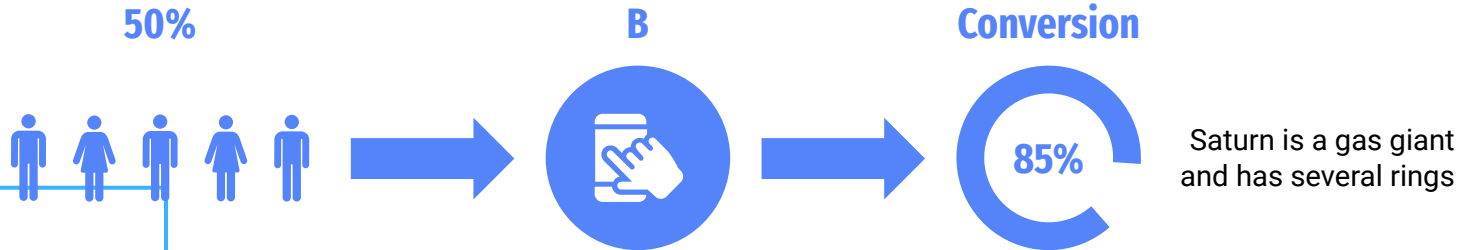
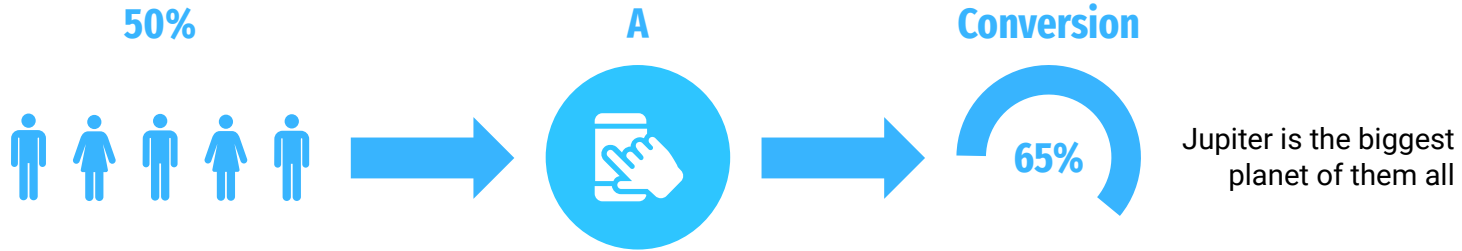
Mercury is the closest planet to the Sun and the smallest one

