Trabalho de Qualidade e Teste

Descrição do Sistema

O Tabela-verdade é um gerador de tabela-verdade de expressões lógicas *open source* criado com React e Gatsby para *web*. A aplicação foi criada para a disciplina de Fundamentos Matemáticos para Computação, no 1° período do curso de Sistemas de Informação, da Universidade Federal Fluminense (UFF). Atualmente, esse objetivo se estendeu e agora está disponível para qualquer pessoa utilizá-lo. A aplicação é formada por quatro módulos principais: source, lexer, parser e generator. Esses módulos operam em sequência para transformar uma expressão lógica arbitrária de entrada em uma tabela-verdade na saída. Pretendemos testar cada um dos módulos e suas dependências unitariamente, testar a integração entre eles e por fim testar a aplicação em seu caso de uso típico.

Medidas dos atributos de qualidade

Escala de 0 a 10.

Qualidade do produto

- Adequação Funcional
 - o Atribuição: 5
 - Justificativa: Como esse projeto n\u00e3o foi movido por fins comerciais, n\u00e3o tivemos um cliente externalizando suas necessidades. Entretanto, alguns requisitos foram levantados que o produto final deveria, no m\u00ednimo, ter,
- Eficiência de desempenho
 - o Atribuição: 8
 - Justificativa: dado que a aplicação é executada inteiramente no dispositivo do usuário final, não temos o controle sobre a configuração do ambiente de execução, portanto gostaríamos que a aplicação tivesse um desempenho satisfatório em ambientes com recursos computacionais limitados.
- Compatibilidade
 - Atribuição: 3
 - Justificativa: Considerando que o aplicativo é um stand alone que tem um objetivo específico, não buscamos compatibilidade com outros sistemas.
- Usabilidade

Atribuição:9

Justificativa: Como a aplicação busca fornecer de forma rápida um resultado

de uma operação matemática com um certo grau de complexidade, a

Usabilidade se vê como um dos pilares do projeto; Usando diversos símbolos

que encontramos na literatura oferecendo para o usuário final adequabilidade

aos padrões matemáticos, mas também uma facilidade de uso e de

aprendizado do sistema.

Confiabilidade

o Atribuição: 10

Justificativa: Por se tratar de um aplicativo acadêmico, busca-se garantir o

menor número possível de estados de falha, e que a aplicação fique

disponível o maior tempo possível.

Segurança

o Atribuição: 3

o Justificativa: a aplicação não é projetada para realizar nenhuma troca de

informação com ambientes externos ou manter quaisquer tipos de estado

sobre sua execução e informações inseridas durante esse processo; portanto

a segurança não deve ser um requisito estritamente priorizado.

Manutenibilidade

o Atribuição: 10

o Justificativa: sendo open source, é fundamental que a aplicação tenha o

código fonte altamente analisável e modificável, facilitando o entendimento

por parte de terceiros interessados em estudá-lo e/ou contribuir diretamente

para o seu desenvolvimento.

Portabilidade

Atribuição: 8

o Justificativa: Considerando o aplicativo como uma aplicação web, busca-se

que o mesmo seja compatível com o maior número de OS e Navegadores

possíveis, buscando alienar o menor número possível de usuários finais.

Qualidade em uso

Eficácia

Atribuição: 10

Justificativa: Por se tratar de um domínio matemático, a aplicação deve

sempre ser a mais precisa possível, erros de output não são tolerados.

Eficiência

o Atribuição: 6

 Justificativa: Tratando-se de um aplicação web, entende-se que devemos nos preocupar com a Eficiência, mas ao mesmo tempo oferecer algum grau de leniência no executar da aplicação.

Satisfação

- Atribuição: 9
- Justificativa: Como o sistema busca facilitar a vida do usuário final nesse domínio, entende-se que a satisfação do mesmo é inegociável.

