

Backlog – Evolução do SEND para aceitar veículos To-Be

FASE 1 – SEND para aceitar veículos

Descrição:

Evoluir o SEND para processar contratos de alienação de **veículos** além de consórcio, incluindo fluxo de tratamento de documentos não encontrados (24h) e integração com Kafka para eventos.

Título:

SEND – Fase 1: Inclusão de veículos e tratamento not found

História de Usuário:

[Como] analista de integração de gravames

[Quero] que o SEND aceite e processe também contratos de alienação de veículos

[Para] reduzir manualidade e riscos, ampliando cobertura da automação.

Narrativa de negócio:

Hoje os analistas só têm processo automatizado para consórcio. Para veículos, é necessário fazer tratamento manual. Com a evolução, os veículos passam a ser aceitos pelo SEND, mantendo compatibilidade com consórcio, e eventos de documentos não encontrados são publicados em Kafka.

Fluxo:

1. API Gravames de Veículos recebe solicitação.
2. processar-inclusao-gravame-lambda-function inicia o fluxo.
3. Payload enviado para processamento-envio-contrato-alienacao-queue.
4. preca-processar-envio-contrato-alienacao-lambda-function executa normalização.
5. Documento encontrado -> base64-di4-file -> envca-enviar-contrato-alienacao-lambda-function -> B3-SEND/BitOne.
6. Documento não encontrado -> doc-notfound-handler -> fifo-doc-not-found-event (delay 2h) -> di4-notfound-error-handler -> dlq-not-found ou Kafka (highway-topic-veiculo-doc-notfound).
7. Documentos enviados com sucesso -> Kafka (highway-topic-veiculos-docs-enviados-sucesso-B3Send).

Impactos esperados:

- SEND deixa de ignorar veículos.
- Redução da carga manual de analistas.
- Publicação de eventos de erro em Kafka para rastreabilidade.

Premissas de desenvolvimento:

- Usar Lambdas, SQS, SNS, Kafka (AWS managed services).
- Não quebrar fluxo de consórcio.
- Garantir compatibilidade com B3-SEND e BitOne.

Regras de Negócio:

- **RN1:** Documentos de veículos devem ser aceitos no SEND.
- **RN2:** Documentos não encontrados em 24h -> Kafka highway-topic-veiculo-doc-notfound.
- **RN3:** Documentos encontrados -> enviados via B3-SEND/BitOne.
- **RN4:** Funções devem manter compatibilidade com consórcio.
- **RN5:** Logs e métricas segregam veículos x consórcio.

Critérios de Aceite:

- CA1: Payload de veículos é aceito e processado corretamente.
- CA2: Documentos não encontrados em 24h -> evento Kafka.
- CA3: Documentos encontrados são enviados via B3-SEND/BitOne.
- CA4: Logs diferenciam consórcio e veículos.

Requisitos Funcionais:

- RF1: Evoluir preca-... e envca-... para veículos.
- RF2: Criar doc-notfound-handler e di4-notfound-error-handler.
- RF3: Implementar filas fifo-doc-not-found-event, dlq-not-found.

- RF4: Publicar eventos Kafka de erro e sucesso.

Requisitos Não Funcionais:

- RNF1: Performance equivalente ao fluxo consórcio.
- RNF2: Confiabilidade
- RNF3: Observabilidade separada por domínio.
- RNF4: Segurança: logs sem PII.

♦ FASE 2 – Governança via Feature Toggle e rollout seguro

Descrição:

Incluir camada de **Feature Toggle** (Portal Manager -> sync -> quickconfig-cluster -> cache-parametros), permitindo rollout gradual de veículos sem impactar consórcio, com fallback seguro e alta cadência (~400k execuções/dia).

Título:

SEND – Fase 2: Feature Toggle, Cache e Governança

História de Usuário:

[Como] analista de integração de gravames
 [Quero] habilitar/desabilitar o processamento de veículos via Portal Manager (Feature Toggle)
 [Para] controlar rollout de forma segura sem impactar o consórcio.

Narrativa de negócio:

Hoje o SEND processa consórcio e, com a Fase 1, passa a processar veículos. Porém, é necessário controlar a ativação via Feature Toggle, garantindo rollback rápido e rollout gradual em ambientes DEV/HML/PRD.

