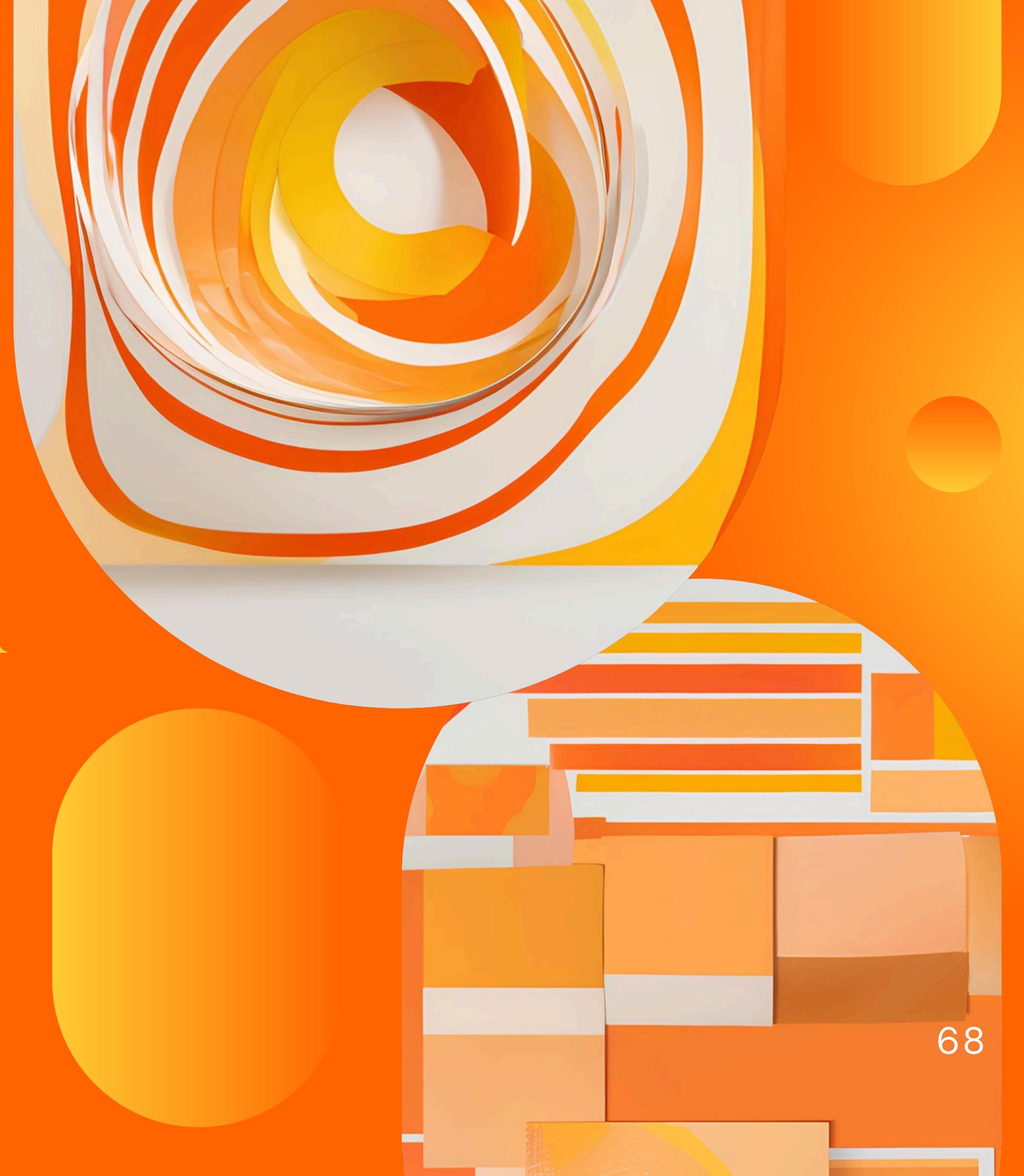
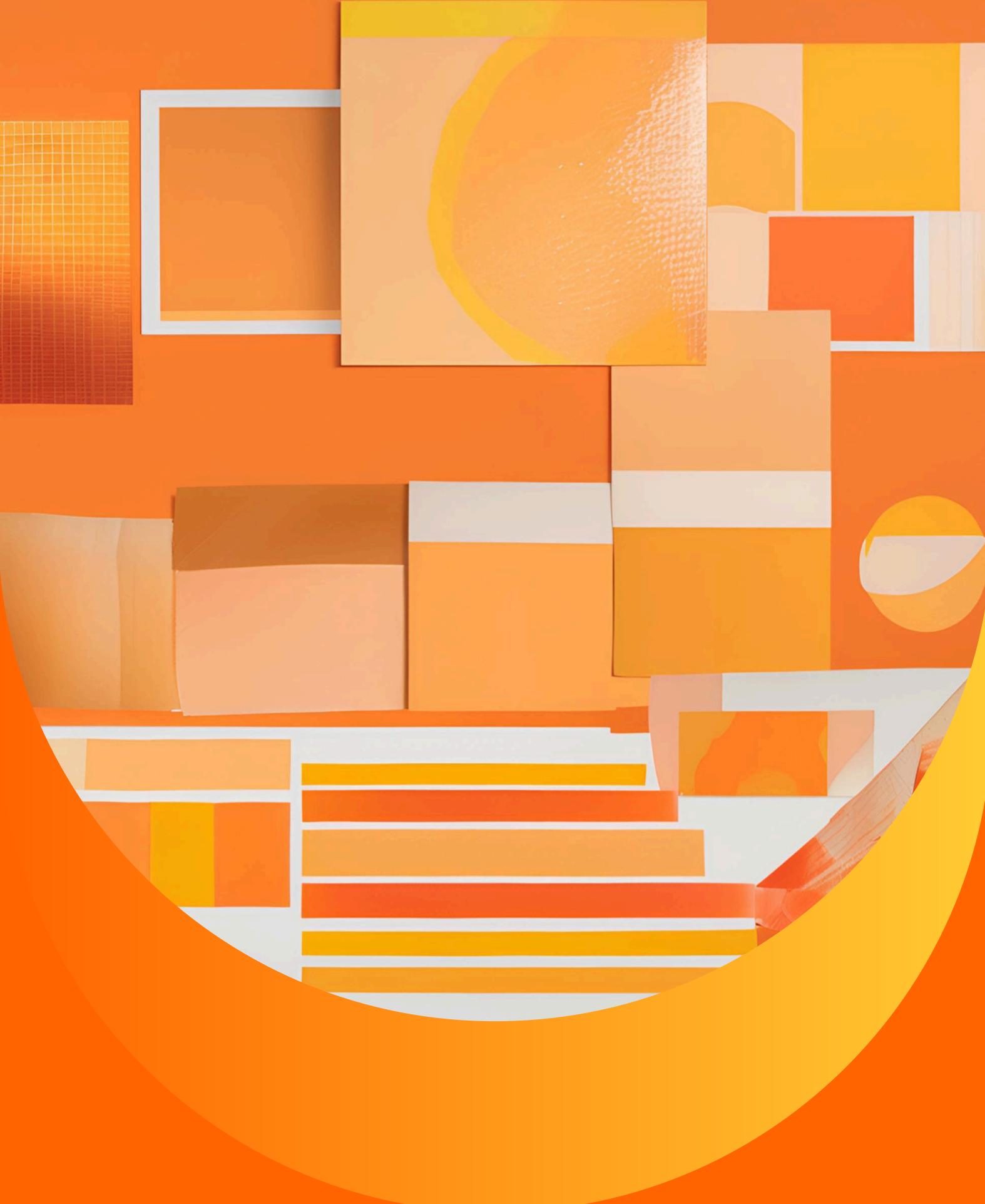


