

O Formigueiro em imagens

Guia visual da arquitetura: como a demanda circula entre as colônias, como o conhecimento se acumula e como os textos viram resumos.

O que este guia responde

- **Páginas 2 e 3:** o caminho de uma demanda, colônia por colônia, e o que cada uma processa.
- **Página 4:** o fluxo do conhecimento agregado, de baixo para cima.
- **Páginas 5 a 7:** a concatenação explicada: do texto ao campo, o algoritmo por dentro e o LanguageTool.
- **Páginas 8 a 10:** os resumos micro e macro e o dossiê dos 10 buracos com a demanda 11.
- **Páginas 11 a 13:** dúvidas frequentes, glossário e mapa de eventos.

A ideia em uma frase

Um ciclo concluído vira um caso. Os casos iguais se somam em um perfil. O perfil vira um dossiê explicado. A próxima demanda nasce com esse caminho na mão.

Como ler os diagramas

- **Caixa** = colônia (um módulo com estado próprio).
- **Seta** = evento publicado no barramento.
- **Caixa verde** = o que o público vê.
- **Caixa tracejada** = peça de qualidade ou fase futura.

Cores: azul é a trilha da demanda, âmbar é o conhecimento, verde é a saída pública, cinza é o núcleo, roxo são os fluxos paralelos de lugar e empresa.

Princípios que valem em todas as páginas

- **IA não decide.** Ela sugere, resume e sinaliza. A decisão é humana.
- **Nada se perde.** O log de eventos é append-only e serve de fonte para reconstruir tudo.
- **Colônias não se conhecem.** Elas conversam só por eventos. Nenhuma lê o banco da outra.
- **Público por padrão.** O que sai é sanitizado antes de virar projeção.
- **Todo automático é marcado.** Com versão de método, data e fontes.

Os números e textos de exemplo deste guia são fictícios e servem para explicar o mecanismo. Os nomes de colônias, eventos e campos são os reais do sistema.

Fluxo da demanda entre as colônias

Cada caixa é um módulo independente. As setas são eventos no barramento. Cada colônia mostra o que publica.

Sequência principal Saída pública Núcleo Módulo de apoio / paralelo



Anatomia de uma colônia e o caminho de um evento

Entender as quatro partes de qualquer colônia é o que permite ler todos os diagramas deste guia.

Uma colônia, quatro partes

- 1

Consome eventos

Assina tipos no barramento, guarda um cursor por tipo e faz replay do que perdeu no boot.
- 2

Estado próprio

Tabelas no schema da colônia, por exemplo `d24.*`. Nenhuma outra colônia escreve ali.
- 3

Publica eventos

A saída vai para o log append-only. É assim que o resto do sistema fica sabendo do que aconteceu.
- 4

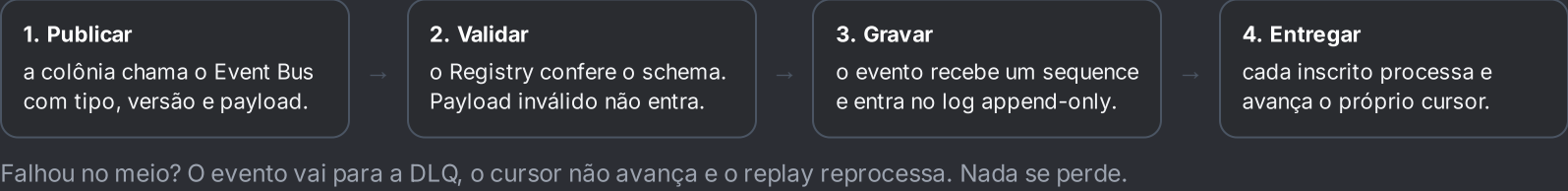
Projeção de leitura

Para mostrar algo, mantém uma cópia local de leitura. Nunca consulta o schema da vizinha.

As quatro regras duras

- Colônia não importa código de outra colônia.
- Colônia não escreve no banco de outra.
- Toda comunicação é por evento. Nada de chamada direta.
- A única dependência permitida é o núcleo: Event Bus, Registry e Observabilidade.

