



Universidade Federal do Pampa

Disciplina de Técnicas Digitais

circuitMaps

Quine-Mccluskey

Tiago Leite Brito

Bagé / 2017

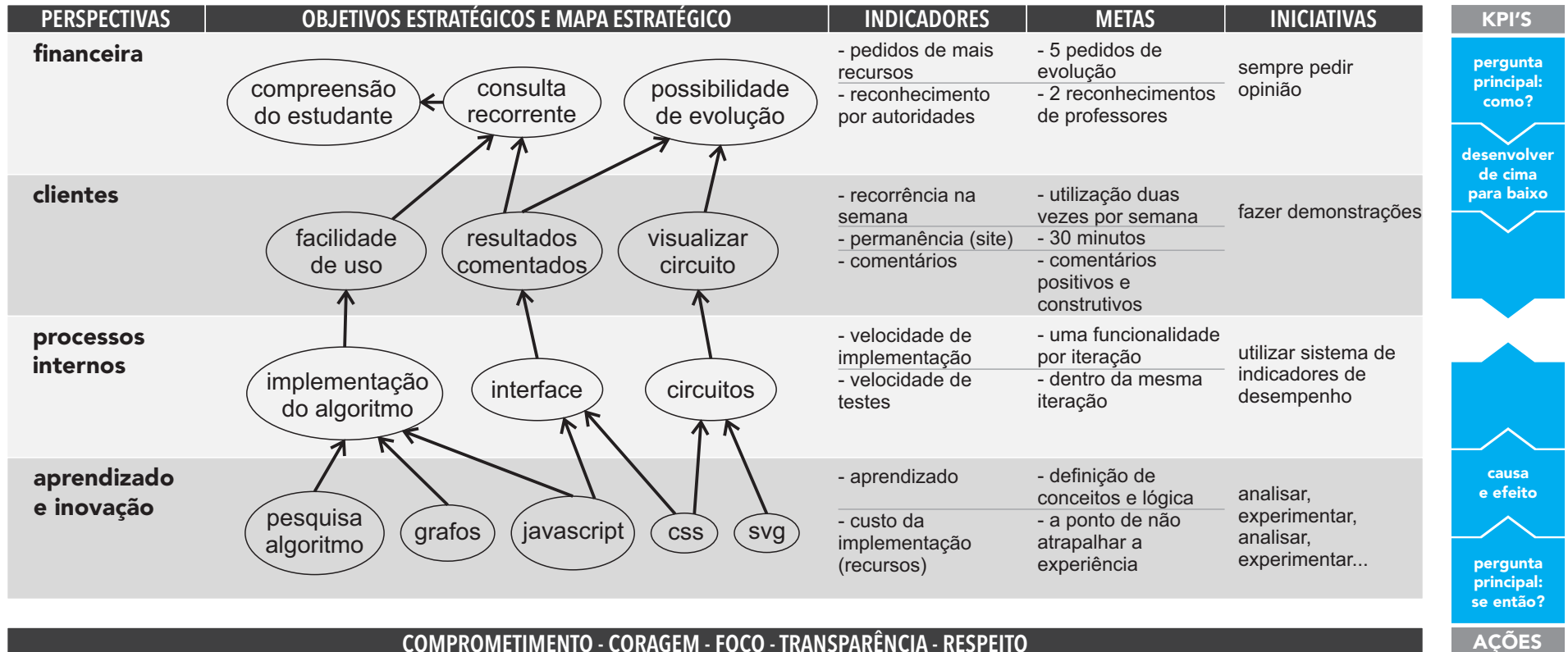
método de desenvolvimento - BscScrum

MAPA ESTRATÉGICO

CircuitMaps - software para estudo de mapas de Karnaugh e QMC

Data: 11/11/17
Release/sprint: 0/0
Revisão: 00

VISÃO	material didático interativo sobre mapas de Karnaugh para estudantes de técnicas digitais em um ambiente online.										
MISSÃO	desenvolver uma ferramenta para solução de mapas de Karnaugh prezando pelo desempenho e usabilidade.										
TEMAS ESTRATÉGICOS	algoritmos			interface			entradas e saídas			extras	
ÉPICOS ESTRATÉGICOS	pesquisa (sem comparação)	implementação (dinâmico)	tabela de cobertura	entrada com mapa	entrada com tabela	interface com usuário	importar mapa	importar tabela	exportar resultado	montar circuito	máquina de estados
RESULTADOS ESTRATÉGICOS	melhor desempenho	poder trabalhar com o máximo de variáveis possível	análise dos resultados das comparações	poder informar um mapa para analisar	informar a tabela verdade para gerar mapa e analisar	simplificar o processo de inserção de dados e passo a passo	poder importar um mapa de um arquivo	poder importar uma tabela de um arquivo	gerar um resultado que possa ser baixado ou impresso	gerar o circuito da solução	implementar análise de máquinas de moore



COMPROMETIMENTO - CORAGEM - FOCO - TRANSPARÊNCIA - RESPEITO

product backlog

10/11			inception
	iteração 1	recapitular código anterior verificar possibilidade de utilizar grafos/árvores nas comparações verificar estrutura dos grupos nas comparações verificar estrutura das colunas nas comparações verificar possibilidade de utilizar mais de 8 variáveis	tema: pesquisa