Safety

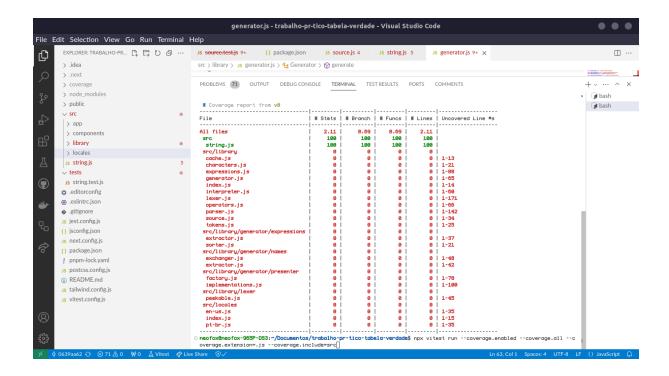
- o Atribuição: 1
- Justificativa: A aplicação não lida com dados sensíveis, credenciais ou ameaças diretas para o usuário final. Por tanto, não existe nenhum risco intrínseco a sua utilização.

Usabilidade

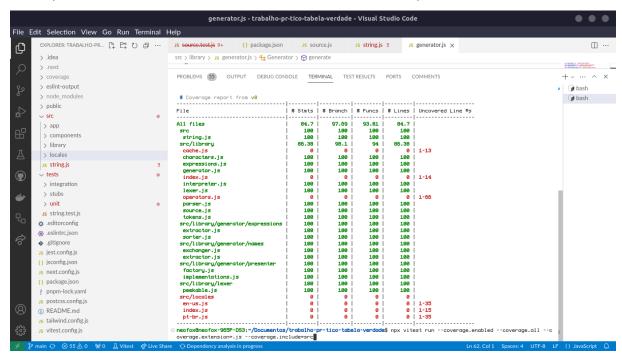
- o Atribuição:9
- Justificativa: Como a aplicação busca fornecer de forma rápida um resultado de uma operação matemática com um certo grau de complexidade, a Usabilidade se vê como um dos pilares do projeto; Usando diversos símbolos que encontramos na literatura oferecendo para o usuário final adequabilidade aos padrões matemáticos, mas também uma facilidade de uso e de aprendizado do sistema.

Análise com o Vitest Coverage

O Vitest Coverage é uma opção para a ferramenta Vitest que gera um relatório da cobertura do código fonte por testes unitários. Utilizamos a ferramenta para tomarmos conhecimentos das partes do código fonte não testadas e então projetarmos testes unitários paras as mesmas.



Antes (commit 0639aa62e88d78ffc0ec91e6b8413c7d3c50f339)



Depois (commit e25dc524d9c665f822c1b247b51429f01d2acf3d)

Análise com o ESLint

ESLint é uma ferramenta de análise de código fonte que reporta inconformidades na escrita do código fonte em relação às regras de escrita previamente estabelecidas para o projeto. Visando obter um estilo de escrita padronizado ao longo do código fonte de toda a

aplicação, refatoramos trechos de código conforme as inconformidades reportadas pela ferramenta.