O caminho de um evento



Exemplo real de payload

Evento `demanda.normalizada`, publicado pela D-1b e consumido por D-2, D-3, D-7 e D-12.

```
{
  "demanda_id": "a1b2c3d4- ... ",
  "titulo": "Falta de água na Rua das Flores",
  "descricao_limpa": "Falta d'água na Rua das Flores",
  "confianca_normalizacao": 0.95
}
```

Por que isso importa para o conhecimento: como cada colônia guarda o próprio estado e publica o que fez, qualquer projeção pode ser reconstruída a partir do log. O histórico é a fonte, e o presente é derivado dele.

Como nasce o conhecimento agregado

Leitura de baixo para cima: o ciclo concluído vira caso, o caso vira caminho, o caminho acelera a próxima demanda.



Conhecimento (D-24)



Saída pública



Entrada e projeção



Fase futura / qualidade

Etapa 5 · Projeção pública (D-7)

O acompanhamento e o relatório exportado mostram a mesma visão completa: resumo do agregado, resumo do ciclo e dossiê do caminho.

Etapa 4 · Resumo macro

A D-12 concatena todos os relatos do agregado em um resumo público.

Etapa 4 · Resumo micro

A D-6b concatena o ciclo completo, atualização por atualização, até o desfecho.

Etapa 3 · Perfil do caminho

Contagem de casos, canais que funcionaram, órgãos acionados, documentos pedidos, prazos medianos, gargalos recorrentes e taxa de resolução. Gera o dossiê explicado, versionado como **caminho_v1**.

Etapa 2 · D-24 Memória de Caminhos

Extrai o caminho real de resolução e guarda um caso por ciclo concluído. Agrega por município e subcategoria (ex.: buracos na via no município).

canal

órgão

protocolo

documentos

prazos

gargalos

desfecho

Etapa 1 · Entradas

Ciclos concluídos: contato, protocolo, documento, entrave, prazo e desfecho. Conclusões e agregados de duplicidade entram junto.

demanda.georreferenciada

demanda.categorizada

conselheiro.atualização_publicada

conselheiro.ciclo_concluído

demanda.concluída

duplicidade.agregada

Peças de qualidade

Assistência de texto (LanguageTool) e tradução da legenda (Florence). Entram nos inputs, fora do ciclo central.

Futuro · VPS de IA

Modelo de linguagem, embeddings e busca semântica para o dossiê narrativo.

Etapa 6 · Consumo

A 11ª demanda da mesma subcategoria nasce com o caminho das 10 anteriores. O conselheiro vê o micro completo e o macro como contexto.

Etapa 7 · Ciclo de feedback

O caminho real registrado na 11ª demanda vira um novo caso e atualiza o perfil.

↶ retorna à Etapa 2

Do texto ao campo: como um relato vira dado

O sistema não cola textos soltos. Ele transforma cada atualização em campos tipados e é desses campos que nasce o resumo.

1. O conselheiro escreve

"Liguei hoje para a Secretaria de Obras e abri o protocolo SO-2026/4127. A vistoria ficou para a semana que vem."

Texto livre, em linguagem natural. É assim que a atualização chega pelo app.

Relato em áudio segue o mesmo caminho: a D-1b transcreve e a atualização é estruturada do mesmo jeito.

2. A D-6b estrutura

Ao publicar, o conselheiro escolhe o tipo da atualização. O sistema valida os campos obrigatórios daquele tipo.

TIPO	CAMPOS EXTRAÍDOS
contato_realizado	canal = telefone data_contato = 12/03
protocolo_aberto	órgão = Secretaria de Obras protocolo = SO-2026/4127
prazo_registrado	data_estimada = 20/04

3. A D-24 guarda o caso

O caso recebe os campos, não o texto inteiro:

canal = telefone	órgão = Secretaria de Obras	protocolo = SO-2026/4127
prazo = 38 dias	desfecho = concluída	

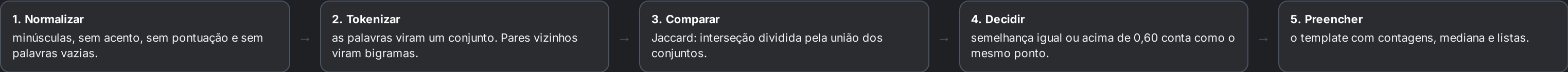
Telefone, nome de pessoa e outros dados pessoais são removidos na sanitização antes de qualquer persistência ou publicação.