	iteração 2		
	iteração 3		
	iteração 4		
	iteração 5		
15/11			release: 0
	iteração 1	implementar funções e variáveis implementar algoritmos de comparação e alternativo testes de integridade e desempenho	tema: algoritmo
	iteração 2		
	iteração 3		
18/11			release: 1
	iteração 1	interface base adicionar mapas adicionar tabelas	tema: interface
	iteração 2		
	iteração 3		
	iteração 4		
22/11			release: 2
	iteração 1	importar mapas importar tabelas exportar resultado	tema: entradas e saídas
	iteração 2		
	iteração 3		
	iteração 4		
	iteração 5		
	iteração 6		
29/11			release: 3
	iteração 1	circuito máquina de estados	tema: extras
	iteração 2		
	iteração 3		
	iteração 4		
	iteração 5		
05/12	iteração 6		release: 4

código anterior

ab/cde	000	001	011	010	110	111	101	100
00	0 ▾ 0	X ▾ 1	1 ▾ 3	0 ▾ 2	0 ▾ 6	1 ▾ 7	0 ▾ 5	0 ▾ 4
01	1 ▾ 8	X ▾ 9	X ▾ 11	0 ▾ 10	0 ▾ 14	0 ▾ 15	0 ▾ 13	0 ▾ 12
11	0 ▾ 24	0 ▾ 25	0 ▾ 27	0 ▾ 26	0 ▾ 30	0 ▾ 31	0 ▾ 29	0 ▾ 28
10	1 ▾ 16	1 ▾ 17	X ▾ 19	0 ▾ 18	X ▾ 22	0 ▾ 23	1 ▾ 21	1 ▾ 20
Submit	Modelo	Modelo1	Modelo2	Modelo3	Modelo4	Modelo5		
coluna0	coluna1	coluna2	coluna3					
			5					
00001 (1) 01000 (8) 10000 (16)	000_1 (1,3) 0_001 (1,9) _0001 (1,17) 0100_ (8,9) 1000_ (16,17) 10_00 (16,20)	0_0_1 (1,3,9,11) _00_1 (1,3,17,19) 0_0_1 (1,9,3,11) _00_1 (1,17,3,19) 10_0_ (16,17,20,21) 10_0_ (16,20,17,21)	6					
00011 (3) 01001 (9) 10001 (17) 10100 (20)	00_11 (3,7) 0_011 (3,11) _0011 (3,19) 010_1 (9,11) 100_1 (17,19) 10_01 (17,21) 1010_ (20,21) 101_0 (20,22)							
00111 (7) 01011 (11) 10011 (19) 10101 (21) 10110 (22)								

algoritmo Quine-Mccluskey

coluna 0

00000 (0)

00001 (1)
01000 (8)
10000 (16)

00011 (3)
01001 (9)
10001 (17)
10100 (20)