# TESTES CAIXA BRANCA

Análise da lógica da rota de  
cadastro de contatos





# CAIXA BRANCA

## O QUE É?

A técnica de teste de caixa branca consiste na análise do fluxo interno do código-fonte. Ela permite que o desenvolvedor/testador avalie os caminhos de execução, decisões lógicas e o uso de variáveis com base nas entradas fornecidas.



## POR QUE USAMOS?

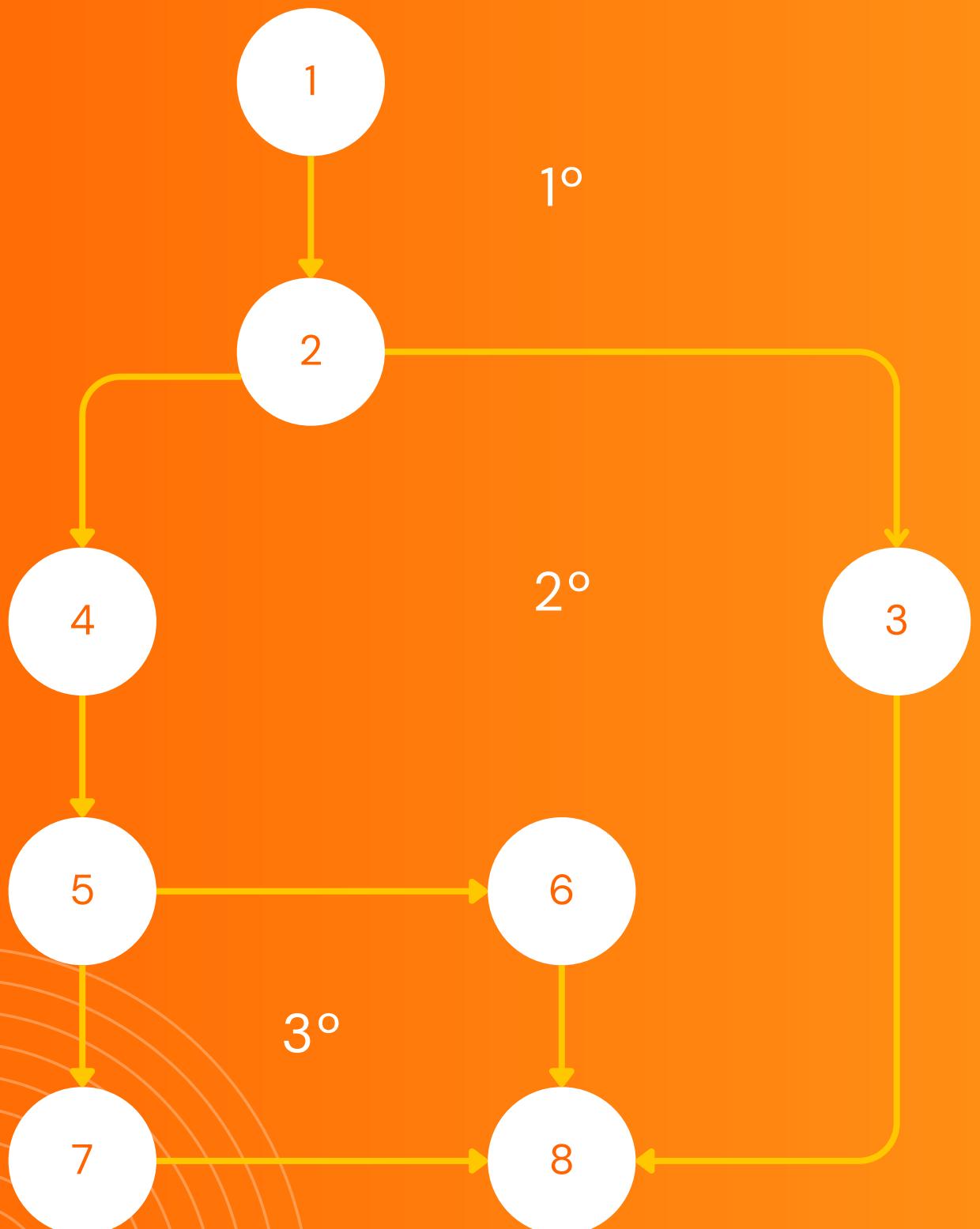
No projeto "Agenda de Contatos", essa técnica foi aplicada para garantir que as funcionalidades implementadas estão lógica e tecnicamente corretas, mesmo antes da interface.