Generator — Brendo Costa

```
generator.js - trabalho-pr-tico-tabela-verdade - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Ð
                                                    src > library > Js generator.js > 😭 Generator > 🈭 prepareNames
        > public
                                                      10
                                                      11 export default class Generator {
         > app
                                                                constructor (parser) {
                                                      13
                                                                     this.parser = parser:

√ library

         > generator
                                                      15
                                                      16
                                                                 prepareNames (expression) {
          Js cache.js
                                                      17
                                                                      const extractor = new NamesExtractor();
         Js characters.js
                                                      18
          Js expressions.js
                                                      19
                                                                    extractor.extract(expression);
         us generator.js
                                                      20
                                                      21
                                                                    const exchanger = new NamesExchanger();
         Js interpreter.js
                                                      22
          Js lexer.js
                                                            return exchanger.exchange(extractor.getNames());
                                                      23
          Js operators.js
         Js parser.js
                                                      25
                                                                 prepareExpressions (expression) {
                                                      26
         Js tokens.js
                                                                      const extractor = new ExpressionsExtractor();
         > locales
         JS string.js
                                                     PROBLEMS 67 OUTPUT DEBUG CONSOLE TERMINAL TESTRESULTS PORTS COMMENTS Filter (e.g. text, **/*.ts, !**/node_modules/... \Top{This posterior}

√ tests

√ Js generator.js src/library (12)

                                                        > stubs
                                                        ⊗ Expected 'this' to be used by class method 'prepareExpressions'. eslint(class-methods-use-this) [Ln 26, Col 5]

√ unit / library

                                                        Expected 'this' to be used by class method 'generateValues'. eslint(class-methods-use-this) [Ln 36, Col 5]

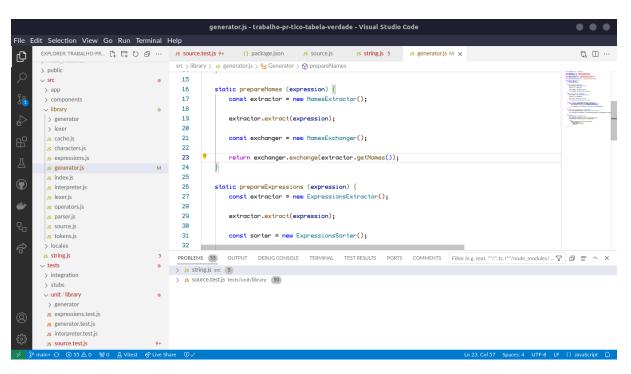
⊗ Expected parentheses around arrow function argument. eslint(arrow-parens) [Ln 37, Col 46]

          Js expressions.test.js
                                                         \textcircled{8 Expected parentheses around arrow function argument. eslint} (\underline{\text{arrow-parens}}) \text{ [Ln 40, Col 36]} 
                                                        ⊗ Unexpected block statement surrounding arrow body; move the returned value immediately after the `=>`. eslint(arrow-body-style) [Ln 40, Col 50]
          us interpreter.test.is

⊗ Expected 'this' to be used by class method 'generatePresentableExpressions'. eslint(class-methods-use-this) [Ln 46, Col 5]

          Js source.test.js
                                                          Expected dot to be on same line as object, esli
```

Antes



Depois

Lexer — Mathews Reis

```
src > library > JS lexer.js > 😭 Lexer > 😭 next
      38
      39
                    } from './tokens';
      40
      41
                    export function isWhiteSpace (ch) {
                             45
       46
       47
      48
      49
      50
      51
      52
                   export function isNumber (ch) {
      53 | return (typeof ch === 'string' && ch.length === 1) && (ch >= '0' && ch <= '9');
      54
      55
                   export function isAlphabetic (ch) {
   return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z');</pre>
      56
      57
      58
      59
      60
                   export default class Lexer {
      61
                             constructor (source) {
      62
                              this.source = source;
      63
      64
      65
                              next () {
      66
                                this.source.skipWhile(isWhiteSpace);
      67
                                                                                                                                Ln 71, Col 24 Espaços: 4 UTF-8 LF ( ) JavaScript Layout: Portuguese (Brazilian ABNT) 🚨
1 (A) 0 Z
  Eii [FLEANMF'WG 12 | SOnlice'le 4 | Sonlice'le 5 | 
                                                                                                                                                                                                                                                                                                                   ŀŶ Ш ...
    \mathsf{src} > \mathsf{library} > \ \mathbf{JS} \ \ \mathsf{lexer,js} > \mathbf{\mbox{\mbox{$\mbox{$^{\bullet}$}$}}\ \mathsf{Lexer} > \ \pmb{\diamondsuit} \ \ \mathsf{next}
       77
       78
                                                    case CARET:
       79
                                                   case ASTERISK:
       80
                                                   case AMPERSAND:
                                                   case LOGICAL AND:
       81
       82
                                                        this.source.bump();
       83
       84
                                                          return new And();
       85
       86
                                                    case PLUS:
                                                    case VERTICAL LINE:
       87
       22
                                                    case LOGICAL OR:
       89
                                                             this.source.bump();
       98
       91
                                                             return new Or();
       92
       93
                                                    case RIGHTWARDS ARROW:
       94
                                                   case RIGHTWARDS DOUBLE ARROW:
       95
                                                          this.source.bump();
        96
       97
                                                             return new Conditional();
       98
       99
                                                    case LEFT RIGHT ARROW:
     100
                                                   case LEFT RIGHT DOUBLE ARROW:
     101
                                                              this.source.bump();
     102
     103
                                                             return new Biconditional();
     104
                                                    case OPENING PARENTHESIS:
     105
     106
                                                            this.source.bump();
                                                                                                                               Ln 75, Col 1 Espaços: 4 UTF-8 LF ( ) JavaScript Layout: Portuguese (Brazilian ABNT)
∆ 0 (A) 1
```

