

---

# Documentação de Projeto

para o sistema

## Covering

Versão 1.0

Projeto de sistema elaborado pelo alunos Arthur Rocha Amaral e Guilherme Oliveira Antônio e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo do professor José Laerte Xavier, orientação acadêmica do professor Marco Rodrigo Costa e orientação de TCC II do professor (a ser definido no próximo semestre).

27/03/2022

# Tabela de Conteúdo

<b>Tabela de Conteúdo</b>	<b>ii</b>
<b>Histórico de Revisões</b>	<b>ii</b>
<b>1. Introdução</b>	<b>1</b>
<b>2. Modelo de Requisitos</b>	<b>3</b>
2.1 Descrição de Atores	3
2.2 Modelo de Usuários	3
2.3 Modelo de Casos de Uso	7
2.3.1 Diagrama de caso de uso	7
2.3.2 História dos usuarios	8
<b>3. Modelo de Projeto</b>	<b>1</b>
3.1 Diagrama de Classes	1
3.2 Diagramas de Sequência	1
3.3 Diagramas de Comunicação	1
3.4 Arquitetura Lógica: Diagramas de Pacotes	1
3.5 Diagramas de Estados	1
3.6 Diagrama de Componentes	1
<b>4. Projeto de Interface com Usuário</b>	<b>10</b>
<b>5. Modelo de Dados</b>	<b>2</b>
<b>6. Modelo de Teste</b>	<b>2</b>

## Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Arthur Rocha Amaral	25/03/2022	Criação do documento	1.0
Guilherme Oliveira Antônio	26/03/2022	Atualização do sumário Adição da descrição de modelos de usuários e requisitos Adição da descrição de atores Adição de modelos de usuários	1.1
Arthur Rocha Amaral e Guilherme Oliveira Antônio	27/03/2022	Atualização dos atores Adição dos casos de uso Adição do diagrama de caso de uso Adição do projeto de interface de usuário	1.2

## 1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema Covering. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema. Tal especificação acompanha este documento. Anexo a este documento também se encontra o Glossário.


## 2. Modelos de Usuário e Requisitos

Os modelos de usuário e os requisitos do sistema são apresentados nesta seção. Esta apresentação é composta pelas seguintes divisões: descrição dos atores (Seção 2.1); descrição dos modelos de usuário (Seção 2.2) e descrição dos casos de uso e histórias do usuário (Seção 2.3).

### 2.1 Descrição de Atores

Os atores identificados nesta etapa representam pessoas desenvolvedoras de diferentes senioridades. Dessa forma, se entende que as pessoas que necessitam desta aplicação, são aquelas que trabalham diretamente com desenvolvimento de software e testes automatizados. Desse modo, os atores aqui apresentados englobam desenvolvedores juniores, plenos e seniores, os quais estão em posições diferentes no dia a dia de um projeto. Porém estes possuem frustrações semelhantes relacionadas a teste de software, que pode ser que representada por um interseção entre os atores. Um outro ator também apresentado é um gerente de projeto, o qual tem o papel de gerenciar o desenvolvimento de sistemas. Dessa forma, busca sempre dados que representam a confiabilidade do software e garantindo padrões de qualidade entre as entregas.

### 2.2 Modelos de Usuários

 <p>Fonte: freepik (<a href="https://br.freepik.com">https://br.freepik.com</a>). Imagem livre de direitos autorais.</p>	<p>Paula, desenvolvedora junior, 25 anos. Trabalha em uma empresa de tecnologia de médio porte (de 50 a 99 empregados). Desde a infância gosta de lógica e está se graduando em Engenharia de Software. É uma pessoa que gosta de testar seus códigos, mas ainda não domina fortemente as técnicas de teste.</p>
<p>Objetivos</p>	<p>Seu objetivo é se desenvolver na área de tecnologia, buscando conhecimento técnico e de gestão.</p>

Habilidades	Lógica, algoritmos, REST, Javascript (JS), Typescript (TS), Jest, inglês básico, Scrum.
Tarefas	Desenvolver funcionalidades, dar suporte ao sistema.
Frustrações	Ainda não domina fortemente as técnicas de teste e por isso tem dificuldade de avaliar se seus testes estão bons ou não.

Tabela 1. Primeira persona identificada. Criada para representar desenvolvedora junior.