O caso também guarda a data de abertura, o prazo e o desfecho. É isso que permite calcular a mediana da página 8.

Por que isso funciona sem modelo de linguagem: a estruturação já existe. A D-6b tem seis tipos de atualização com campos obrigatórios e valores permitidos. O canal não é adivinhado no texto: ele vem de uma lista fechada escolhida no app. A D-24 não interpreta texto livre: ela colhe o que já é dado tipado. É esse recorte que torna a concatenação determinística e auditável.

Por dentro do algoritmo: token, Jaccard e template

O que acontece entre o relato e o resumo, passo a passo. Nenhum passo entende o significado do texto.



Jaccard com números

CONJUNTO	TOKENS (EXEMPLO SIMPLIFICADO)
A	buraco · rua · flores · carro
B	buraco · rua · flores · fundo

Interseção: 3 tokens. União: 5 tokens.

3 ÷ 5 = 0,60. No limiar, os dois trechos são tratados como o mesmo ponto.

Os bigramas entram no mesmo cálculo. "buraco grande" e "grande buraco" têm as mesmas palavras, mas pares diferentes.

No código: `mvp-api · src/duplicidade/texto-similaridade.ts` (*normalizarTexto*, *tokenizar* e *similaridadeJaccard*). O limiar de 0,60 vive em `src/demanda/d-12-deteccao-duplicidade/d12.constants.ts`.

Contagem e mediana

Contar é agrupar: quantas vezes cada canal, órgão e gargalo aparece no conjunto de casos.

Mediana é ordenar e pegar o meio:

[22, 14, 12, 5, 45, 14, 6, 8, 18, 15]
→ [5, 6, 8, 12, 14, 14, 15, 18, 22, 45]
→ (14 + 14) ÷ 2 = 14

Com 10 valores (par), a mediana é a média dos dois do meio. O caso de 45 dias não puxa o resultado como puxaria a média.

Sem interpretação: em nenhum passo o sistema adivinha o que o texto quer dizer. Ele compara conjuntos, conta ocorrências e ordena números. É isso que torna o resultado determinístico e auditável.

LanguageTool: um motor de regras, não um modelo

O que ele é, o que a nossa API envia, o que volta e o que o usuário faz com isso.

O que é

- Motor de regras para ortografia, gramática, pontuação e estilo.
- Escrito em Java, open source (LGPL), com conjunto de regras pt-BR.
- Não gera texto e não aprende com o uso. Ele aponta o trecho e as alternativas.
- Roda em container próprio, sem dado saindo do servidor.
- Os n-gramas ficam desligados para economizar memória.

O que trafega

O endpoint recebe `{campo, texto}` e chama o LanguageTool com `language=pt-BR`.

O texto do campo vai inteiro para o motor. Nada é salvo: a operação é stateless e não bloqueia o envio do formulário.

O que volta

```
{
  "matches": [{
    "message": "Use a crase...",
    "offset": 12,
    "length": 7,
    "replacements": [ "... " ],
    "rule": { "id": "PT_CRASE" }
  }]
}
```

A API traduz para o contrato do app: trecho, mensagem, tipo e substituições. O usuário aplica uma, várias ou nenhuma.

Sanitização não é IA

Remover telefone, CPF e e-mail do texto é casamento de padrão (regex), não compreensão. Roda antes de persistir ou publicar.

O canal "telefone" do caso é outra coisa: vem de uma lista fechada escolhida no formulário, não de leitura automática do texto.

Resumindo: LanguageTool é um conjunto de regras com parâmetros. Ele não reescreve por conta própria e não decide nada.