# Cadastrar Contato



# Grafo



```
1° app.post('/cadastrar-contato', (req, res) => {  
2°   const { nome, email, telefone, usuario_id } = req.body;  
3°   if (!nome || !email || !telefone || !usuario_id) {  
4°     return res.status(400).send('Todos os campos são  
5°     obrigatórios.');//  
6°   }  
7°   const sql = "INSERT INTO tb_contatos (nome, email,  
8°     telefone, usuario_id) VALUES (?, ?, ?, ?);  
9°   db.query(sql, [nome, email, telefone, usuario_id], (err, result)  
10°  => {  
11°    if (err) {  
12°      return res.status(500).send('Erro ao cadastrar contato.');//  
13°    }  
14°    return res.status(201).send('Contato cadastrado com  
15°    sucesso!');//  
16°  });//  
17°});
```



# CALCULO DA COMPLEXIDADE CICLOMÁTICA

$$F1: M(G) = NR$$

Resultado:  $M(G) = 3$

---

$$F2: M(G) = NP + 1$$

Resultado:  $M(G) = 2 + 1 = 3$

---

$$F3 M(G) = E - N + 2$$

Resultado:  $M(G) = 9 - 8 + 2 = 3$

## Tabela nós

Nº	1	2	3	4	5	6	7	8
Nós	1	2	3	4	5	6	7	8

## Tabela todos os arcos

Nº	1	2	3	4	5	6	7	8	9
Arcos	(1,2)	(2,3)	(2,4)	(3,8)	(4,5)	(5,6)	(5,7)	(6,8)	(7,8)

