

# Soluções Cross Dados - R3 S4 2025

## Seção: Monitora Ops - COPF - AA e AB

### História: [Delivery][Frontend] Desenvolvimento dos big numbers (cards) da tela de Home Page (dashboard) (Frontend)

#### Visão de Produto

Acreditamos que, ao **exibir Big Numbers (indicadores resumidos e de alto impacto visual) na tela inicial do COPF (dashboard)**, página visível apenas para **analistas operacionais e gestores (não estará visível para fornecedores)**, resultará em **uma visão imediata da situação geral das ocorrências e permitirá priorizações rápidas e orientadas por dados.**

Saberemos que isso é verdade quando **os usuários conseguirem, com apenas uma olhada na dashboard, entender a gravidade do cenário atual, como número de ocorrências críticas, agências VIPs afetadas e equipamentos inoperantes.**

---

#### Descrição

Como time de frontend da COPF, queremos **desenvolver e disponibilizar na tela inicial do produto os Big Numbers**, com destaque visual e navegabilidade, para que **os times operacionais possam ter uma leitura rápida e objetiva dos principais KPIs da ferramenta.**

Indicadores exibidos:

1. **Ocorrências pendentes**
2. **Ocorrências críticas**
3. **Agências afetadas** (com destaque para VIPs)
4. **Equipamentos inoperantes**

Cada indicador contará com:

- Valor absoluto atualizado a cada vez que o usuário clicar no botão de atualizar.
  - Acesso a "Detalhes" com redirecionamento para as telas de listagem e filtros aplicados.
- 

## Visão do Usuário

Usuários operacionais e gestores precisam visualizar os principais indicadores de status logo ao acessar a ferramenta, para tomar decisões rápidas, priorizar ações e distribuir esforços da equipe com base na gravidade do cenário atual.

---

## Contexto / Narrativa de Negócio:

Antes da exibição desses indicadores, os times precisavam abrir várias telas ou consultar painéis externos para entender a situação da rede. Isso dificultava reações rápidas em contextos críticos. Os Big Numbers oferecem uma leitura executiva e operacional da situação em tempo real.

---

## Premissas

1. Os indicadores são calculados com base em views ou queries já existentes.
  2. Os dados utilizados são os mesmos da tela de ocorrências, garantindo consistência.
  3. A navegação para "Detalhes" aplica filtros correspondentes ao indicador clicado.
- 

## Regras de Negócio

1. **Ocorrências pendentes:** Total de ocorrências abertas.
  - a. Status da ocorrência: A iniciar + Em Andamento + Com impedimentos
2. **Ocorrências críticas:** Subconjunto das pendentes, marcadas como críticas.
  - a. Status da ocorrência: A iniciar + Em Andamento + Com impedimentos
  - b. Criticidade: Críticas
3. **Agências afetadas:** Contagem de agências com ao menos uma ocorrência pendente; VIPs devem aparecer em destaque.

4. **Equipamentos inoperantes:** Contagem de equipamentos com status de indisponibilidade operacional.
    - a. Status da ocorrência: A iniciar + Em Andamento + Com impedimentos
    - b. Status do equipamento: Inoperantes
- 

## Informações Técnicas

- Os valores devem ser renderizados com destaque visual (ex: fonte grande, ícones se aplicável).
  - O componente de **"Detalhes"** deve permitir redirecionamento com filtros aplicados.
  - Os dados devem ser atualizados de forma reativa com base no backend atual (streaming ou polling).
- 

## Tarefas

1. Mapear os dados necessários para cada indicador.
  2. Construir os componentes de Big Numbers no front-end.
  3. Criar a navegação para as páginas de detalhe com os filtros aplicados.
  4. Realizar testes de exibição e performance.
- 

## Cenários para Teste e Homologação

1. A home carrega corretamente os quatro indicadores com dados atualizados.
  2. Cada botão "Detalhes" leva à tela correta com os filtros esperados.
  3. Layout responsivo e legível em diferentes resoluções.
  4. Os dados exibidos batem com as queries de base.
- 

## Impacto Esperado

- Aumento da velocidade de análise e reação dos times operacionais.
  - Maior clareza do status geral da operação logo na entrada da ferramenta.
  - Redução de cliques e tempo para identificar prioridades.
-