Fluxo:

1. Analista altera toggle no Portal Manager.
2. Serviço sincroniza-valores-cadastrados atualiza quickconfig-cluster.
3. quickconfig-cluster propaga valores para cache-parametros.
4. Lambdas (processar-inclusao-gravame, preca, envca) consultam cache antes de processar veículos.
5. Se toggle ativo -> veículos processados normalmente.
6. Se toggle inativo -> veículos ignorados (fluxo consórcio segue inalterado).

Impactos esperados:

- Controle granular de rollout.
- Rollback rápido via toggle.
- Segurança para ativação em ambientes separados.

Premissas de desenvolvimento:

- Integração Portal Manager ↔ sync ↔ quickconfig ↔ cache.
- Toggle auditável e propagação em ≤ 60s.
- Fallback: em falha, veículos ficam desativados.

Regras de Negócio:

- **RN1:** Veículos só processados se toggle ativo.
- **RN2:** Se cache indisponível -> fallback = veículos desativados.
- **RN3:** Alterações do Portal Manager refletem no cache em ≤ 60s.
- **RN4:** Toggles devem suportar granularidade (ambiente, UF, tipo de veículo).
- **RN5:** Telemetria deve diferenciar tráfego consórcio x veículos.

Critérios de Aceite:

- CA1: Com toggle false, veículos ignorados, consórcio inalterado.
- CA2: Com toggle true, veículos processados e enviados.
- CA3: Alterações no Portal Manager propagam em ≤ 60s.
- CA4: Rollback via toggle corta tráfego de veículos em ≤ 60s.
- CA5: Latência p95 mantida ≤ baseline+5% mesmo em 400k execuções/dia.

Requisitos Funcionais:

- RF1: Implementar integração Portal Manager ↔ sync ↔ quickconfig ↔ cache.
- RF2: Adaptar lambdas (processar-inclusao-gravame, preca, envca) para checar toggle.
- RF3: Criar governança de toggles (nomenclatura, granularidade).
- RF4: Dashboards segregados por consórcio x veículos.

Requisitos Não Funcionais:

- RNF1: Rollout/rollback ≤ 60s.
- RNF2: Confiabilidade ≥ 99,5% mesmo sob 400k execuções/dia.
- RNF3: Segurança: toggles auditáveis e RBAC no Portal Manager.
- RNF4: Escalabilidade: cache com baixa latência e fallback seguro.

FASE 1 — Evolução do SEND para aceitar veículos

@Fase1 @Dominio-Veiculos

Feature: SEND processa contratos de alienação de VEÍCULOS além de Consórcio

Como analista de integração de gravames

Quero que o SEND aceite/processa veículos

Para reduzir manualidade e riscos mantendo compatibilidade com Consórcio

Background:

Given o fluxo atual aceita Consórcio e ignora Veículos

And os serviços de DI4, B3-SEND e BitOne estão acessíveis

And filas e tópicos (SNS/SQS/Kafka) estão criados e com políticas corretas

@Story - 1.1 @Lambda-preca

Scenario Outline: Normalização e preparo de payload de veículo na preca-...

Given um payload de "<tipoDominio>" com campos "<chassi>", "<renavam>", "<placa>", "<uf>"

When a função preca-processar-envio-contrato-alienacao-lambda-function recebe o evento

Then o payload é validado e normalizado segundo o schema de veículos

And é aplicada idempotência com chave "<chaveIdempotencia>"

And o evento é enfileirado em processamento-envio-contrato-alienacao-queue

And métricas e logs são marcados com dominio="<tipoDominio>"

Examples:

tipoDominio	chassi	renavam	placa	uf	chaveIdempotencia
VEICULO	9BWZZZ377VT004	12345678901	ABC1D23	SP	chassi+contrato+uf
VEICULO	8ADYX1234K000	98765432100	XYZ2E45	MG	chassi+contrato+uf

@Story - 1.2 @Lambda-envca

Cenário: Envio de documento de VEÍCULO para provedor (B3-SEND/BitOne)

Given um documento DI4 convertido para base64-di4-file

And a função envca-enviar-contrato-alienacao-lambda-function está habilitada para VEÍCULOS

When o documento é roteado para o provedor compatível

Then a autenticação específica de veículos é utilizada

And o envio é realizado com confirmação de aceite

And métricas de sucesso são registradas com dominio="VEICULO"

@Story - 1.3 @Erro-NotFound

Cenário: Documento não encontrado após 2h -> reprocesso com atraso (FIFO)