## Variation B

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



# A/B Testing Infographics



# A/B Testing Infographics

01

## Design base layout

Jupiter is a gas giant and also the biggest planet

03

## Define test plan

Despite being red, Mars is a very cold place

05

## Run with best option

Neptune is the farthest planet from the Sun



02

## Create variations

Mercury is the closest planet to the Sun

04

## Collect data

Venus is the second 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

04

## Evaluate results

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

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



# 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

## 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



## 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

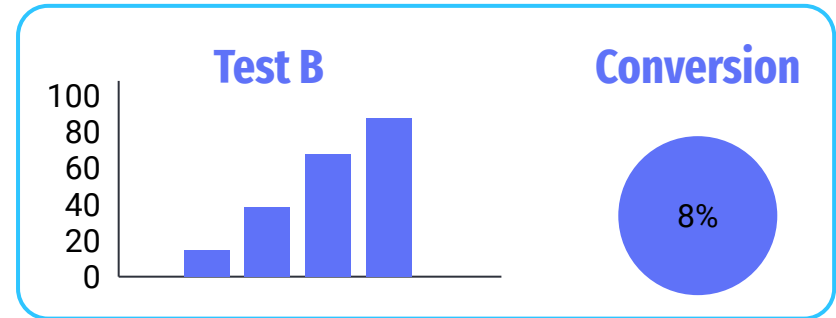
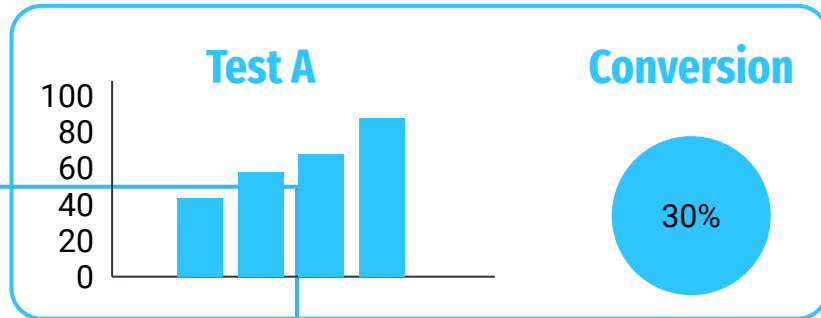
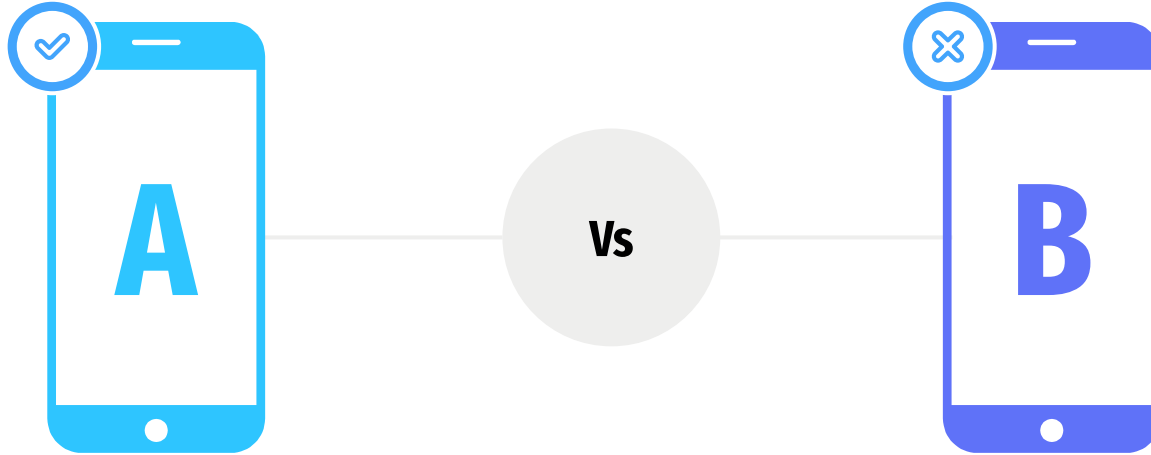
## Venus