 <p>Fonte: freepik (<a href="https://br.freepik.com">https://br.freepik.com</a>). Imagem livre de direitos autorais.</p>	<p>Julio, desenvolvedor pleno, 27 anos. Trabalha em uma empresa de marketing de pequeno porte (de 10 a 49 empregados), na qual desenvolve o sistema da empresa e em alguns casos fazendo sites para clientes. Sempre está bem humorado, por fazer o que gosta e por ter liberdade no seu emprego. É uma pessoa que gosta de desafios e almeja conseguir se desenvolver profissionalmente.</p>
Objetivos	Se desenvolver profissionalmente e ser reconhecido no que faz.
Habilidades	Lógica de programação, diversas linguagens de programação, metodologias ágeis, banco de dados, padrões para desenvolvimento e testes.
Tarefas	Desenvolvimento de sistemas, gerenciar equipe, e gerenciar projetos.
Frustrações	Tem dificuldade para revisar as funcionalidades de outros desenvolvedores e acha ineficiente o jeito que sua equipe analisa a cobertura de software do projeto.

Tabela 2. Segunda persona identificada. Criada para representar desenvolvedor pleno.



 <p>Fonte: freepik (<a href="https://br.freepik.com">https://br.freepik.com</a>). Imagem livre de direitos autorais.</p>	<p>Matheus, desenvolvedor sênior, 30 anos. Trabalha em uma empresa de tecnologia de grande porte (100 ou mais empregados). Busca atender e auxiliar os desenvolvedores de sua equipe e sempre traz novidades para a empresa que trabalha. É uma pessoa que gosta de ensinar e é extremamente comunicativo. Ele está produzindo um curso para o desenvolvimento dos outros funcionários.</p>
<p>Objetivos</p>	<p>Busca conhecimento técnico, de gestão e deseja produzir cursos para ajudar outras pessoas e para ser reconhecido.</p>
<p>Habilidades</p>	<p>Banco de dados, padrões de projetos, padrões de testes, docker, kubernetes, infraestrutura em nuvem, gestão de pessoas, gestão de requisitos.</p>
<p>Tarefas</p>	<p>Gerenciamento de pessoas e de projeto, desenvolvimento de sistemas.</p>
<p>Frustrações</p>	<p>Entrega bons códigos, mas tem dificuldade de manter o padrão dentro dos projetos. Apesar de ter mostrado ao time como deveria ser feito, ele não consegue fazer com que os demais integrantes façam, cada um, a sua parte.</p>

Tabela 3. Terceira persona identificada. Criada para representar desenvolvedor sênior.

 <p>Fonte: freepik (<a href="https://br.freepik.com">https://br.freepik.com</a>). Imagem livre de direitos autorais.</p>	<p>Fernanda é gerente de projeto e tem 30 anos. Trabalha em uma empresa de tecnologia de grande porte (100 ou mais empregados). Busca formas práticas de documentar a confiabilidade do projeto para apresentações. É uma pessoa extremamente comunicativa e gosta de investigar as melhores práticas para o desenvolvimento de projetos. Ela se importa bastante com qualidade de código pois sabe o quanto as entregas mal feitas e pouco testadas atrasam o projeto.</p>
<p>Objetivos</p>	<p>Crescer dentro da empresa, trazendo novidades e dando suporte aos times que ela integra. Ela</p>

	quer aumentar a produtividade dos times garantindo o mínimo de retrabalho possível nas equipes.
Habilidades	Gestão de pessoas, gerenciamento de requisitos, boa comunicadora, noção em tecnologias usadas nos projetos e métricas de processo.
Tarefas	Gerenciamento de projetos, suporte à equipe de desenvolvimento, tomada de decisões estratégicas de negócio.
Frustrações	Ela tem dificuldade de entender métricas técnicas de produto, mas entende a importância destas. Tem dificuldade em cruzar os dados de produtividade com de qualidade de código.

Tabela 4. Quarta persona identificada. Criada para representar gerente de tecnologia do projeto.