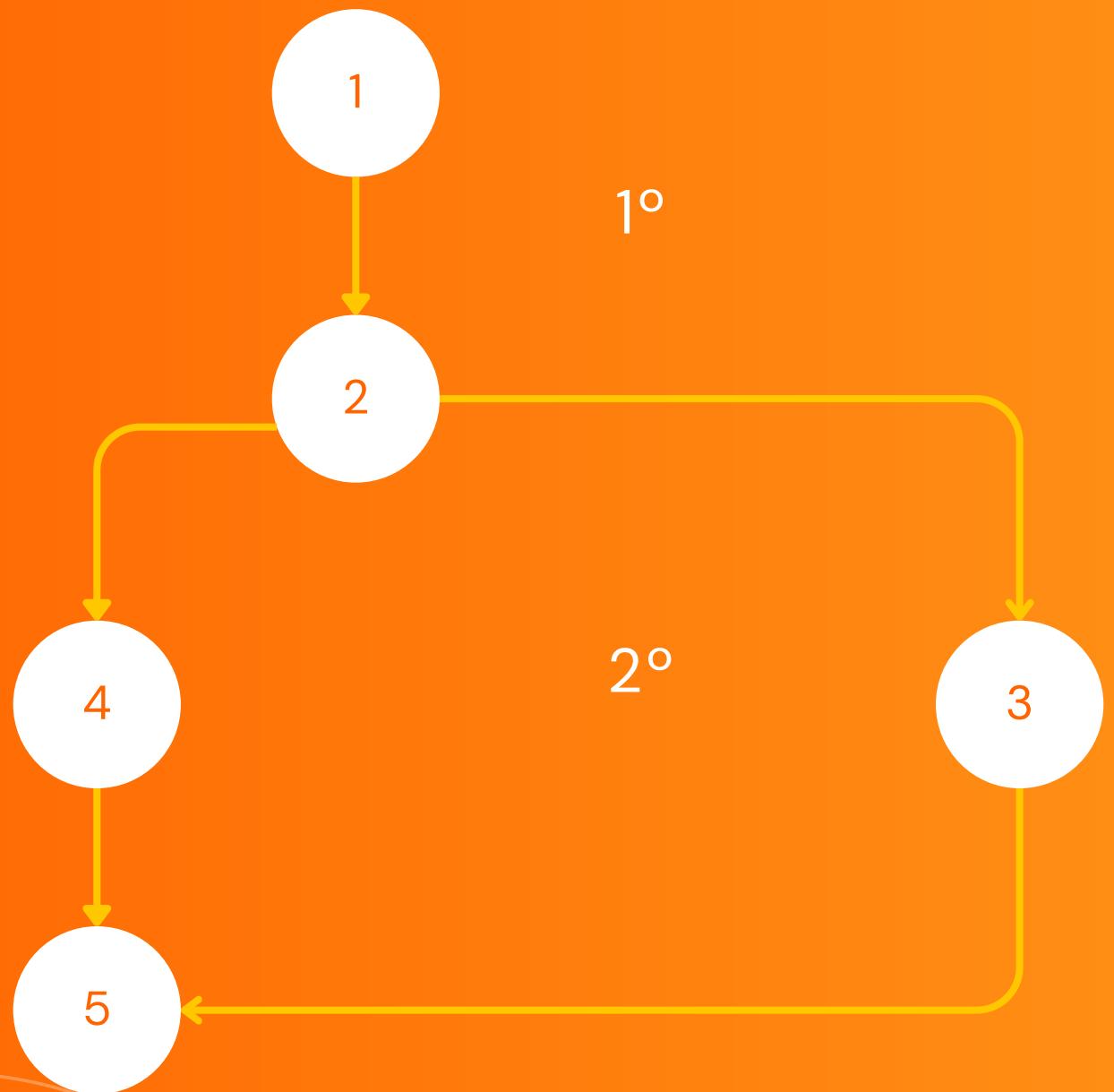
## Tabela todos os caminhos independentes

Nº	1	2	3
Caminhos Independentes	$C_1 = 1, 2, 4, 5, 7, 8$	$C_2 = 1, 2, 3, 8$	$C_3 = 1, 2, 4, 5, 6, 8$

# Lista Contatos



# Grafo



```
1° app.get('/contatos', (req, res) => {  
2°   db.query('SELECT * FROM tb_contatos', (err, results) => {  
3°     if (err) {  
4°       return res.status(500).send(err);  
5°     }  
6°     res.json(results);  
7°   });  
8°});  
9°});
```



# CALCULO DA COMPLEXIDADE CICLOMÁTICA

$$F1: M(G) = NR$$

Resultado:  $M(G) = 2$

---

$$F2: M(G) = NP + 1$$

Resultado:  $M(G) = 1 + 1 = 2$

---

$$F3 M(G) = E - N + 2$$

Resultado:  $M(G) = 5 - 5 + 2 = 2$

## Tabela nós

Nº	1	2	3	4	5
Nós	1	2	3	4	5

## Tabela todos os arcos

Nº	1	2	3	4	5
Arcos	(1,2)	(2,3)	(2,4)	(3,5)	(4,5)