Antes

```
JS lexer.test.js 9+, U
                                             JS lexer.js 3, M × @ .eslintrc.json M
                                                                              JS source.js 4
 [Preview] README.md
 src > library > JS lexer.js > ...
  40
  41
        export function isWhiteSpace (ch) {
   42
            // New line
            return ch === '\n' ||
   43
   44
            // Return
            ch === '\r' ||
   45
   46
            // Return
             ch === '\t' ||
   47
   48
            // Tab
                ch === '\v' ||
   49
   50
            // Vertical tab
            ch === '\b' ||
   51
            // Backspace
   52
   53
                ch === '\f' ||
            // Form feed
   54
   55
                ch === '\x20' ||
   56
            // Space
   57
              ch === '\xA0';
   58
            // No-break space
PROBLEMAS (202) SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS (1) COMENTÁRIOS
src > library > JS lexer.js > 😝 Lexer > ♦ next
 76
              this.source.skipWhile(isWhiteSpace);
 77
 78
               const ch = this.source.peek();
 79
 80
               switch (ch) {
                  case TILDE:
 81
                  case EXCLAMATION:
 82
 83
                   case LOGICAL_NOT:
 84
                      this.source.bump();
 85
 86
                      return new Not();
 87
 88
                   case CARET:
 89
                   case ASTERISK:
                   case AMPERSAND:
 90
 91
                   case LOGICAL_AND:
 92
                      this.source.bump();
 93
                      return new And();
 94
 95
 96
                   case PLUS:
 97
                   case VERTICAL_LINE:
                   case LOGICAL OR:
 98
 99
                      this.source.bump();
100
101
                       return new Or();
102
103
                   case RIGHTWARDS_ARROW:
                   case RIGHTWARDS_DOUBLE_ARROW:
104
                       this.source.bump():
                                                           Ln 125, Col 21 Espaços: 4 UTF-8 LF () JavaScript Layout: Portuguese
0 (A) 1
```

Depois

Source — Isaac Luiz

```
🔞 Go source ja - Irabahho-pr-isco-tab: X 🕒 Isting directory /esint-output/ X 📳 Tabahho-Qualidade e Teste - D: X 🗎 Tabahho-Qualidade e Teste - D: X 🗎 O Coverage report missing for unit: X 📳 TemplateParoDeTeste - Docum: X + +
                  C https://upgraded-trout-q7gwv54594g294q6.github.dev
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Į AVIS Management. 🐧 Whatlapp 🚭 YouTube 📜 Google Sale de Avis 👸 Unikedin 💍 Sistema de Certifica. 🛅 Compras 🛅 For que os talentos... 📆 The 9 books that co... 🔞 Headdert Gamer Sec. 💍 Véja onde está o se... 🞜 Creatina Monochidur. 🔞 Caneca Cubo Super...
                 BPUCAAOOR ... IS sourcejs 4 X | Estendac Estint

VERMANNOPRICOTABELA VERDADE (CODESTAC...
) idea

) idea

) next

) constructor (input) {

constructor (input) {

this.offsper - input;

this.finper - input;

}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            □ …
  <sub>C</sub>
    P
    d 
    ₽
                                                                                                                                                peek () {
    return this.input[this.offset];
                                                                                                                                                 bump () {
   this.offset += 1;
                                                                                                                                                 takeWhile (f) {
   const start = this.offset;
                                                                                                                                                        this.skipWhile(f);
                                                                                                                                                      const end = this.offset;
                                                                                                                                                 return this.span(start,_end);
}
                                                                                                                                                  skipWhile (f) {
   while (f(this.peek())) {
      this.bump();
}
                                                                                                                        PROBLEMAS () SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS () COMENTÁRIOS Filtrar (por exemplo, texto, ""/"ts, !""/node, modules/"")
                                                                                                      PROBLEMS: ② SAINA CONCIDE DE PRIMARA I TENhama. Influence of the problems of 
    () package ison
    ! pnpm-lock.yaml
> ESTRUTURA DO CÓDIGO
> LINHA DO TEMPO
                            tes: upgraded trout $2 main* ← 51.41 ⊗ 4 △ 0 ₩ 2
                                                                                                                                                                                                                                                                                                                                                                                                                                          Ln 9, Col 6 Espaços: 4 UTF-8 LF () JavaScript Ø Port: 5500 Layout: Portuguese (Brazilian ABNT) 🚨
```