00111 (7)
01011 (11)
10011 (19)
10101 (21)
10110 (22)

01111 (15)
10111 (23)

coluna 1

0000_ (0,1)
0_000 (0,8)
_0000 (0,16)

000_1 (1,3)
0_001 (1,9)
_0001 (1,17)
0100_ (8,9)
1000_ (16,17)
10_00 (16,20)

00_11 (3,7)
0_011 (3,11)
_0011 (3,19)
010_1 (9,11)
100_1 (17,19)
10_01 (17,21)
1010_ (20,21)
101_0 (20,22)

0_111 (7,15)
_0111 (7,23)
01_11 (11,15)
10_11 (19,23)
101_1 (21,23)
1011_ (22,23)

coluna 2

0_00_ (0,1,8,9)
000 (0,1,16,17)
0_00_ (0,8,1,9)
000 (0,16,1,17)

0_0_1 (1,3,9,11)
_00_1 (1,3,17,19)
0_0_1 (1,9,3,11)
_00_1 (1,17,3,19)
10_0_ (16,17,20,21)
10_0_ (16,20,17,21)

0__11 (3,7,11,15)
_0__11 (3,7,19,23)
0__11 (3,11,7,15)
_0__11 (3,19,7,23)
10__1 (17,19,21,23)
10__1 (17,21,19,23)
101__ (20,21,22,23)
101__ (20,22,21,23)

CDE \ AB	000	001	011	010	110	111	101	100
00	1	x	1	0	0	1	0	0
01	1	x	x	0	0	1	0	0
11	0	0	0	0	0	0	0	0
10	1	1	x	0	x	1	1	1

implicantes primos:

0_00_ (0,1,8,9)
000 (0,1,16,17)
0_0_1 (1,3,9,11)
_00_1 (1,3,17,19)
10_0_ (16,17,20,21)

implicantes primos essenciais:

0_00_ (0,1,8,9)
0__11 (3,7,11,15)
000 (0,1,16,17)
101__ (20,21,22,23)

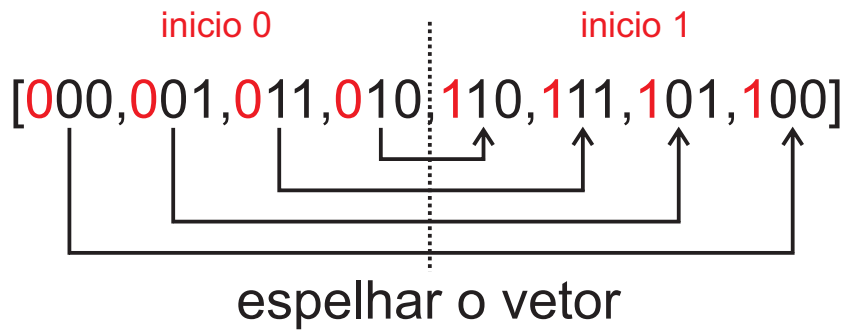
0__11 (3,7,11,15)
_0__11 (3,7,19,23)
10__1 (17,19,21,23)
101__ (20,21,22,23)

tabela de cobertura

	0	3	7	8	15	16	17	20	21	23
0_00_ (0,1,8,9)	x			x						
000 (0,1,16,17)	x					x	x			
0_0_1 (1,3,9,11)		x								
_00_1 (1,3,17,19)		x					x			
10_0_ (16,17,20,21)						x	x	x	x	
0__11 (3,7,11,15)		x	x		x					
_0__11 (3,7,19,23)		x	x							x
10__1 (17,19,21,23)							x		x	x
101__ (20,21,22,23)								x	x	x

Resposta: $\bar{A}\bar{C}\bar{D} + \bar{A}DE + \bar{B}\bar{C}\bar{D} + A\bar{B}C$

análise de leitura do mapa de karnaugh



2 variáveis

b \ a	0	1
0	00	01
1	02	03

3 variáveis

bc \ a	00	01	11	10
0	00	01	03	02
1	04	05	07	06

4 variáveis

cd \ ab	00	01	11	10
00	00	01	03	02
01	04	05	07	06
11	12	13	15	14
10	08	09	11	10

5 variáveis

cde \ ab	000	001	011	010	110	111	101	100
00	00	01	03	02	06	07	05	04
01	08	09	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

número **ímpar** de variáveis dobra a tabela para a **direita**,
número **par** de variáveis dobra a tabela para **baixo**