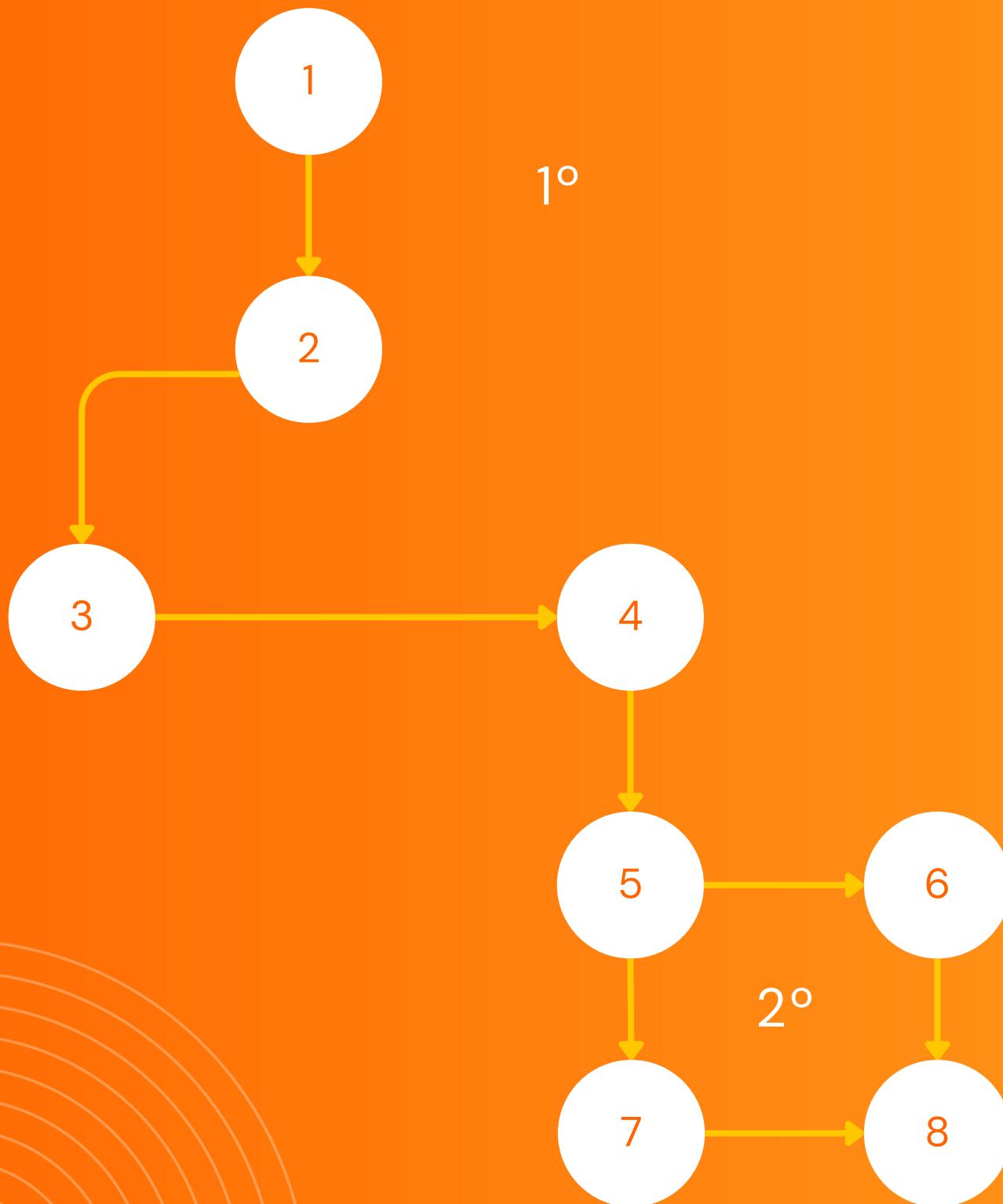
## Tabela todos os arcos

Nº	1	2
Caminhos Independentes	C1 = 1,2,4,5	C2 = 1,2,3,5

# Atualizar Contato



# Grafo



```
1° app.put('/contatos/:id', (req, res) => {  
2°   const { id } = req.params;  
3°   const { nome, email, telefone } = req.body;  
4°   const sql = 'UPDATE tb_contatos SET nome = ?, email = ?,  
      telefone = ?, atualizado_em = NOW() WHERE id = ?';  
5°   db.query(sql, [nome, email, telefone, id], (err, result) => {  
6°     if (err) {  
       console.error('Erro ao atualizar contato:', err);  
       return res.status(500).send('Erro ao atualizar contato');  
     }  
7°     res.send('Contato atualizado com sucesso!');  
8°   });  
});
```



# CALCULO DA COMPLEXIDADE CICLOMÁTICA

$$F1: M(G) = NR$$

Resultado:  $M(G) = 2$

---

$$F2: M(G) = NP + 1$$

Resultado:  $M(G) = 1 + 1 = 2$

---

$$F3 M(G) = E - N + 2$$

Resultado:  $M(G) = 8 - 8 + 2 = 2$

## Tabela nós

Nº	1	2	3	4	5	6	7	8
Nós	1	2	3	4	5	6	7	8

## Tabela todos os arcos

Nº	1	2	3	4	5	6	7	8
Arcos	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)	(5,7)	(6,8)	(7,8)