Antes

```
Discreption of the control of the co
```

Depois

Peekable — Daniel Lima

```
📢 File Edit Selection View Go Run …
                                                                                                                                                                                                                                                                                                                                                                               P trabalho-pr-tico-tabela-verdade
                                                                                                                                                   ··· us peekable.js 5 🗙
C
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              EXPLORER
                                                                                                                                                                                      src > library > lexer > JS peekable.js > 😘 PeekableLexer > 😚 constructor
                                                                                                                                                                                                 c) Nobary Free preciames - Free constructions - Free construction 
                                      > 📭 components

> 🛅 library
                                              > expressions
> names
                                                                                                                                                                                                                                       wasPeeked () {
   return this.peeked != null;
H
                                                  > presenter
                                          ✓ 🗁 lexer
                                                                                                                                                                                                                                      wasNotPeeked () {
    return this.peeked == null;
                                                      Js cache.js
                                                      characters.js
sexpressions.js
generator.js
                                                                                                                                                                                                                               isPeekTypeOf (type) {
   const peeked = this.peek();
                                                      us index.js
interpreter.js
                                                       us lexer.js
                                                      operators.js
parser.js
                                                                                                                                                                                                                                        peek () {
    if (this.wasNotPeeked()) {
        this.peeked = this.lexer.next();
    }
}
                                                      us tokens.js
                                        ∨ 👼 locales
                                                   us en-us.js
us index.js
us pt-br.js
                     > OUTLINE
                       > TIMELINE
     $\mathbb{V}$ main \cdot \mathbb{W}$ \otimes 5 \textit{\Delta} 0 \quad \mathbb{W}$ 0 \quad \mathbb{P}$ Live Share
```

antes

```
★ File Edit Selection View Go Run

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 Ð
                                                                                                                               ··· peekable.js M X
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              ე ↔ ⊹ ⊹ № 🗓 …
                                                                                                                                                       src > library > lexer > 🐹 peekable.js > ધ PeekableLexer > 😚 peek
                        V TRABALHO-PR-TICO-TABELA-VERDADE
                                                                                                                                                                                 yary | lexer | properties | proceedings | pr
                             96

✓ 

    library

√ ■ generator

                                                                                                                                                                                              wasPeeked () {
    return this.peeked !== null;
}
   <del>u</del>
                                           > names
                                            > presenter
                                                                                                                                                                                                wasNotPeeked () {
    return this.peeked === null;
                                                 us peekable.js
                                               s cache.js characters.js
                                               expressions.js
generator.js
index.js
                                                                                                                                                                                               isPeekTypeOf (type) {
   const peeked = this.peek();
                                                                                                                                                                                                               return peeked instanceof type;
                                               interpreter.js
lexer.js
                                                                                                                                                                                               peek () []
    if (this.wasNotPeeked()) {
        this.peeked = this.lexer.next();
}
                                                us operators.js
                                               source.js
tokens.js
                                             us en-us.js
us index.js
us pt-br.js
   > OUTLINE
     🛂 🖟 main* 🗘 🐉 🛞 0 🖄 0 💖 0 ể Live Share (,) : Scanning. 🖷 Connected to Discord 🕴 🗘 Isaac Ferreira, Iast week 📙 1.27, Col 6 Spaces: 4 UTF-8 LF () JavaScript 🖗 Go Live 🛷 Prettier 🕏
```

depois