## 2.3 Modelo de Casos de Uso e Histórias de Usuários

### 2.3.1 Diagrama Casos de Uso

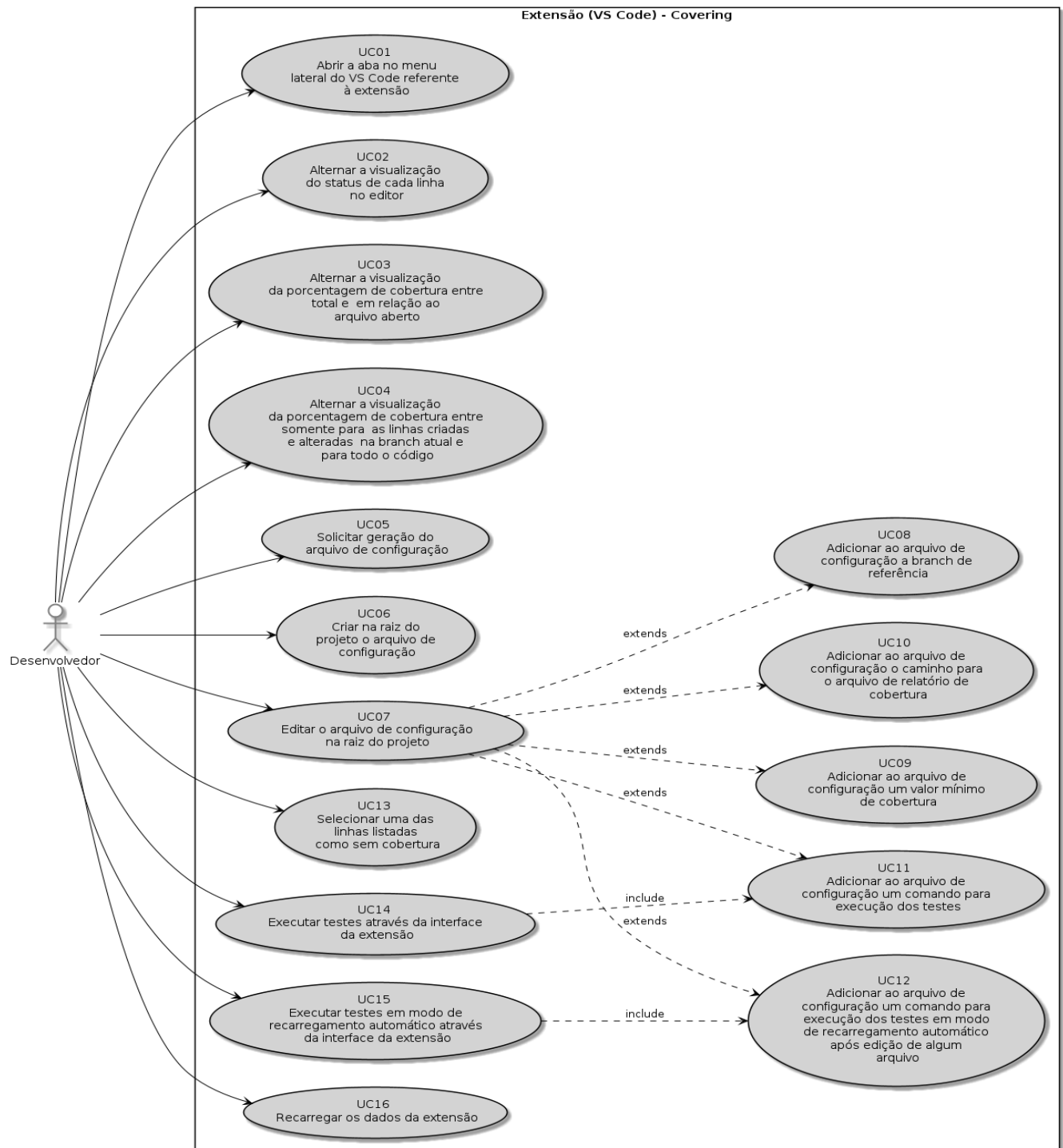


Figura 1. Diagrama de caso de uso.

### **2.3.2 Histórias dos usuarios**

UC01: Como desenvolvedor, desejo abrir a aba no menu lateral do VS Code, trazendo informações pertinentes à extensão.

UC02: Como desenvolvedor, desejo alternar a visualização do status de cada linha no editor.

UC03: Como desenvolvedor, desejo alterar a visualização da porcentagem de cobertura entre total e em relação ao arquivo aberto.

UC04: Como desenvolvedor, desejo alternar a visualização da porcentagem de cobertura entre somente para as linhas criadas e alteradas na branch atual e para todo o código.

UC05: Como desenvolvedor, desejo solicitar a geração do arquivo de configuração.

UC06: Como desenvolvedor, desejo criar na raiz do projeto o arquivo de configuração.

UC07: Como desenvolvedor, desejo editar o arquivo de configuração na raiz do projeto.

UC08: Como desenvolvedor, desejo adicionar ao arquivo de configuração a branch de referência.

UC09: Como desenvolvedor, desejo adicionar ao arquivo de configuração um valor mínimo de cobertura.

UC10: Como desenvolvedor, desejo adicionar ao arquivo de configuração o caminho para o arquivo de relatório de cobertura.

UC11: Como desenvolvedor, desejo adicionar ao arquivo de configuração um comando para execução dos testes.

UC12: Como desenvolvedor, desejo adicionar ao arquivo de configuração um comando para execução dos testes em modo de recarregamento automático após edição de algum arquivo.

UC13: Como desenvolvedor, desejo selecionar uma das linhas listadas como sem cobertura.

UC14: Como desenvolvedor, desejo executar testes através da interface da extensão.