0	0	0	0	
0	0	0	1	2
0	0	1	1	
0	0	1	0	4
0	1	1	0	
0	1	1	1	
0	1	0	1	
0	1	0	0	8
1	1	0	0	
1	1	0	1	
1	1	1	1	
1	1	1	0	
1	0	1	0	
1	0	1	1	
1	0	0	1	
1	0	0	0	16

proposta alternativa - prever comparação

```
var g0 = 0;
var g1 = [1,2,4,8,16,32,64,128];
var g2 =
[3,5,6,9,10,12,17,18,20,24,33,34,36,40,48,65,66,68,72,80,
96,129,130,132,136,144,160,192];
var g3 =
[7,11,13,14,19,21,22,25,26,28,35,37,38,41,42,44,49,50,
52,56,67,69,70,73,74,76,81,82,84,88,97,98,100,104,112,131,1
33,134,137,138,140,145,146,148,152,161,162,168,176,193,19
4,196,200,208,212,224];
var g4 =
[15,23,27,29,30,39,43,45,46,51,53,54,57,58,60,71,75,77,
78,83,85,86,89,90,92,99,101,102,105,106,108,113,114,116,12
0,135,139,141,142,147,149,150,153,154,156,163,165,166,169
,170,172,177,178,180,184,195,197,198,201,202,204,208,210,
212,216,225,226,228,232,240];
```

```
function coluna(v) {
  //grupo0
  if (mapa[0] == 1 || mapa[0] == "x") {
    var tex = document.querySelector("#c0g0");
    tex.innerHTML = "(0)";
  }
```

grupo1 = 1

```
  //grupo1
  var texto = "";
  for (let i = 0; i < v; i++) {
    var temp = parseInt(g1[i]);
    if ((mapa[temp] == 1) || (mapa[temp] == "x")) {
      texto = texto+"("+temp+")<br>";
    }
    var tex = document.querySelector("#c0g1");
    tex.innerHTML = texto;
  }
```

grupo1 = n

```
//grupo2
var v2 = ((v-1)*v)/2;
texto = "";
for (let i = 0; i < v2; i++) {
  var temp = parseInt(g2[i]);
  if ((mapa[temp] == 1) || (mapa[temp] == "x")) {
    texto = texto+"("+temp+")<br>";
  }
  var tex = document.querySelector("#c0g2");
  tex.innerHTML = texto;
}
```

grupo2 = $\frac{(n-1)n}{2}$

```
//grupo3
var v3 = 0;
for (let i = 0; i < v; i++) {
  v3 = v3+(((v-1)*v)/2);
}
texto = "";
for (let i = 0; i < v3; i++) {
  var temp = parseInt(g3[i]);
  if ((mapa[temp] == 1) || (mapa[temp] == "x")) {
    texto = texto+"("+temp+")<br>";
  }
  var tex = document.querySelector("#c0g3");
  tex.innerHTML = texto;
}
```

grupo3 = $\sum_{n=1}^n \frac{(n-2)(n-1)}{2}$

proposta alternativa - prever comparação

```
var coluna0 = [ ["0"],
  ["1","10","100","1000","10000"],
  ["11","101","110","1001","1010","1100","10001","10010","10100","11000"],
  ["111","1011","1101","1110","10011","10101","10110","11001","11010","11100"],
  ["1111","10111","11011","11101","11110"],
  ["11111"]];
var coluna1 = [ ["0_","_0","_00","_000","_0000"],
  ["_1","1_","_01","_10","10_","1_0"],
  ["_11","1_1","11_"],
  [],
  []];
var coluna2 = [ ["0__","_0_","__0"],
  ["__1","_1_","1__"],
  [],
  []];
var coluna3 = [ [],
  [],
  []];
var coluna4 = [ [],
  []];
```

```
var col0 = [[1,2,1],
  [1,3,3,1],
  [1,4,6,4,1],
  [1,5,10,10,5,1]];
```

```
var col1 = [[2,2],
  [3,6,3],
  [4,12,12,4],
  [5,20,30,20,5]];
```

```
var col2 = [[],
  [3,3],
  [6,12,6],
  [10,30,30,10]];
```

```
var col3 = [[],
  [],
  [4,4],
  [10,20,10]];
```

```
var col4 = [[],
  [],
  [],
  [5,5]];
```