Venus is the second planet from the Sun

## Saturn

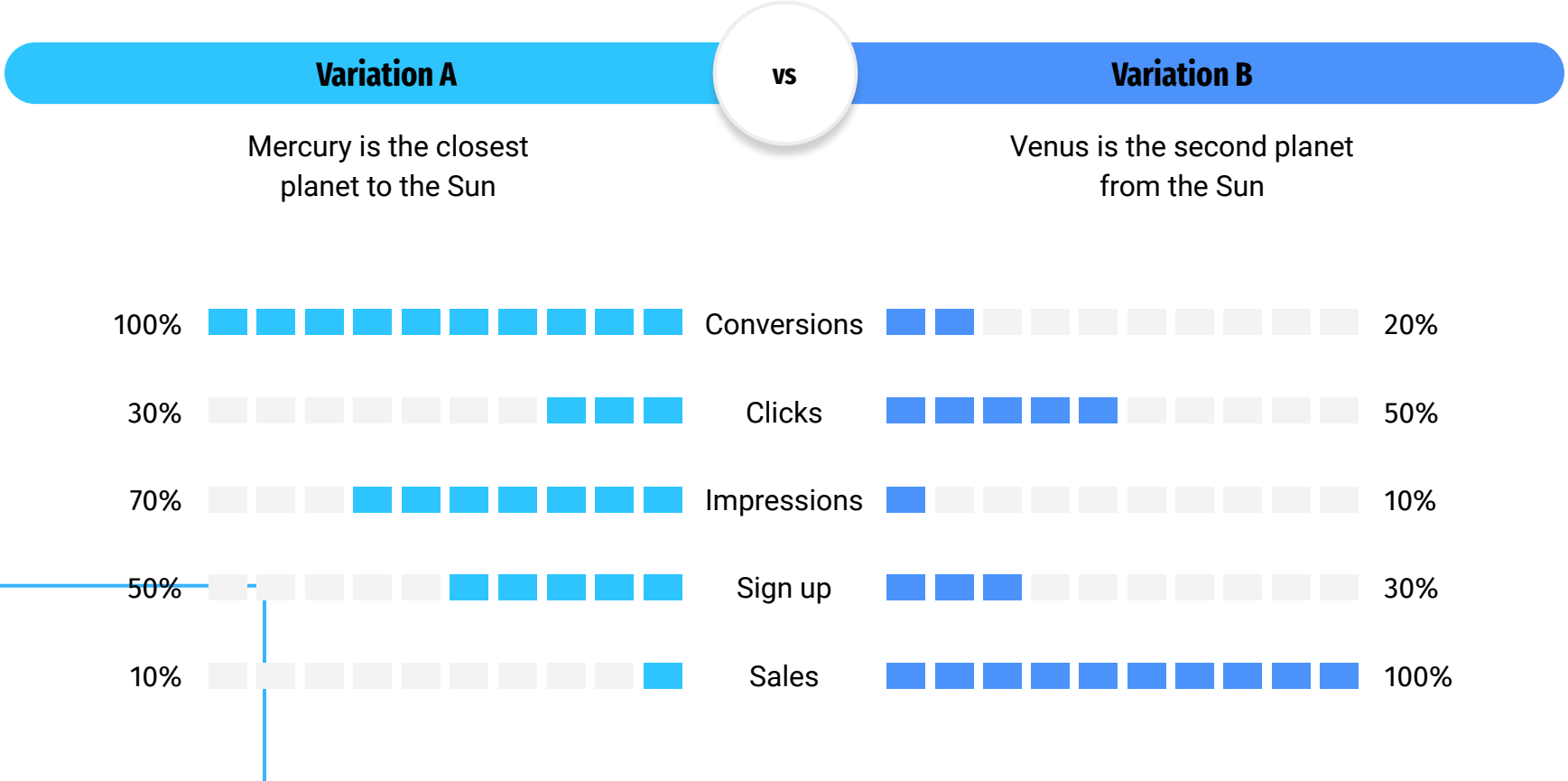
Saturn is the ringed one. It's a gas giant