UC15: Como desenvolvedor, desejo executar testes em modo de recarregamento automático através da interface da extensão.

UC16: Como gerente de projeto, desejo recarregar os dados da extensão.



## 2.4 Diagrama de Sequência do Sistema e Contrato de Operações

Nesta subseção é apresentado o diagrama de sequência do sistema e os Contratos de Operações.

Formato para cada contrato de operação

<b>Contrato</b>	
<b>Operação</b>	
<b>Referências cruzadas</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	

## 3. Modelos de Projeto

### 3.1 Diagrama de Classes

Diagrama de classes do sistema

### 3.2 Diagramas de Sequência

Diagramas de sequência para realização de casos de uso.

### 3.3 Diagramas de Comunicação

Diagramas de comunicação para realização de casos de uso.

### 3.4 Arquitetura

Pode ser descrita com um diagrama apropriado da UML ou C4 Model

### 3.5 Diagramas de Estados

Diagramas de estados do sistema.

### **3.6 Diagrama de Componentes e Implantação.**

Diagramas de componentes do sistema. Diagrama de implantação mostrando onde os componentes estarão alocados para a execução.

## **4. Projeto de Interface com Usuário**

Nesta seção são apresentados os protótipos das interfaces do sistema. Para criação destas interfaces foi utilizado o software Figma, o qual disponibiliza a criação de interfaces com base em componentes prontos, feitos por instituições, como neste caso, ou pela própria comunidade. Esse projeto foi utilizado o modelo “Visual Studio Code Toolkit” feito pela Microsoft. As estruturas utilizadas têm como base os componentes do Visual Studio Code (VS Code) e estas foram utilizadas na criação das interfaces. Porém, os componentes que não são nativos dos VS Code, foram criados exclusivamente para o projeto apresentado neste documento pelos próprios autores.

A primeira tela modelada é representada na Figura 2, onde pode ser observado o destaque dos status de cobertura de cada linha no editor de texto. Para isso cada cor em destaque representa um status da linha como:

- verde, para as linhas totalmente cobertas;
- vermelho, para as linhas que não foram cobertas;
- amarelo, para as linhas que foram parcialmente cobertas;
- não destacada, para linhas que não são levadas em consideração no cálculo de cobertura.

Ainda na Figura 2 é possível notar que na barra lateral esquerda, chamada de “Barra de Atividades”, possui seis itens. Sendo o sexto item relacionado à extensão Covering. Também pode se destacar nessa tela, a visualização da porcentagem de cobertura do arquivo aberto no editor na barra chamada “Barra de Status”. Esta está localizada na parte inferior da interface, em azul, apresentando o valor 93%.

