Trabalho Cálculo II

Links

- **Código fonte no GitHub:** https://github.com/LucasVinicius314/integral-e-do-mal
- **Ambiente do projeto hospedado:** https://integral-e-do-mal.herokuapp.com/

Integrantes do grupo:

- Breno Lopes
- João Pedro Barroso
- Lucas Carvalho
- Lucas Coelho

Representação Computacional Escolhida

O grupo optou por escolher Dart como linguagem de desenvolvimento por uma maior portabilidade e conhecimento do grupo. O projeto está sendo hospedado na internet e possui a possibilidade de ser baixado. Além da web o projeto estará disponível na Play Store.

Detalhes da codificação

Os cálculos são divididos em duas partes, a parte de cima (numerador) e a parte de baixo (denominador).

```
static String calcularIntegral({
    required String cima,
    required String baixo,
    final int sup = 0,
    final int inf = 0,
  })
```

Após essa separação temos os seguintes Métodos:

```
/// Método para calcular funções de segundo grau.
  static List<Numero> _segundoGrau(final List<Numero> baixos) {
    final respostas = <Numero>[];

    if (baixos.length == 3) {
      // Só calcula se tiver 3 elementos na lista

      // Calcular só com 2
      final bhaskara = _calculaBhaskara(baixos);

      respostas.add(bhaskara.primeiro);
      respostas.add(bhaskara.segundo);
    } else if (baixos.length == 2) {
      // Calcular função incompleta
      final segundoIncompleta = _segundoIncompleta(baixos);

      respostas.add(segundoIncompleta.primeiro);
      respostas.add(segundoIncompleta.segundo);
    }

    return respostas;
  }
```

```
/// Método para calcular funcão de segundo grau utilizando bhaskara.
  static Bhaskara _calculaBhaskara(final List<Numero> baixos) {
    final first = baixos.first;
    final second = baixos[1];
    final third = baixos[2];

    // Cálculo de delta
    var delta = math.pow(second.numX, 2) - 4 * first.numX * third.numSX;
    var bhaskaraPositivo =
        (-1 * second.numX + math.sqrt(delta)) / 2 * first.numX;
    var bhaskaraNegativo =
        (-1 * second.numX - math.sqrt(delta)) / 2 * first.numX;

    final primeiroNumX = first.numX;
    final segundoNumX = first.numX;

    final primeiroNumSX = bhaskaraPositivo * -1 * first.numX;
    final segundoNumSX = bhaskaraNegativo * -1 * first.numX;

    var primeiro = Numero(numSX: primeiroNumSX, numX: primeiroNumX);
    var segundo = Numero(numSX: segundoNumSX, numX: segundoNumX);

    return Bhaskara(primeiro: primeiro, segundo: segundo);
  }
```

```
// Método para calcular funções de segundo grau incompletas.
  static Bhaskara _segundoIncompleta(final List<Numero> baixos) {
    final first = baixos.first;
    final second = baixos[1];

    final primeiro = Numero(numSX: 0, numX: first.numX);
    final segundo = Numero(numSX: 0, numX: first.numX);

    if (second.numSX < 0) {
      second.numSX *= -1;

      primeiro.numSX = math.sqrt(second.numSX);
      segundo.numSX = primeiro.numSX * -1;
    } else {
      primeiro.numX = math.sqrt(second.numX);
      segundo.numSX = primeiro.numSX;
    }

    return Bhaskara(primeiro: primeiro, segundo: segundo);
  }
```

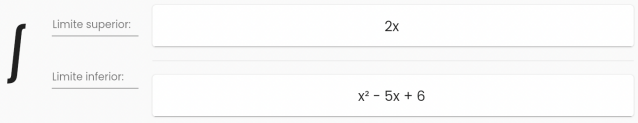
Após fazer os cálculos necessários utilizando esses métodos fazemos um tratamento para as expressões em uma lista de números.

Outros métodos utilizados como complementos podem ser visualizados no repositório no caminho “integral-e-do-mal/lib/core/calculadora.dart”

Resultado das integrais

$$\int_0^1 \frac{3x}{(x+1)(x+2)}dx$$

b) $\int \frac{2x}{x^2 - 5x + 6}\, dx$



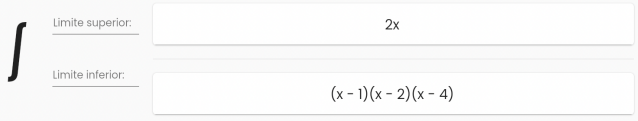
Resultado

Clique para copiar

$+ 6\ln(x - 3) - 4\ln(x - 2) + C$

c) $\int \frac{2x}{(x - 1)(x - 2)(x - 4)}\, dx$

Integral é do mal



Resultado

Clique para copiar

$- 0.5\ln(x - 1) - 4\ln(x - 2) + 1.6\ln(x - 4) + C$