# A/B Testing Infographics



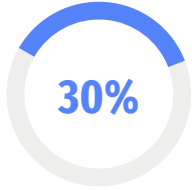


# A/B Testing Infographics



# A/B Testing Infographics

## Conversion

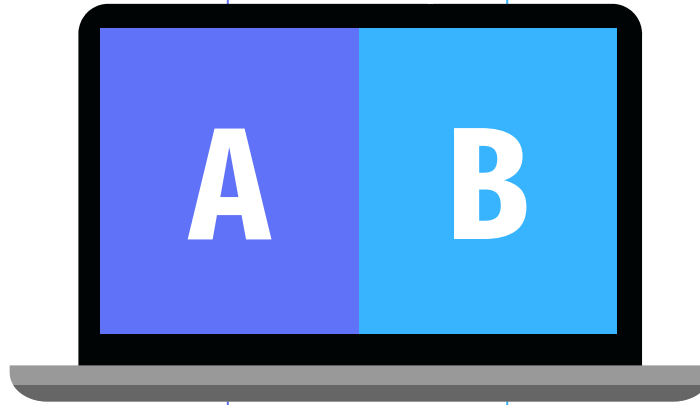


### Control

Venus is the second planet from the Sun

### Variation

Despite being red, Mars is a very cold place



## Conversion



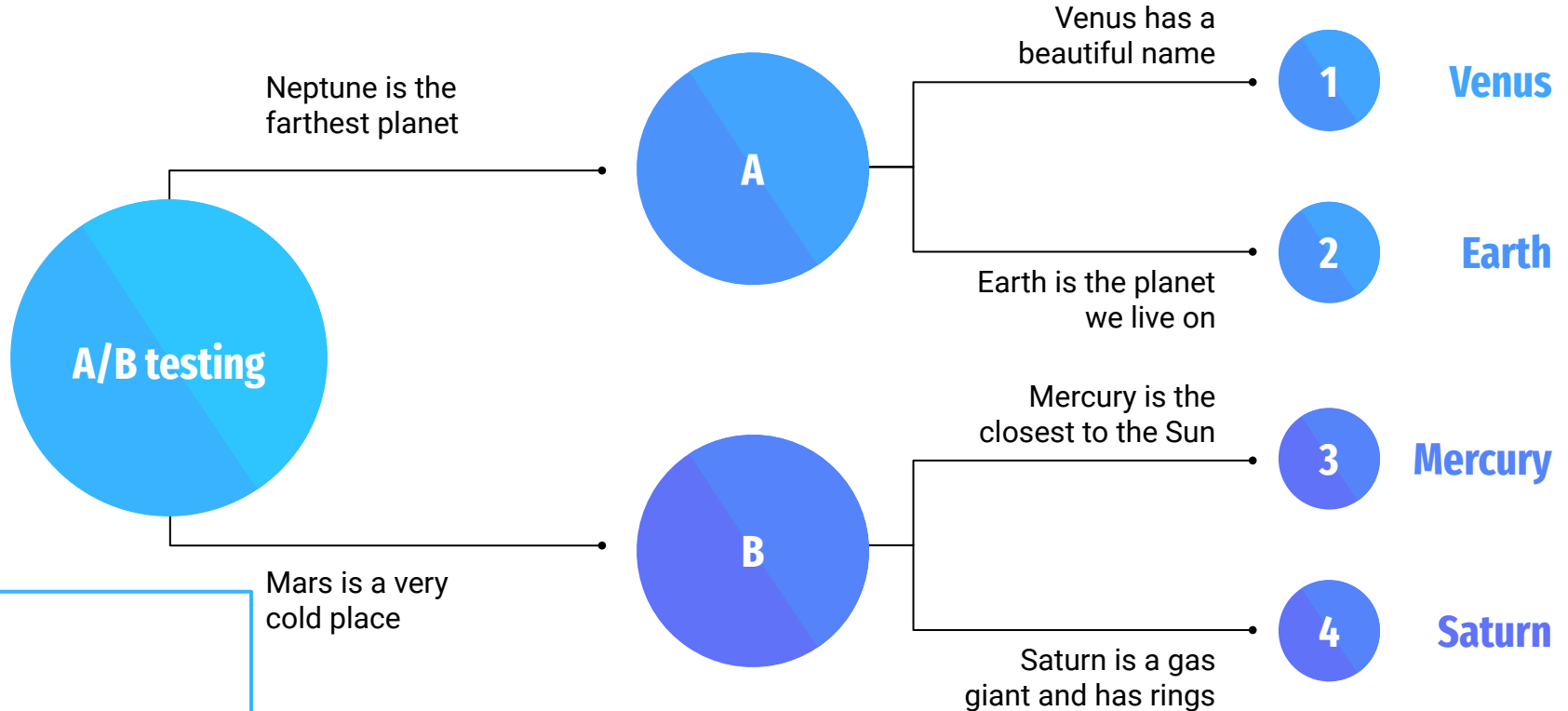
### Control

Neptune is the farthest planet from the Sun