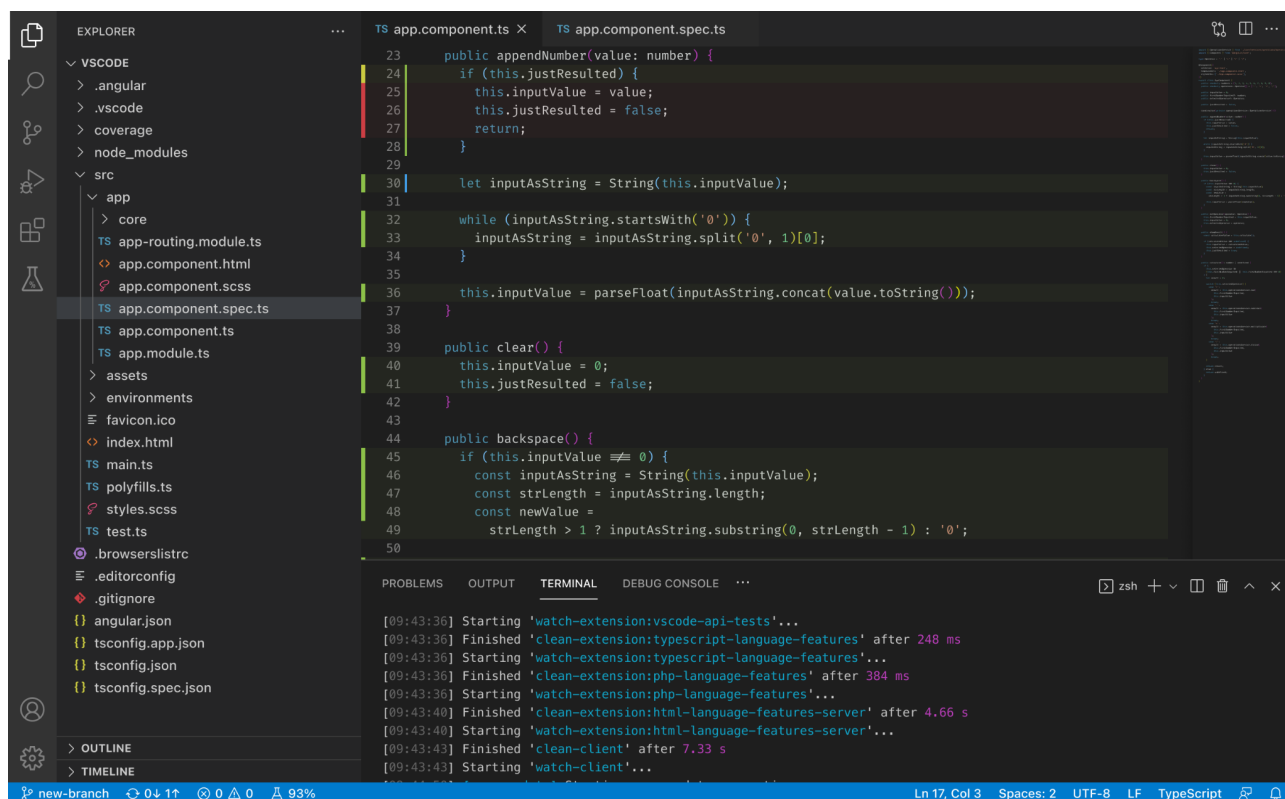


Figura 2. Destaque das linhas no editor.

Fonte: [\(AMARAL e ANTÔNIO, 2022\)](#)

Ao selecionar o item referente à extensão Covering na “Barra de Atividades”, o sistema passa a exibir os dados de cobertura na lateral esquerda do VS Code, região essa chamada de “Barra Lateral”. Assim como na Figura 1, onde o conteúdo da “Barra Lateral” exibe o explorador de arquivos. Já na Figura 2, é exibido os dados gerados pela extensão, possibilitando também a interação do usuário com as configurações e dados mostrados.

Esse conteúdo é dividido em três partes principais. A primeira é referente às configurações da extensão, sendo elas:

- a alteração da visualização do status de cada linha no editor;
- a alteração da visualização da porcentagem de cobertura para somente para as linhas criadas e alteradas na branch atual;
- a alteração do modo de visualização da porcentagem de cobertura.

A segunda parte é referente à meta estabelecida e o status atual de cobertura de código. A terceira exibe as linhas de código que permanecem sem cobertura, possibilitando a interação com o navegador de arquivos do VS Code.