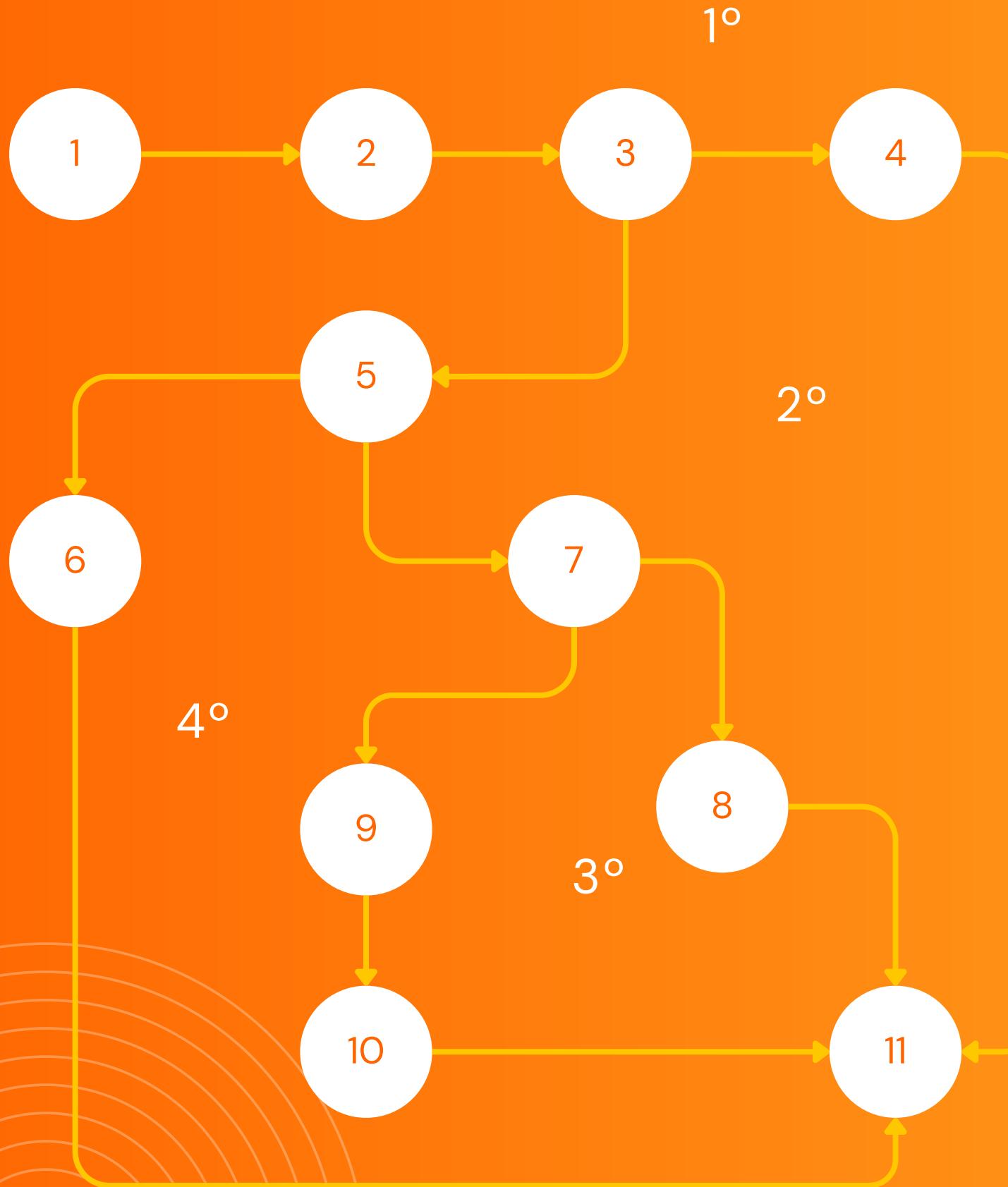
## Tabela todos os caminhos independentes