análise da tabela de comparações

número de grupos = $nVar + 1$

número de colunas = $nVar$

tabela de cobertura

a	b	c	d	1	2	5	6	7	8	9	10	
\bar{a}	0	0	\bar{d}	•					•	•		0,1,8,9
\bar{a}	0	\bar{c}	0		•				•		•	0,2,8,10
\bar{a}	\bar{c}	0	1	•		•				•		1,5,9,13
\bar{a}	1	\bar{c}	1			•		•				5,7,13,15
0	\bar{b}	1	0		•		•					2,6
0	1	1	\bar{d}				•	•				6,7
				2	2	2	2	2	2	2	1	

a	b	c	d	1	2	5	6	7	8	9	10	
\bar{a}	0	0	\bar{d}	•					•	•		0,1,8,9
\bar{a}	0	\bar{c}	0		•				•		•	0,2,8,10
\bar{a}	\bar{c}	0	1	•		•				•		1,5,9,13
\bar{a}	1	\bar{c}	1			•		•				5,7,13,15
0	\bar{b}	1	0		•		•					2,6
0	1	1	\bar{d}				•	•				6,7
				2	2	2	2	2	2	2	1	

a	b	c	d	1	5	6	7	9	
\bar{a}	\bar{b}	0	1	•	•			•	1,5,9,13
\bar{a}	1	\bar{c}	1		•		•		5,7,13,15
0	1	1	\bar{d}			•	•		6,7
				1	2	1	2	1	

a	b	c	d	1	5	6	7	9	
\bar{a}	\bar{b}	0	1	•	•		•		1,5,9,13
\bar{a}	1	\bar{c}	1		•		•		5,7,13,15
0	1	1	\bar{d}			•	•		6,7
				1	2	1	2	1	

arrays de objetos

>> resultado

```
← ▾ [...]
  ▶ 0: Object { coluna: 0, grupo: [...] }
  ▶ 1: Object { coluna: 1, grupo: [...] }
  ▾ 2: {}
    coluna: 2
    ▾ grupo: [...]
      ▶ 0: Array [ {} ]
      ▶ 1: Array [ {}, {}, {} ]
      ▶ 2: Array [ {}, {}, {}, ... ]
      ▾ 3: [...]
        ▶ 0: Object { binario: "1__11", implicante: 1, posicao: "19 27 23 31" }
        ▶ 1: Object { binario: "1_1_1", implicante: 2, posicao: "21 29 23 31" }
        ▶ 2: Object { binario: "1_11_", implicante: 1, posicao: "22 30 23 31" }
        ▶ 3: Object { binario: "11__1", implicante: 1, posicao: "25 29 27 31" }
        ▶ 4: Object { binario: "111__", implicante: 1, posicao: "28 30 29 31" }
        length: 5
        ▶ __proto__: Array []
      ▶ 4: Array []
      ▶ 5: Array []
        length: 6
        ▶ __proto__: Array []
      ▶ __proto__: Object { ... }
  ▶ 3: Object { coluna: 3, grupo: [...] }
  ▶ 4: Object { coluna: 4, grupo: [...] }
    length: 5
  ▶ __proto__: Array []
```