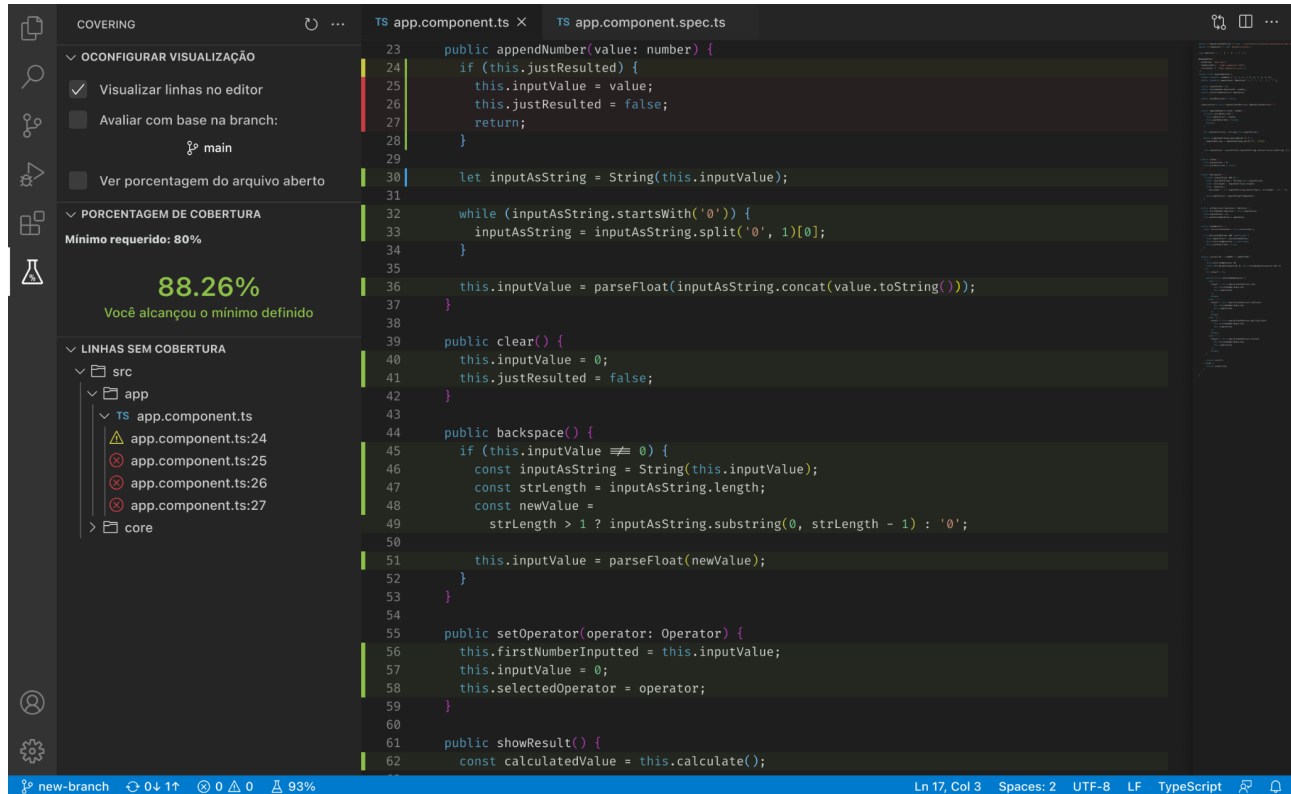


Figura 3. Tela com conteúdo da extensão na barra lateral.

Fonte: [\(AMARAL e ANTÔNIO, 2022\)](#)

As Figuras 4 e 5, exibem variações da Figura 3. Na Figura 4 pode se notar que a opção de “Ver porcentagem do arquivo aberto” está selecionada e isso altera como os dados são exibidos. Para essa opção, só é levado em consideração os dados do arquivo “*app.component.ts*”, exibindo a porcentagem de cobertura deste arquivo e suas linhas que foram cobertas por testes. Já na figura 5, a opção selecionada é a “Avaliar com base na branch: *main*”. Sendo assim, todos os dados exibidos na aba lateral, são referentes às linhas criadas e editadas até o momento.