Resumo micro: o ciclo vira um texto cronológico

A D-6b percorre as atualizações publicadas e monta uma linha por marco, até o desfecho. Método resumo_v1.

Entrada: as atualizações do ciclo

- 12/03 **Contato realizado.** Canal: telefone.
- 14/03 **Protocolo aberto.** SO-2026/4127, Secretaria de Obras.
- 28/03 **Entrave registrado.** Vistoria atrasou, gravidade média.
- 02/04 **Prazo registrado.** Estimativa para 20/04.
- 19/04 **Demanda concluída.** Resumo final do conselheiro.

Modelo de montagem

```
Ciclo de {dias} dias, {n} atualizações.  
{data} {tipo}: {trecho sanitizado}  
...  
Desfecho: {resumo_final}
```

Saída: o resumo publicado

```
Ciclo de 38 dias, 5 atualizações.  
12/03 Contato por telefone com o órgão responsável.  
14/03 Protocolo SO-2026/4127 aberto na Secretaria de Obras.  
28/03 Entrave de gravidade média: vistoria atrasou.  
02/04 Prazo estimado para 20/04.  
19/04 Demanda concluída.
```

Atualizado a cada publicação e fechado na conclusão. Limite de 6000 caracteres.

Resumo macro: vários relatos viram um só

A D-12 recebe os relatos do agregado, remove repetições e soma os detalhes diferentes. Método resumo_v1.

1. Três relatos do mesmo buraco

- 12/03

"Tem um buraco grande na Rua A em frente ao número 120, os carros estão desviando."
- 12/03

"Buraco na Rua A perto do número 120, muito fundo, já faz uma semana."
- 13/03

"Buraco na Rua A, o ônibus passa raspando, perto do 120."

2. Limpeza e comparação

- Texto normalizado: sem acento, minúsculas, pontuação removida.
- Comparação por palavras. Trechos com 60% ou mais de semelhança contam como o mesmo ponto.
- "buraco na rua a" se repete nos três e vira o ponto em comum.
- O que não se repete é preservado: profundidade, tempo, desvio de carros, ônibus.

A mesma heurística que a D-12 usa para detectar duplicatas.

3. Saída: o resumo do agregado

3 relatos registrados entre 12/03/2026 e 13/03/2026.

Ponto em comum: buraco na Rua A, próximo ao número 120.

Detalhes somados: profundidade, tempo de existência, desvio de veículos e prejuízo ao ônibus.

Limite de 3000 caracteres. Republicado quando o agregado ganha um membro novo.

O dossiê do caminho: os 10 buracos e a demanda 11

A D-24 agrega os casos por município e subcategoria, calcula os indicadores e monta o texto explicado. Método caminho_v1.

Os 10 casos concluídos

#	ABERTURA	CANAL	PRAZO	GARGALO
1	02/02	telefone	22 dias	vistoria lenta
2	09/02	app	14 dias	—
3	15/02	telefone	12 dias	vistoria lenta
4	24/02	telefone	5 dias	—
5	03/03	presencial	45 dias	material em falta
6	11/03	telefone	14 dias	falta de protocolo
7	18/03	app	6 dias	—
8	27/03	telefone	8 dias	vistoria lenta
9	05/04	app	18 dias	falta de protocolo
10	12/04	telefone	15 dias	vistoria lenta

O dossiê publicado

10 demandas de "Buracos na via" concluídas em Ituiutaba, entre 02/02/2026 e 12/04/2026.

Canal mais usado: telefone (6), depois app (3) e presencial (1).

Órgão acionado: Secretaria de Obras.

Prazo mediano: 14 dias (mínimo 5, máximo 45).

Gargalos recorrentes: vistoria lenta (4) e falta de protocolo (2).

Documentos mais pedidos: foto do local (7) e endereço completo (5).

Demanda 11 · o que ela recebe

O dossiê aparece no acompanhamento público, no relatório exportado e no workspace do conselheiro no momento em que ele assume a demanda.

O conselheiro decide: segue o caminho, adapta ou faz diferente. O que ele registrar vira o caso 11 e atualiza o perfil.