>> cobertura2

```
← ▾ [...]
  ▶ 0: Object { binario: "_0000", posicao: "0 16", validos: [...] }
  ▶ 1: Object { binario: "0_0_0", posicao: "0 8 2 10", validos: [...] }
  ▶ 2: Object { binario: "0__10", posicao: "2 10 6 14", validos: [...] }
  ▶ 3: Object { binario: "01__0", posicao: "8 12 10 14", validos: [...] }
  ▶ 4: Object { binario: "10_0_", posicao: "16 20 17 21", validos: [...] }
  ▶ 5: Object { binario: "_01_1", posicao: "5 21 7 23", validos: [...] }
  ▶ 6: Object { binario: "__101", posicao: "5 21 13 29", validos: [...] }
  ▶ 7: Object { binario: "_011_", posicao: "6 22 7 23", validos: [...] }
  ▶ 8: Object { binario: "__110", posicao: "6 22 14 30", validos: [...] }
  ▶ 9: Object { binario: "_110_", posicao: "12 28 13 29", validos: [...] }
  ▶ 10: Object { binario: "__11_0", posicao: "12 28 14 30", validos: [...] }
  ▶ 11: Object { binario: "1___1", posicao: "17 25 21 29 19 27 23 31", validos: [...] }
  ▶ 12: Object { binario: "1_1__", posicao: "20 28 22 30 21 29 23 31", validos: [...] }
    length: 13
  ▶ __proto__: Array []
```

evolução

10/11 inception

tema:
pesquisa

recapitular código anterior
verificar possibilidade
de utilizar grafos/árvores
nas comparações
verificar estrutura dos grupos
nas comparações
verificar estrutura das colunas
nas comparações
verificar possibilidade de utilizar
mais de 8 variáveis

15/11 release: 0

tema:
algoritmo

implementar funções e variáveis

implementar algoritmos de
comparação e alternativo

testes de integridade e
desempenho

18/11 release: 1

tema:
interface

interface base

adicionar mapas

adicionar tabelas

22/11 release: 2

tema:
entradas e saídas

importar
mapas

importar
tabelas

exportar
resultado

29/11 release: 3

tema:
extras

circuito

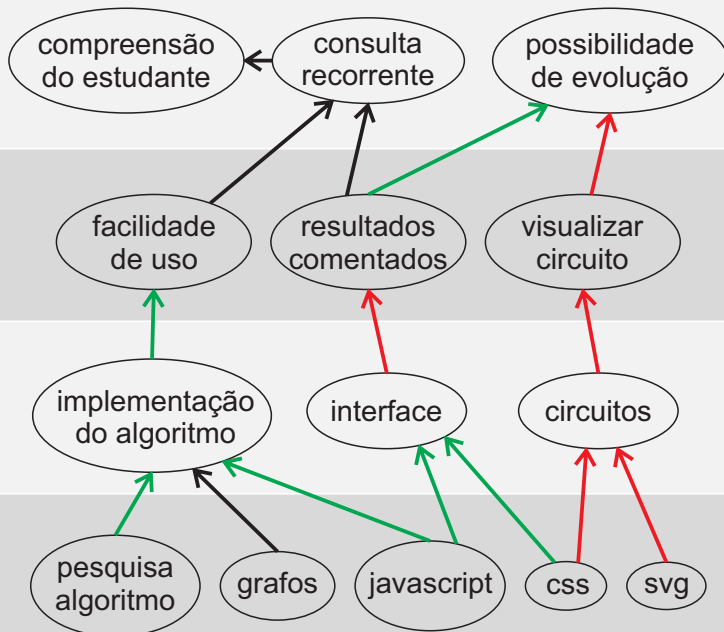
máquina
de
estados

05/12 release: 4

OBJETIVOS ESTRATÉGICOS E MAPA ESTRATÉGICO

INDICADORES

METAS



- pedidos de mais recursos
- reconhecimento por autoridades

- 5 pedidos de evolução
- 2 reconhecimentos de professores

- recorrência na semana
- permanência (site)
- comentários

- utilização duas vezes por semana
- 30 minutos
- comentários positivos e construtivos

- velocidade de implementação
- velocidade de testes

- uma funcionalidade por iteração
- dentro da mesma iteração

- aprendizado
- custo da implementação (recursos)

- definição de conceitos e lógica
- a ponto de não atrapalhar a experiência