### Variation

Mercury is the closest planet to the Sun

# A/B Testing Infographics



# 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

05

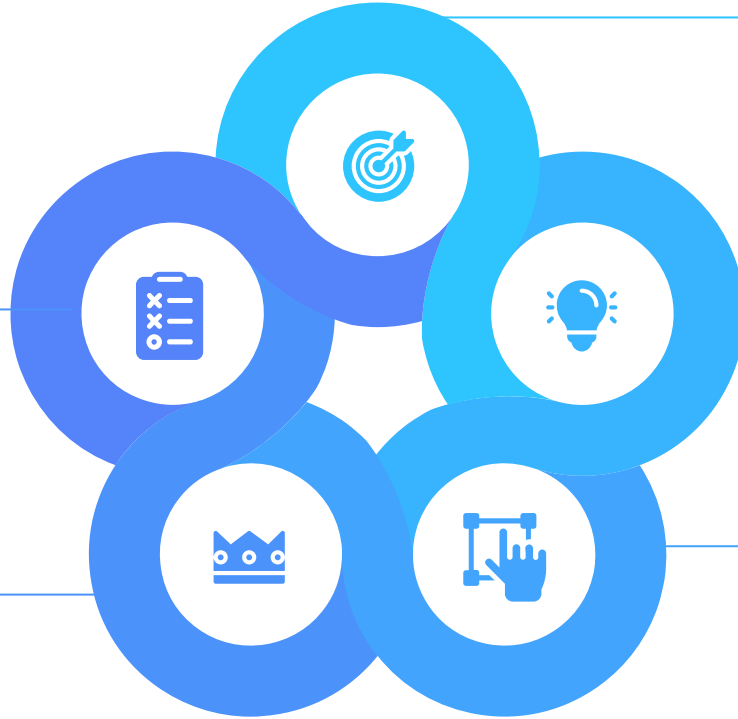
## Test until significant

Mercury is the closest planet to the Sun

04

## Crown the champion

Despite being red, Mars is a cold place



# 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

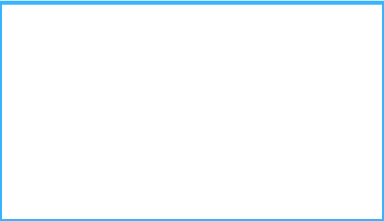
Mercury is the closest planet to the Sun