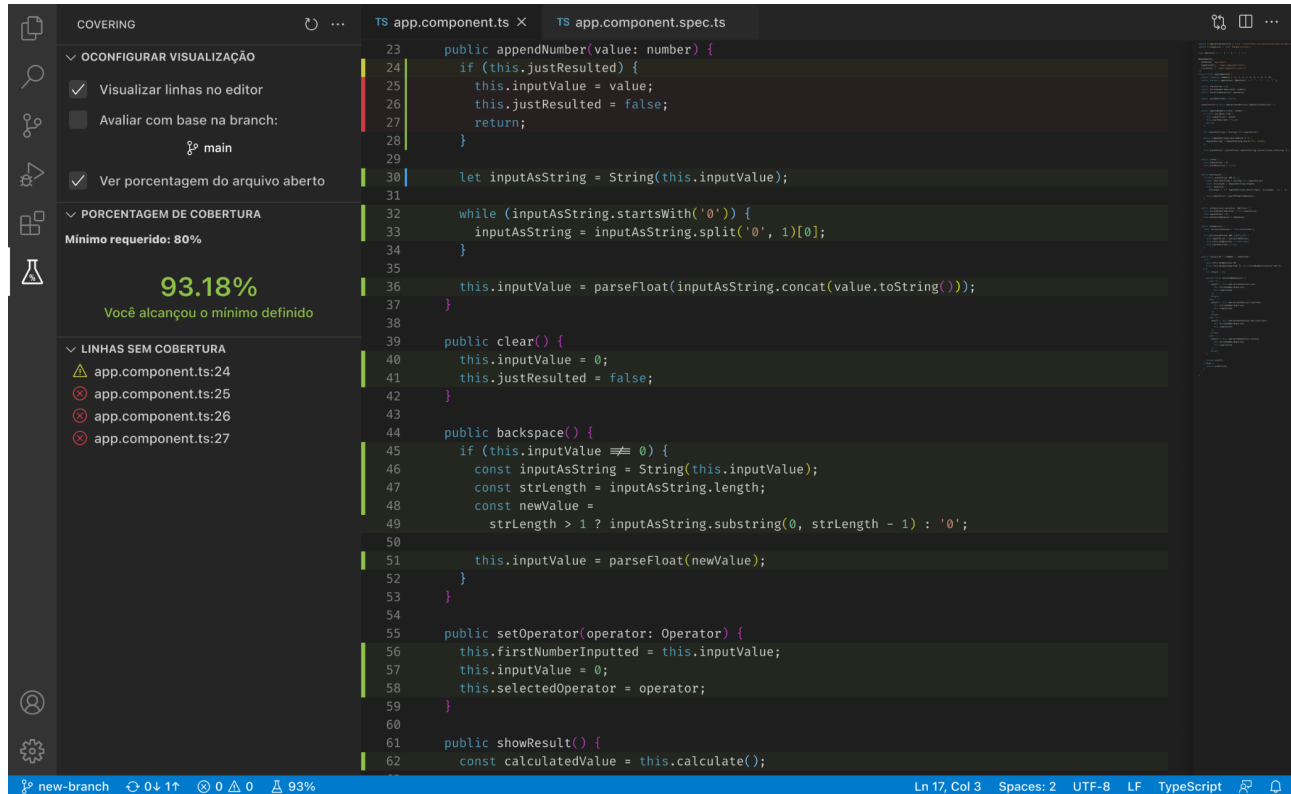


Figura 4. Tela com conteúdo da extensão na barra lateral somente do arquivo atual.

Fonte: (AMARAL e ANTÔNIO, 2022)

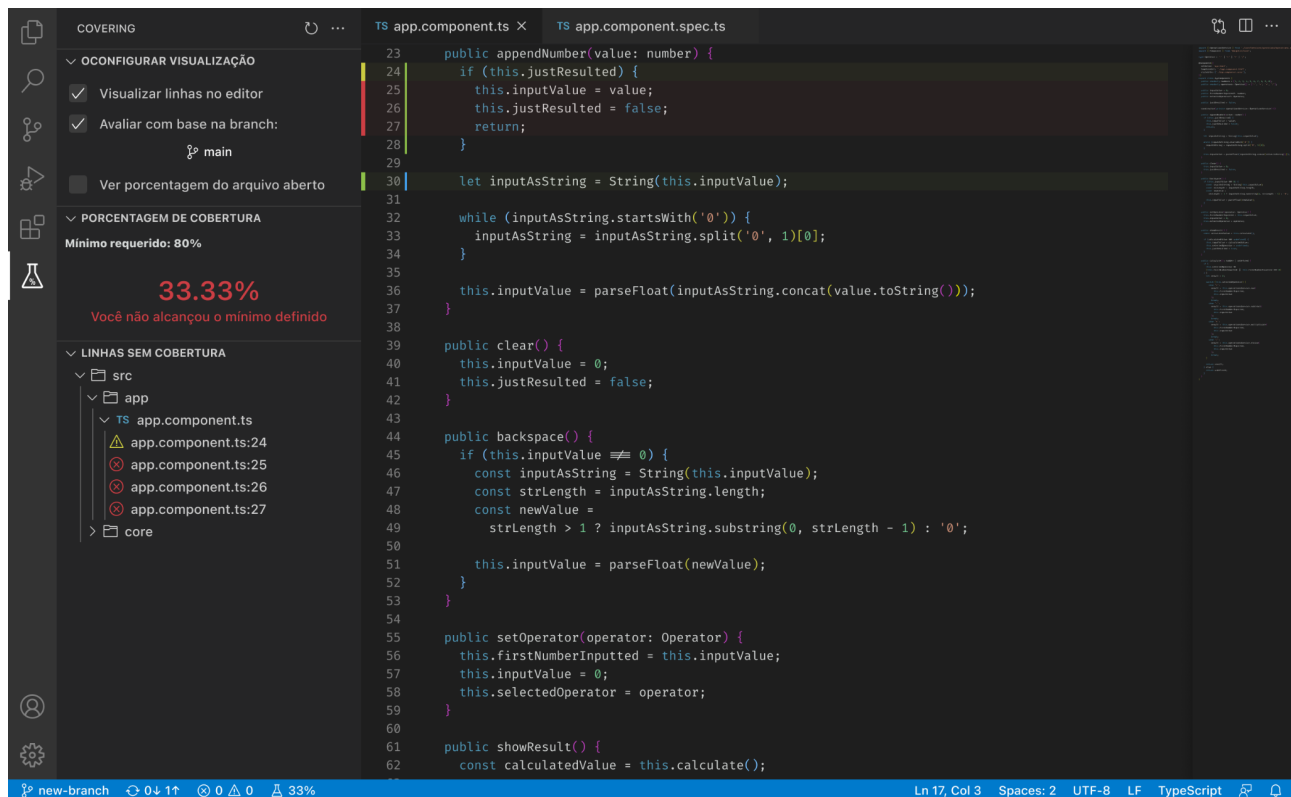


Figura 5. Tela com conteúdo da extensão na barra lateral com base na branch de referência.