Given o doc-notfound-handler recebeu indicação de documento ausente

When o evento é publicado em fifo-doc-not-found-event com delay de 2h

Then o reprocesso é tentado após o atraso

And o resultado é registrado (success/failure) mantendo idempotência

@Story - 1.3 @Erro-NotFound-24h

Cenário: Documento não encontrado em até 24h -> publicar em Kafka tópico de not-found

Given o di4-notfound-error-handler acompanha a janela de 24h

When o documento permanece ausente por 24h

Then o evento é publicado no highway-topic-veiculo-doc-notfound

And o item é movido para dlq-not-found

And logs e métricas refletem o status "not_found_24h"

@Story - 1.4 @Kafka-Sucesso

Cenário: Publicação de sucesso em Kafka após envio concluído

Given o envio do documento para o provedor foi concluído com sucesso

When a confirmação de aceite é recebida

Then publicar no highway-topic-veiculos-docs-enviados-sucesso-B3Send

And registrar métricas por dominio="VEICULO" e provedor

@Aceite

Cenário: Critérios de aceite da Fase 1

Given payloads de veículos válidos

When processados por preca/envca
Then veículos deixam de ser ignorados pelo SEND
And documentos não encontrados são tratados (2h e 24h) com publicação em Kafka
And consórcio continua funcionando sem regressões (paridade de performance e sucesso)

FASE 2 — Governança via Feature Toggle, Cache e Rollout Seguro

@Fase2 @FeatureToggle

Feature: Governança do processamento de VEÍCULOS via Feature Toggle
Como analista de integração de gravames
Quero habilitar/desabilitar VEÍCULOS via Portal Manager
Para realizar rollout por ambiente e rollback rápido sem afetar Consórcio

Background:

Given existe o Portal Manager (Foundation Manager) com chaves de toggle
And existe o serviço sincroniza-valores-cadastrados
And existe o quickconfig-cluster
And existe o cache-parametros operacional (ex: Redis/ElastiCache)
And lambdas consultam o cache antes de processar VEÍCULOS

@Story - 2.1 @Pipeline-Config

Scenario Outline: Propagação de alteração de toggle do Portal ao cache
Given a chave "<chaveToggle>" está com valor "<valorInicial>"
When o Analista altera para "<valorFinal>" no Portal Manager
Then o sincroniza-valores-cadastrados publica a alteração
And o quickconfig-cluster processa e valida o payload
And o cache-parametros recebe "<chaveToggle>" com "<valorFinal>" em até 60s

Examples:

chaveToggle	valorInicial	valorFinal	
feature.send.vehicles.enabled	false	true	
feature.send.vehicles.enabled	true	false	

@Story - 2.2 @Lambdas-Toggle

Cenário: Lambdas respeitam toggle para VEÍCULOS
Given feature.send.vehicles.enabled=false no cache-parametros
When chega um payload de dominio="VEICULO"
Then preca/envca/processar-inclusao-gravame não processam o payload
And registrar decisão de bloqueio por toggle
And consórcio segue normal

@Story - 2.2 @Fallback

Cenário: Fallback seguro quando o cache de parâmetros está indisponível
Given indisponibilidade temporária do cache-parametros
When uma lambda tenta consultar feature.send.vehicles.enabled
Then aplicar fallback seguro "veiculos desativados"
And gerar métrica e log de fallback
And consórcio permanece inalterado

@Story - 2.3 @Rollout-Ambientes

Scenario Outline: Rollout por ambiente (DEV -> HML -> PRD)
Given feature.send.vehicles.enabled=false em todos os ambientes
When ativamos em "<ambiente>"
Then somente o tráfego de "<ambiente>" processa VEÍCULOS
And métricas e tracing separam dominios e ambientes
And rollback em "<ambiente>" é efetivo em ≤ 60s

Examples:

ambiente	
DEV	
HML	
PRD	

@Story - 2.4 @Observabilidade

Cenário: Observabilidade e governança de toggles
Given métricas e tracing configurados por dominio
When VEÍCULOS é ativado
Then dashboards exibem taxas de sucesso/erro, p95/p99 e volume por dominio
And alarmes disparam para cache-miss, toggle-drift e erro de provedor

@Aceite

Cenário: Critérios de aceite da Fase 2
Given o toggle existe e propaga ≤ 60s do Portal ao cache
When alternamos ON/OFF para VEÍCULOS
Then o comportamento muda sem redeploy e sem afetar Consórcio
And sob 400k execuções/dia p95 ≤ baseline+5%