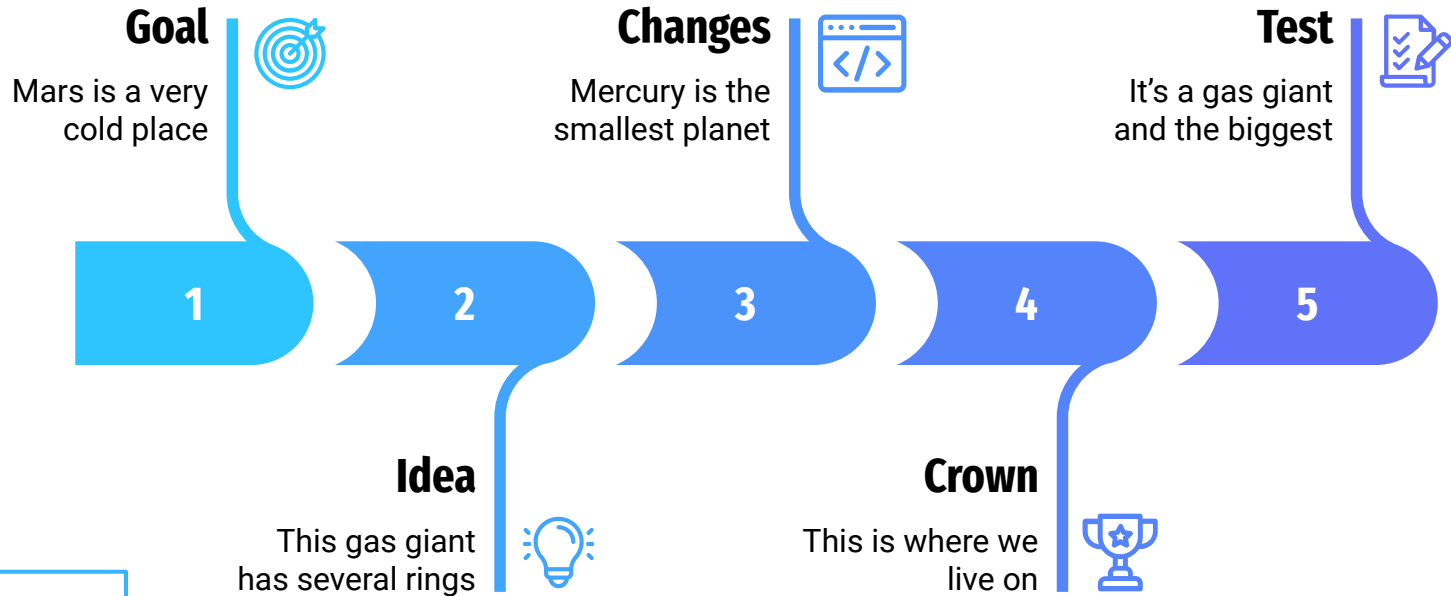
05

## Test until significant

Despite being red, Mars is a cold place



# A/B Testing Infographics



# A/B Testing Infographics



**Variation A**

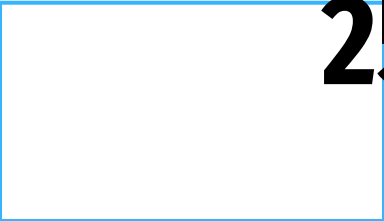
Mercury is the closest  
planet to the Sun and  
the smallest one

**75%**

**Variation B**



**25%**

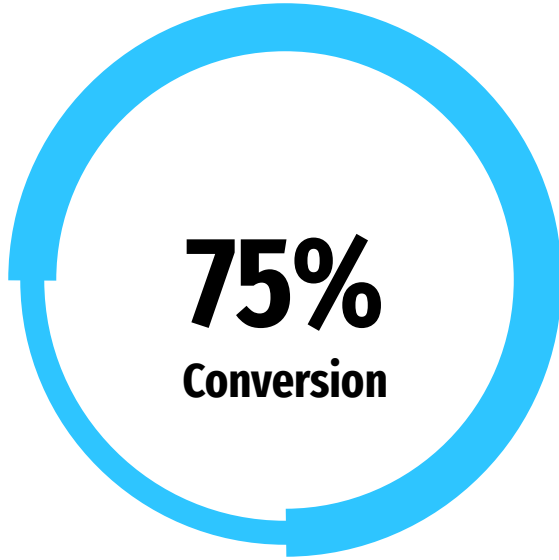


# A/B Testing Infographics

	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



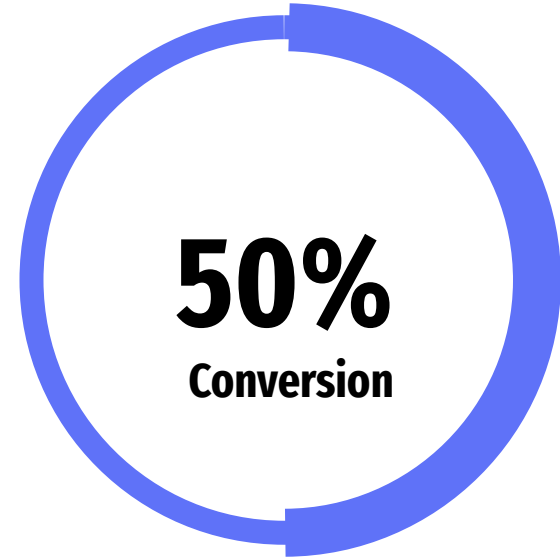
# A/B Testing Infographics



## Variation A

Despite being red,  
Mars is a cold place

**VS**



## Variation B

Venus is the second  
planet from the Sun

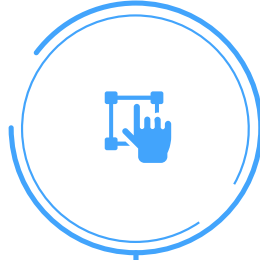
# A/B Testing Infographics



01

## Analyze data

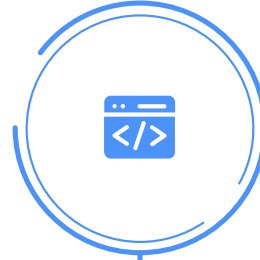
Mercury is the closest planet to the Sun and the smallest one of them all