Nº	1	2	81
Caminhos Independentes	C1 = 1,2,3,4,5,7,8	C2 = 1,2,3,4,5,6,8	

# Deletar Contato



# Grafo



```
1° app.delete('/contatos/:id', (req, res) => {  
2°   const { id } = req.params;  
3°   db.query('DELETE FROM tb_contatos WHERE id = ?', [id], (err, result) => {  
4°     if (err) {  
5°       console.error('Erro ao excluir contato:', err);  
6°       return res.status(500).send('Erro ao excluir contato');  
7°     }  
8°     db.query('INSERT INTO tb_contatos_excluidos (contato_id) VALUES (?)',  
9°     [id], (err2) => {  
10°       if (err2) {  
11°         console.error('Erro ao registrar exclusão:', err2);  
12°         return res.status(500).send('Erro ao registrar exclusão');  
13°       }  
14°       db.query('SELECT COUNT(*) AS totalExcluidos FROM  
15°       tb_contatos_excluidos', (err3, result3) => {  
16°         if (err3) {  
17°           console.error('Erro ao contar excluídos:', err3);  
18°           return res.status(500).send('Erro ao contar contatos excluídos');  
19°         }  
20°         const totalExcluidos = result3[0].totalExcluidos;  
21°         res.json({ mensagem: 'Contato excluído com sucesso!', totalExcluidos });  
22°       });  
23°     });  
24°   });  
25° });  
26°});  
27°});  
28°});  
29°});  
30°});  
31°});  
32°});  
33°});  
34°});  
35°});  
36°});  
37°});  
38°});  
39°});  
40°});  
41°});  
42°});  
43°});  
44°});  
45°});  
46°});  
47°});  
48°});  
49°});  
50°});  
51°});  
52°});  
53°});  
54°});  
55°});  
56°});  
57°});  
58°});  
59°});  
60°});  
61°});  
62°});  
63°});  
64°});  
65°});  
66°});  
67°});  
68°});  
69°});  
70°});  
71°});  
72°});  
73°});  
74°});  
75°});  
76°});  
77°});  
78°});  
79°});  
80°});  
81°});  
82°});  
83°});  
84°});  
85°});  
86°});  
87°});  
88°});  
89°});  
90°});  
91°});  
92°});  
93°});  
94°});  
95°});  
96°});  
97°});  
98°});  
99°});  
100°});  
101°});  
102°});  
103°});  
104°});  
105°});  
106°});  
107°});  
108°});  
109°});  
110°});  
111°});  
112°});  
113°});  
114°});  
115°});  
116°});  
117°});  
118°});  
119°});  
120°});  
121°});  
122°});  
123°});  
124°});  
125°});  
126°});  
127°});  
128°});  
129°});  
130°});  
131°});  
132°});  
133°});  
134°});  
135°});  
136°});  
137°});  
138°});  
139°});  
140°});  
141°});  
142°});  
143°});  
144°});  
145°});  
146°});  
147°});  
148°});  
149°});  
150°});  
151°});  
152°});  
153°});  
154°});  
155°});  
156°});  
157°});  
158°});  
159°});  
160°});  
161°});  
162°});  
163°});  
164°});  
165°});  
166°});  
167°});  
168°});  
169°});  
170°});  
171°});  
172°});  
173°});  
174°});  
175°});  
176°});  
177°});  
178°});  
179°});  
180°});  
181°});  
182°});  
183°});  
184°});  
185°});  
186°});  
187°});  
188°});  
189°});  
190°});  
191°});  
192°});  
193°});  
194°});  
195°});  
196°});  
197°});  
198°});  
199°});  
200°});  
201°});  
202°});  
203°});  
204°});  
205°});  
206°});  
207°});  
208°});  
209°});  
210°});  
211°});  
212°});  
213°});  
214°});  
215°});  
216°});  
217°});  
218°});  
219°});  
220°});  
221°});  
222°});  
223°});  
224°});  
225°});  
226°});  
227°});  
228°});  
229°});  
230°});  
231°});  
232°});  
233°});  
234°});  
235°});  
236°});  
237°});  
238°});  
239°});  
240°});  
241°});  
242°});  
243°});  
244°});  
245°});  
246°});  
247°});  
248°});  
249°});  
250°});  
251°});  
252°});  
253°});  
254°});  
255°});  
256°});  
257°});  
258°});  
259°});  
260°});  
261°});  
262°});  
263°});  
264°});  
265°});  
266°});  
267°});  
268°});  
269°});  
270°});  
271°});  
272°});  
273°});  
274°});  
275°});  
276°});  
277°});  
278°});  
279°});  
280°});  
281°});  
282°});  
283°});  
284°});  
285°});  
286°});  
287°});  
288°});  
289°});  
290°});  
291°});  
292°});  
293°});  
294°});  
295°});  
296°});  
297°});  
298°});  
299°});  
300°});  
301°});  
302°});  
303°});  
304°});  
305°});  
306°});  
307°});  
308°});  
309°});  
310°});  
311°});  
312°});  
313°});  
314°});  
315°});  
316°});  
317°});  
318°});  
319°});  
320°});  
321°});  
322°});  
323°});  
324°});  
325°});  
326°});  
327°});  
328°});  
329°});  
330°});  
331°});  
332°});  
333°});  
334°});  
335°});  
336°});  
337°});  
338°});  
339°});  
340°});  
341°});  
342°});  
343°});  
344°});  
345°});  
346°});  
347°});  
348°});  
349°});  
350°});  
351°});  
352°});  
353°});  
354°});  
355°});  
356°});  
357°});  
358°});  
359°});  
360°});  
361°});  
362°});  
363°});  
364°});  
365°});  
366°});  
367°});  
368°});  
369°});  
370°});  
371°});  
372°});  
373°});  
374°});  
375°});  
376°});  
377°});  
378°});  
379°});  
380°});  
381°});  
382°});  
383°});  
384°});  
385°});  
386°});  
387°});  
388°});  
389°});  
390°});  
391°});  
392°});  
393°});  
394°});  
395°});  
396°});  
397°});  
398°});  
399°});  
400°});  
401°});  
402°});  
403°});  
404°});  
405°});  
406°});  
407°});  
408°});  
409°});  
410°});  
411°});  
412°});  
413°});  
414°});  
415°});  
416°});  
417°});  
418°});  
419°});  
420°});  
421°});  
422°});  
423°});  
424°});  
425°});  
426°});  
427°});  
428°});  
429°});  
430°});  
