# Backlog – SEND To-Be

## FASE 1 – Evolução do 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 ≥ 99,5%.
* 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.