

Estória Estrutural: Calculadora

Estória

As calculadoras não são apenas uma das ferramentas mais úteis disponíveis, mas também são uma ótima maneira de entender a interface do usuário e o processamento de eventos em um aplicativo. Este problema você testará uma calculadora que suporta aritmética básica cálculos em números inteiros.

O estilo fica por sua conta, então use sua imaginação e seja criativo! Você pode também pensar que vale a pena experimentar o aplicativo de calculadora em seu dispositivo móvel para entender melhor a funcionalidade básica e os casos extremos.

Comportamento Esperado pelo Usuário

- ☐ Você não pode usar a função ``eval()``` para executar cálculos
- ☐ O usuário pode ver um display mostrando o número atual inserido ou o resultado da última operação.
- ☐ O usuário pode ver um bloco de entrada contendo botões para os dígitos de 0 a 9, operações - '+', '-', '/' e '=', um botão 'C' (para limpar) e um 'AC' botão (para limpar tudo).
- ☐ O usuário pode inserir números como sequências de até 8 dígitos clicando em dígitos no bloco de entrada. A entrada de quaisquer dígitos maiores que 8 será ignorada.
- ☐ O usuário pode clicar em um botão de operação para exibir o resultado disso operação em:
 - * o resultado da operação anterior e o último número digitado OU
 - * os dois últimos números inseridos OU
 - * o último número digitado
- ☐ O usuário pode clicar no botão 'C' para limpar o último número ou o último
- ☐ Operação. Se a última entrada do usuário foi uma operação, a exibição será atualizado para o valor que o precedeu.
- ☐ O usuário pode clicar no botão 'AC' para limpar todas as áreas de trabalho internas e para definir a exibição para 0.
- ☐ O usuário pode ver 'ERR' exibido se alguma operação exceder o Máximo de 8 dígitos.

Para a realização do teste você deve:

- ler [Template para Estudo de Caso Testes Estruturais](#);
- clonar algum dos repositórios abaixo em seu computador;
- selecionar três ou mais itens de comportamento esperado;
- selecionar o trecho de código que diz respeito a esse comportamento esperado;
- abstrair modelar o grafo de fluxo de controle do trecho de código escolhido (sugestão de ferramenta [Code to Flowchart](#));
- escrever os casos de testes identificados no modelo de grafo de fluxo de controle;
- executar os casos de teste escritos e se possível realizar uma análise de cobertura de código;

Exemplos de projetos

- Javascript Calculator (<https://codepen.io/giana/pen/GJMBEv>)
- BHMBBS - JS-Neumorphic-Calculator (<https://barhouum7.github.io/JS-Neumorphic-Calc.github.io/>)
- Javascript iOS Style Calculator (<https://codepen.io/ssmkhrj/full/jOWBQqO>)
- React Calculator (<https://codepen.io/mjijackson/pen/xOzyGX>)
- Javascript-CALC (<https://github.com/x0uter/javascript-calc>)
- Sample Calculator (<https://sevlasnog.github.io/sample-calculator>)
- Python Calculator (<https://github.com/kana800/Side-Projects/tree/master/1-Beginner/calculator>)

Adaptado de: (“app-ideas/Projects/1-Beginner/Calculator-App.md at master · florinpop17/app-ideas”, [s.d.]

Referência:

app-ideas/Projects/1-Beginner/Calculator-App.md at master · florinpop17/app-ideas. Disponível em:

<<https://github.com/florinpop17/app-ideas/blob/master/Projects/1-Beginner/Calculator-App.md?plain=1>>. Acesso em: 9 nov. 2023.