Mediana, não média: a média dos prazos seria 15,9 dias, puxada pelo caso de 45 dias. A mediana de 14 dias representa o caso típico, e é ela que aparece no dossiê.

10

CASOS NO PERFIL

14

PRAZO MEDIANO (DIAS)

6

POR TELEFONE

4

GARGALO: VISTORIA LENTA

Perguntas frequentes sobre a concatenação

As dúvidas que qualquer pessoa nova no projeto vai ter, respondidas sem rodeio.

Como o sistema junta vários textos em um só?

Não colando textos. Cada atualização já sai estruturada em campos tipados. O sistema extrai os campos, conta, ordena e preenche um modelo de frases. O texto final é a leitura desses dados.

Dois relatos dizem a mesma coisa. O que acontece?

Os textos são normalizados e comparados por palavras. Trechos com 60% ou mais de semelhança contam como o mesmo ponto. O repetido vira contagem, e o diferente entra como detalhe somado.

Quem garante que o resumo não distorce o relato?

Cada resumo tem versão de método, data de geração, lista de fontes e marca de automático. O conteúdo bruto continua no log. Se um dado for corrigido, o rebuild refaz a projeção a partir do log.

Isso é aprendizado de máquina?

No MVP é conhecimento estruturado com recuperação determinística. O perfil se forma já com poucos ciclos. O aprendizado estatístico, com embeddings e treino, entra quando houver volume e a VPS de IA.

De onde vêm os textos que formam o resumo?

Do próprio ciclo: as atualizações publicadas pelo conselheiro e o desfecho. Nada é buscado fora do sistema, e o texto bruto permanece no log.

O que acontece na primeira demanda de uma categoria?

Ainda não existe perfil. A tela avisa que o caminho está em formação e convida o conselheiro a registrar o dele. O primeiro caso já aparece no dossiê da segunda demanda.

Quando um modelo de linguagem entra?

Na fase da VPS de IA. A entrada continua a mesma: campos estruturados e textos já sanitizados. O que muda é o motor, registrado no campo próprio. O resultado ficará mais natural, e a auditoria permanece.

A IA decide alguma coisa?

Não. Ela sugere, resume, classifica com score e sinaliza. Quem decide, publica e assume é o humano. A IA não prioriza, não pune e não publica sozinha.

O que a D-24 agrega exatamente?

Por município e por subcategoria: contagem de casos, canais usados, órgãos acionados, documentos pedidos, prazo mediano, gargalos recorrentes e taxa de resolução. O dossiê é a leitura disso em linguagem clara.

Por que por município e não por bairro?

Buraco na via, iluminação e poda são competência municipal. O caminho do órgão é o mesmo para a cidade. O bairro continua visível na demanda e no mapa, mas o caminho se acumula no nível certo.

Como sei que o dossiê está atualizado?

Cada publicação traz data de geração, versão de método e a lista de fontes. Todo caso novo republica o perfil, e o rebuild reconstrói tudo a partir do log quando um dado é corrigido.

E se o texto tiver dado pessoal?

A sanitização remove telefone, nome e endereço exato antes de persistir ou publicar. O que é público é o processo e o resultado agregado, nunca a pessoa.

Glossário do Formigueiro

Os termos usados neste guia, na ordem em que aparecem no projeto.

Colônia — módulo independente do sistema, com estado próprio, que consome e publica eventos. Exemplo: D-24.

Núcleo — infraestrutura comum: Event Bus, Registry e Observabilidade. Não tem regra de negócio.

Evento — fato publicado no barramento, com tipo, versão e payload. É a única forma de uma colônia falar com outra.

Barramento (Event Bus) — canal único por onde todos os eventos passam. Guarda o log append-only e reentrega o que falhou.

Registry — catálogo que define tipos, versões e schemas dos eventos. É o idioma comum do sistema.

Log append-only — histórico imutável dos eventos. Nada é apagado; tudo pode ser reprocessado.

DLQ — fila de eventos que falharam no processamento, para reentrega e reprocessamento.

Replay — reprocessar eventos do log a partir de um cursor. É o que reconstrói projeções.