431°});  
432°});  
433°});  
434°});  
435°});  
436°});  
437°});  
438°});  
439°});  
440°});  
441°});  
442°});  
443°});  
444°});  
445°});  
446°});  
447°});  
448°});  
449°});  
450°});  
451°});  
452°});  
453°});  
454°});  
455°});  
456°});  
457°});  
458°});  
459°});  
460°});  
461°});  
462°});  
463°});  
464°});  
465°});  
466°});  
467°});  
468°});  
469°});  
470°});  
471°});  
472°});  
473°});  
474°});  
475°});  
476°});  
477°});  
478°});  
479°});  
480°});  
481°});  
482°});  
483°});  
484°});  
485°});  
486°});  
487°});  
488°});  
489°});  
490°});  
491°});  
492°});  
493°});  
494°});  
495°});  
496°});  
497°});  
498°});  
499°});  
500°});  
501°});  
502°});  
503°});  
504°});  
505°});  
506°});  
507°});  
508°});  
509°});  
510°});  
511°});  
512°});  
513°});  
514°});  
515°});  
516°});  
517°});  
518°});  
519°});  
520°});  
521°});  
522°});  
523°});  
524°});  
525°});  
526°});  
527°});  
528°});  
529°});  
530°});  
531°});  
532°});  
533°});  
534°});  
535°});  
536°});  
537°});  
538°});  
539°});  
540°});  
541°});  
542°});  
543°});  
544°});  
545°});  
546°});  
547°});  
548°});  
549°});  
550°});  
551°});  
552°});  
553°});  
554°});  
555°});  
556°});  
557°});  
558°});  
559°});  
560°});  
561°});  
562°});  
563°});  
564°});  
565°});  
566°});  
567°});  
568°});  
569°});  
570°});  
571°});  
572°});  
573°});  
574°});  
575°});  
576°});  
577°});  
578°});  
579°});  
580°});  
581°});  
582°});  
583°});  
584°});  
585°});  
586°});  
587°});  
588°});  
589°});  
590°});  
591°});  
592°});  
593°});  
594°});  
595°});  
596°});  
597°});  
598°});  
599°});  
600°});  
601°});  
602°});  
603°});  
604°});  
605°});  
606°});  
607°});  
608°});  
609°});  
610°});  
611°});  
612°});  
613°});  
614°});  
615°});  
616°});  
617°});  
618°});  
619°});  
620°});  
621°});  
622°});  
623°});  
624°});  
625°});  
626°});  
627°});  
628°});  
629°});  
630°});  
631°});  
632°});  
633°});  
634°});  
635°});  
636°});  
637°});  
638°});  
639°});  
640°});  
641°});  
642°});  
643°});  
644°});  
645°});  
646°});  
647°});  
648°});  
649°});  
650°});  
651°});  
652°});  
653°});  
654°});  
655°});  
656°});  
657°});  
658°});  
659°});  
660°});  
661°});  
662°});  
663°});  
664°});  
665°});  
666°});  
667°});  
668°});  
669°});  
670°});  
671°});  
672°});  
673°});  
674°});  
675°});  
676°});  
677°});  
678°});  
679°});  
680°});  
681°});  
682°});  
683°});  
684°});  
685°});  
686°});  
687°});  
688°});  
689°});  
690°});  
691°});  
692°});  
693°});  
694°});  
695°});  
696°});  
697°});  
698°});  
699°});  
700°});  
701°});  
702°});  
703°});  
704°});  
705°});  
706°});  
707°});  
708°});  
709°});  
710°});  
711°});  
712°});  
713°});  
714°});  
715°});  
716°});  
717°});  
718°});  
719°});  
720°});  
721°});  
722°});  
723°});  
724°});  
725°});  
726°});  
727°});  
728°});  
729°});  
730°});  
731°});  
732°});  
733°});  
734°});  
735°});  
736°});  
737°});  
738°});  
739°});  
740°});  
741°});  
742°});  
743°});  
744°});  
745°});  
746°});  
747°});  
748°});  
749°});  
750°});  
751°});  
752°});  
753°});  
754°});  
755°});  
756°});  
757°});  
758°});  
759°});  
760°});  
761°});  
762°});  
763°});  
764°});  
765°});  
766°});  
767°});  
768°});  
769°});  
770°});  
771°});  
772°});  
773°});  
774°});  
775°});  
776°});  
777°});  
778°});  
779°});  
770°});  
771°});  
772°});  
773°});  
774°});  
775°});  
776°});  
777°});  
778°});  
779°});  
780°});  
781°});  
782°});  
783°});  
784°});  
785°});  
786°});  
787°});  
788°});  
789°});  
790°});  
791°});  
792°});  
793°});  
794°});  
795°});  
796°});  
797°});  
798°});  
799°});  
800°});  
801°});  
802°});  
803°});  
804°});  
805°});  
806°});  
807°});  
808°});  
809°});  
8010°});  
8011°});  
8012°});  
8013°});  
8014°});  
8015°});  
8016°});  
8017°});  
8018°});  
8019°});  
8020°});  
8021°});  
8022°});  
8023°});  
8024°});  
8025°});  
8026°});  
8027°});  
8028°});  
8029°});  
8030°});  
8031°});  
8032°});  
8033°});  
8034°});  
8035°});  
8036°});  
8037°});  
8038°});  
8039°});  
8040°});  
8041°});  
8042°});  
8043°});  
8044°});  
8045°});  
8046°});  
8047°});  
8048°});  
8049°});  
8050°});  
8051°});  
8052°});  
8053°});  
8054°});  
8055°});  
8056°});  
8057°});  
8058°});  
8059°});  
8060°});  
8061°});  
8062°});  
8063°});  
8064°});  
8065°});  
8066°});  
8067°});  
8068°});  
8069°});  
8070°});  
8071°});  
8072°});  
8073°});  
8074°});  
8075°});  
807
```



# CALCULO DA COMPLEXIDADE CICLOMÁTICA

$$F1: M(G) = NR$$

Resultado:  $M(G) = 4$

---

$$F2: M(G) = NP + 1$$

Resultado:  $M(G) = 3 + 1 = 4$

---

$$F3 M(G) = E - N + 2$$

Resultado:  $M(G) = 13 - 11 + 2 = 4$

## Tabela nós

Nº	1	2	3	4	5	6	7	8	9	10	11
Nós	1	2	3	4	5	6	7	8	9	10	11

## Tabela todos os arcos

Nº	1	2	3	4	5	6	7	8	9	10	11	12	13
Arcos	(1,2)	(2,3)	(3,4)	(3,5)	(4,11)	(5,6)	(5,7)	(6,11)	(7,8)	(7,9)	(8,11)	(9,10)	(10,11)

## Tabela todos os caminhos

Nº	1	2	3	4
Caminhos Independentes	$C_1 = 1,2,3,4,11$	$C_2 = 1,2,3,5,6,11$	$C_3 = 1,2,3,5,7,8,11$	$C_4 = 1,2,3,5,7,9,10,11$