02

## Form an hypothesis

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



03

## Experiment

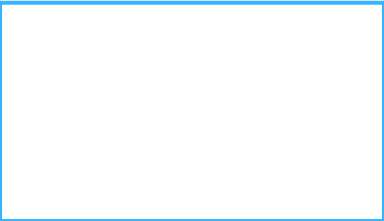
Despite being red, Mars is a very cold planet full of iron oxide dust



04

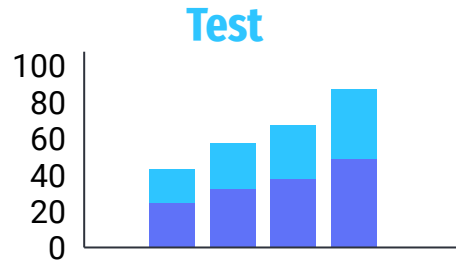
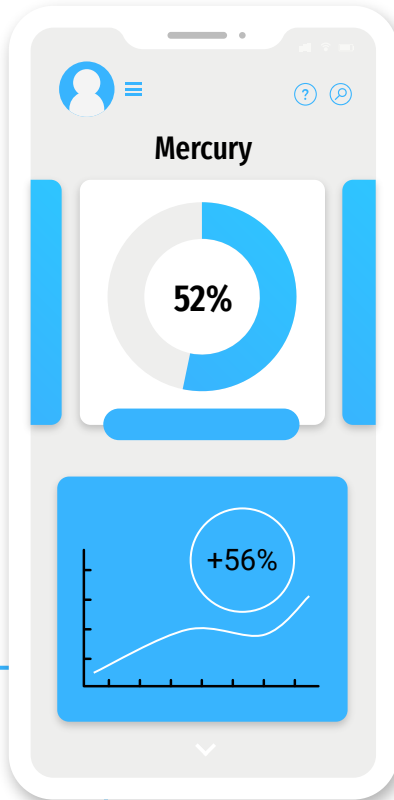
## Evaluate results

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

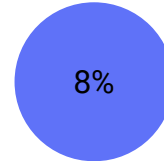
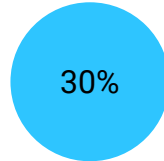


# A/B Testing Infographics

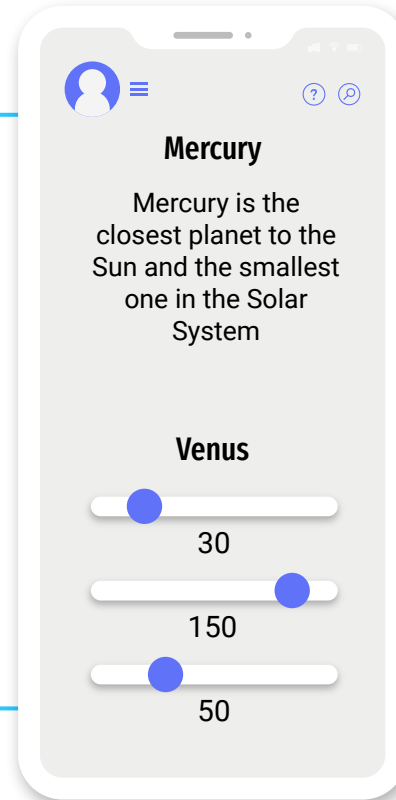
A



Conversion

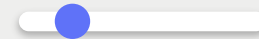


B

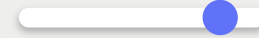


Mercury  
Mercury is the  
closest planet to the  
Sun and the smallest  
one in the Solar  
System