Fonte: (AMARAL e ANTÔNIO, 2022)

Todas essas variações mostradas, baseiam-se em configurações padrões definidas pela própria extensão. Mas a principal funcionalidade proposta por essa aplicação, é a definição de padrões que podem ser estabelecidos para cada repositório, assim como é mostrado na Figura 6. Nessa imagem, pode ser visto as principais variáveis definidas para o repositório. Por tanto, o conteúdo exibido na “Barra Lateral”, varia de acordo com os parâmetros definidos no arquivo. Dentre essas alterações no conteúdo exibido, pode se destacar o valor mínimo de cobertura de teste, que agora aprova o resultado, diferentemente da Figura 5 que havia sido negado. Outro ponto que pode ser destacado é a possibilidade de executar os testes via interface da extensão. Essa função se habilita caso haja no arquivo de configuração um dos dois parâmetros para execução dos testes. Essa função pode ser localizado no cabeçalho da “Barra Lateral” com o ícone de “play”.

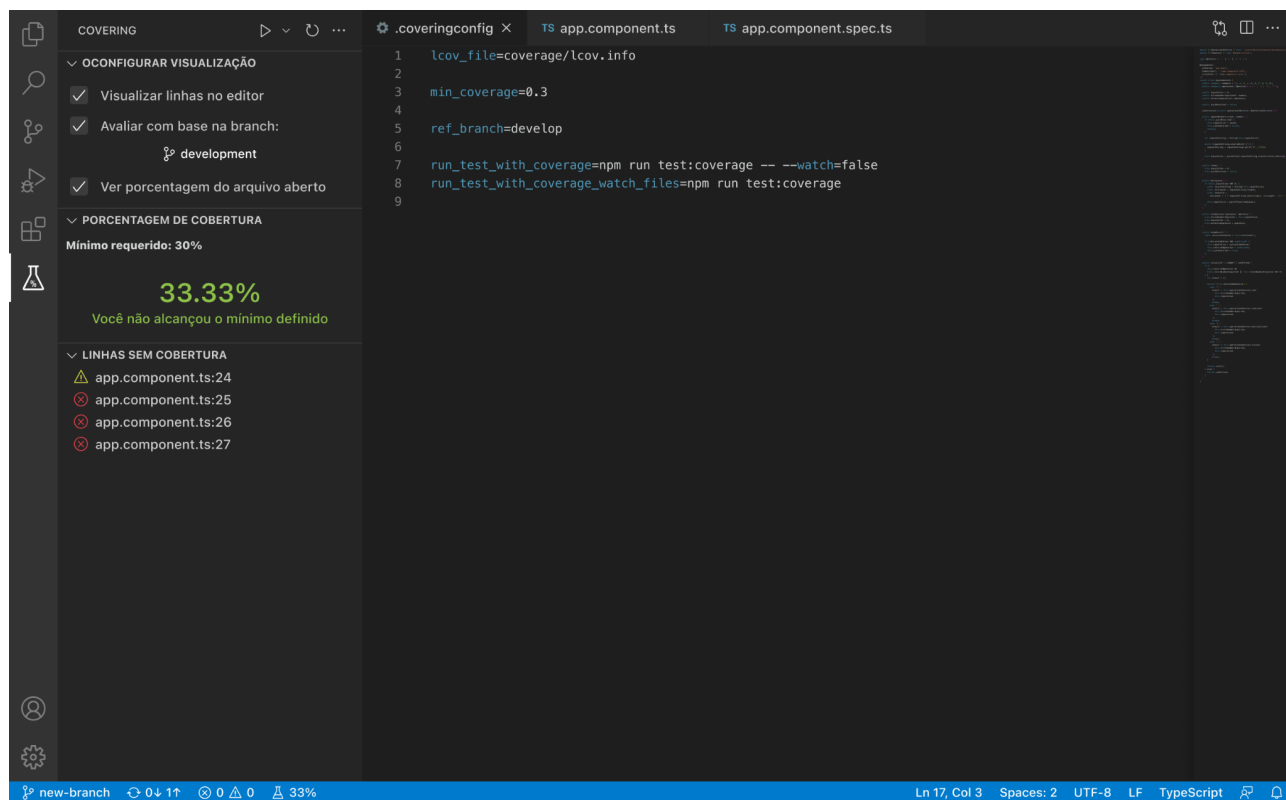


Figura 6. Tela com modelo do arquivo de configuração.

Fonte: (AMARAL e ANTÔNIO, 2022)

## 5. Glossário e Modelos de Dados

Deve-se apresentar o glossário para o sistema. Também apresente esquemas de banco de dados e as estratégias de mapeamento entre as representações de objetos e não-objetos.

## **6. Casos de Teste**

Uma descrição de casos de teste para validação do sistema.

## **7. Cronograma e Processo de Implementação**

Uma descrição do cronograma para implementação do sistema e do processo que será seguido durante a implementação.