PLANO DE TESTE PARA <<NOME DO PROJETO>>

Registro de Mudanças

Versão	Data de Mudança	Por	Descrição
1.1	19/06/2023	Marcos Roberto	Plano de Testes

1 Introdução

Utilizarei a classe gutterIconManager.ts do projeto BracketPair, que é uma extensão do VSCode que melhora a legibilidade do código, destacando parênteses e colchetes e chaves correspondentes. A classe específica é responsável por gerenciar ícones exibidos na área vertical ao lado do código.

1.1 Escopo

1.1.1 No escopo

O teste deve abranger todas as funcionalidades e métodos da classe GutterIconManager, como Dispose(), GetIconUri(), createIcon(), entre outros. Os testes devem verificar se os recursos estão implementados corretamente e se fornecem a funcionalidade esperada.

1.1.2 Fora do escopo

Não será testado nesta classe testes relacionados a segurança e utilização dele e seu desempenho em larga escala.

1.2 Objetivos de Qualidade

Em resumo, o objetivo é fornecer confiança na qualidade e funcionalidade do GutterIconManager, sendo capaz de realizar as seguintes tarefas:

- Verificar a criação de ícone para bracket ou cor inválidos
- Liberação adequada de recursos ao chamar o método dispose
- Reutilização correta de ícones

Assim, os testes ajudariam a garantir confiabilidade no código e eficiência, estando em conformidade com as expectativas de uso e funcionalidade.

1.3 Papéis e Responsabilidades

Descrição detalhada dos papéis e responsabilidades de diferentes membros da equipe como

- Analista de QA
- Gerente de Teste
- Gerente de configuração
- Desenvolvedores
- Equipe de Instalação

Entre outros

2 Metodologia de Teste

2.1 Visão Geral

- A utilização do método ágil nos testes promove agilidade, adaptabilidade e colaboração, sendo possível garantir uma melhor eficácia na busca pela melhor qualidade do código e atender aos objetivos do projeto.

2.2 Fases de Teste

2.3 Triagem de Erros

2.4 Critérios de Suspensão e Requisitos de Retomada

2.5 Completude do Teste

2.6 Atividades do projeto, estimativas e cronograma

Será testado dentro da classe GutterIconManager.ts
- Teste de liberação de recursos: Confirmar se a classe GutterIconManager libera corretamente os recursos quando chamado o método Dispose().

Reutilização correta de ícones: Garantir que a classe GutterIconManager reutilize ícones de URI existentes quando solicitado o mesmo bracket e cor, em vez de criar novos ícones desnecessariamente. Os testes devem verificar se a classe está armazenando corretamente os ícones existentes e se eles são retornados corretamente quando solicitado.

Teste de criação de ícone para bracket ou cor inválidos: Descrição: Verificar se a classe GutterIconManager lida corretamente com bracket ou cor inválidos.

3 Entregáveis de Teste

- Plano de teste

4 Necessidades de Recursos e Ambiente

4.1 Ferramentas de Teste

Utilização do Jest.

4.2 Ambiente de Teste

Mencione os requisitos mínimos **de hardware** que serão usados para testar o software.

Os seguintes softwares são necessários, além de softwares específicos do cliente.

- Windows 8 e superior
- Office 2013 e superior
- MS Exchange, etc.

5 Termos / Acrônimos

Faça uma menção a quaisquer termos ou acrônimos usados no projeto

TERMO / ACRÔNIMO	DEFINIÇÃO
API	*Application Program Interface*
SUT	*Software Under Test*