Venus



30

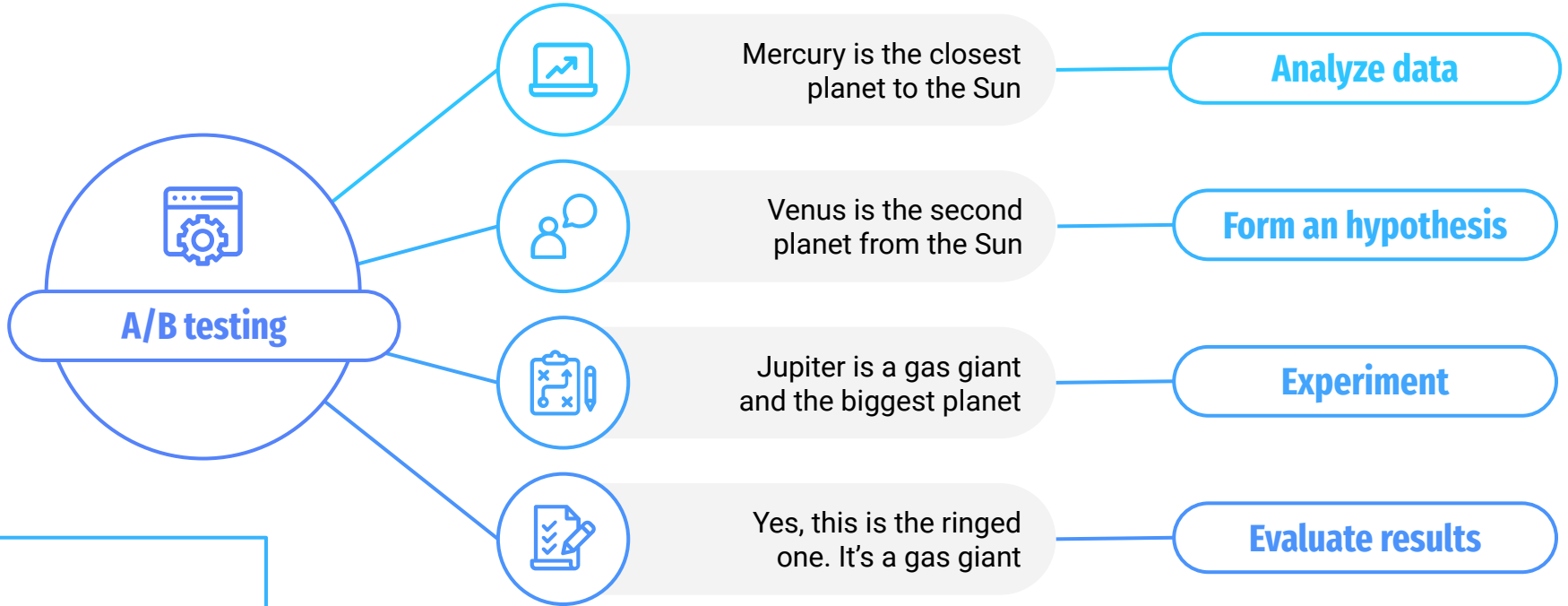


150



50

# A/B Testing Infographics



# 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



# Instructions for use (free users)

In order to use this template, you must credit [Slidesgo](#) by keeping the Thanks slide.

## You are allowed to:

- Modify this template.
- Use it for both personal and commercial purposes.

## You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Delete the “Thanks” or “Credits” slide.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>

# Instructions for use (premium users)

In order to use this template, you must be a Premium user on [Slidesgo](#).

## You are allowed to:

- Modify this template.
- Use it for both personal and commercial purposes.
- Hide or delete the “Thanks” slide and the mention to Slidesgo in the credits.
- Share this template in an editable format with people who are not part of your team.

## You are not allowed to:

- Sublicense, sell or rent this Slidesgo Template (or a modified version of this Slidesgo Template).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>

# 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**.