Cursor — marcador de até onde cada colônia já processou cada tipo de evento.

Projeção — cópia de leitura mantida pela própria colônia, sem consultar o banco da vizinha.

Snapshot — retrato de uma demanda na D-7, com status, categoria, contadores e marcos.

Agregado — conjunto de demandas equivalentes unidas por confirmação coletiva. Tem um representante e os membros.

Relator (conselheiro) — pessoa sorteada que acompanha a demanda, registra o andamento e conclui o ciclo.

Unidade cívica — recorte territorial do sistema, em sete níveis. A demanda é resolvida no menor nível possível.

Caso (D-24) — o caminho real de um ciclo concluído: canal, órgão, protocolo, documentos, prazos, gargalos e desfecho.

Perfil de caminho — a soma dos casos de um município e uma subcategoria. É a estatística que vira dossiê.

Dossiê — texto explicado e concatenado que resume o perfil, com contagens, mediana, gargalos e referências.

Resumo macro — a concatenação dos relatos de um agregado, publicada pela D-12.

Resumo micro — a concatenação das atualizações de um ciclo, publicada pela D-6b.

Sanitização — remoção de dados pessoais antes de persistir ou publicar qualquer texto.

Versão de método — rótulo do gerador do texto, como *resumo_v1* e *caminho_v1*. Permite comparar e evoluir sem quebrar contrato.

IA assistiva — IA que sugere e reduz atrito. Nunca decide, nunca prioriza, nunca publica sozinha.

Mapa de eventos do ciclo da demanda

Quem publica, quem consome e o que cada evento carrega. Toda colônia nova registra seus eventos no Registry antes de publicar.

EVENTO	PUBLICA	CONSUME	O QUE CARREGA
demanda.recebida	D-1a	D-1b, D-1c	Texto bruto, coordenadas, mídias, consentimentos e a categoria escolhida pelo cidadão.
demanda.normalizada	D-1b	D-2, D-3, D-7, D-12	Título inferido, descrição limpa, confiança da normalização e termos suspeitos da denylist.
anexo.processado	D-1c	D-1d, D-7	Hash, tipo, tamanho, classificação NSFW e de pessoa, evidência pública.
moderacao.decidida	D-1d	D-1c, D-7	Decisão humana (aprovar, bloquear ou remover), motivo e trilha auditável.
demanda.georreferenciada	D-2	D-3, D-4, D-7, D-24	UC de menor nível, cadeia completa de UCs, método e confiança da resolução.
demanda.categorizada	D-3	D-4, D-7, D-12, D-24	Categoria, nível de precedência, score horizontal, confiança e alternativas.
demanda.ranqueada	D-4	D-5, D-7	Score final e posição da demanda no ranking da unidade cívica.
ranking.atualizado	D-4	D-5, D-7	Topo do ranking reconstruído, quando a primeira posição muda.
agenda.item_disponível	D-5	D-6a	Demanda pronta para receber conselheiro, com posição no backlog.
conselheiro.sorteado	D-6a	D-5, D-6b	Atribuição da demanda ao conselheiro sorteado.
conselheiro.atualização_publicada	D-6b	D-7, D-24	Texto final do conselheiro mais os campos estruturados do tipo da atualização.
demanda.concluída	D-6b	D-5, D-7, D-24	Fecho do ciclo e dias até a conclusão.
conselheiro.ciclo_concluído	D-6b	D-6a, D-7, D-24	Motivo do encerramento, datas, duração e resumo final em linguagem neutra.
duplicidade.agregada	D-12	D-4, D-5, D-6a, D-6b, D-7, D-24	Representante, membros unidos e confirmações que formaram o agregado.
caminho.atualizado	D-24	D-6b, D-7	Perfil do caminho e dossiê explicado de um município e subcategoria.
demanda.resumo_agregado_atualizado	D-12	D-7	Resumo público dos relatos do agregado, com fontes e versão de método.
demanda.resumo_ciclo_atualizado	D-6b	D-7	Resumo público do ciclo, atualização por atualização, até o desfecho.