## Conclusão

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*
- **Resultado:** *(preencher ao fim da sprint)*

## História: [Delivery][Backend] Desenvolvimento da Lambda de agregação para Big Numbers da Home Page (dashboard)

### Visão de Produto

Acreditamos que, ao **desenvolver uma Lambda responsável por consolidar e entregar os indicadores que compõem os Big Numbers da Home (Dashboard) do COPF**, será possível **garantir atualização eficiente, performance e consistência dos dados que sustentam a experiência de leitura rápida e tomada de decisão por parte dos usuários.**

Saberemos que isso é verdade quando **os Big Numbers forem exibidos com informações atualizadas, consistentes com os dados operacionais, e com tempo de resposta adequado para uso no dia a dia operacional.**

---

### Descrição

Como time de engenharia de dados/backend da COPF, queremos **implementar uma função Lambda em AWS que agregue e exponha os dados necessários para os indicadores principais da Home (Big Numbers)**, para que o frontend possa consumi-los de forma simples e performática.

A função deve:

- Ser acionada sob demanda (ex: clique no botão de atualizar ou ao carregar a página).
- Realizar as agregações conforme as regras de negócio.

- Retornar um JSON contendo os valores dos quatro indicadores.
  - Ser acessada via API Gateway com autenticação apropriada.
- 

## Visão do Usuário

Embora invisível para o usuário final, essa Lambda viabiliza que os dados dos Big Numbers estejam sempre consistentes com a base atual, com performance adequada e baixo custo de infraestrutura.

---

## Contexto / Narrativa de Negócio

A Lambda centraliza o processo de agregação com regras de negócio bem definidas, garantindo dados confiáveis e encapsulados para o dashboard.

---

## Premissas

1. As tabelas e views utilizadas já estão disponíveis no ambiente da VO3 - COPF.
  2. A Lambda será protegida por autenticação e autorizada apenas ao front da COPF. Além de validar se quem está chamando, é da operação, pois somente a operação deve conseguir obter esses valores.
  3. O deploy será realizado via pipeline automatizado.
- 

## Regras de Negócio (para cálculo dos indicadores)

1. **Ocorrências pendentes:** Contar todas com status:
    - A iniciar , Em Andamento , Com impedimentos
  2. **Ocorrências críticas:** Subconjunto das pendentes com **criticidade = Crítica**.
  3. **Agências afetadas:** Contagem de agências com ao menos uma ocorrência pendente.
    - a. Indicar também quantas são VIPs.
  4. **Equipamentos inoperantes:**
    - Ocorrências com status A iniciar , Em Andamento , Com impedimentos
    - E equipamentos com status = Inoperante
- 

## Informações Técnicas

- Linguagem sugerida: Python 3.11
- Infra: AWS Lambda + API Gateway
- Output (JSON) - Sugestão IA:

```
{
  "ocorrencias_pendentes": 747,
  "ocorrencias_criticas": 254,
  "agencias_afetadas": {
    "total": 211,
    "vips": 41
  },
  "equipamentos_inoperantes": 326
}
```

---

## Tarefas

1. Especificar as queries de agregação para cada indicador.
2. Implementar a Lambda com controle de erros e logging.
  - a. Identificar com o TechLead, Desenvolvedor FrontEnd e Designer, os valores de retorno quando ocorrer erro.
3. Realizar testes unitários e de performance.

---

## Cenários para Teste e Homologação

1. Lambda retorna todos os indicadores com dados consistentes.
2. Manipulação de falhas (ex: erro na query, retorno vazio).
3. Integração validada com o front-end (mock ou ambiente real).
4. Logs gerados para cada execução.

---

## Critérios de Aceite

- ☐ A Lambda está disponível em endpoint autenticado.
- ☐ O retorno segue o padrão de JSON acordado.
- ☐ Os dados retornados batem com as views de origem.

☐ O código está versionado e integrado no repositório da COPF.

---

## Impacto Esperado

- Consolidação da lógica de negócio em um ponto único.
  - Maior confiabilidade e consistência dos dados apresentados no dashboard.
- 

## Conclusão

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*
- **Resultado:** *(preencher ao fim da sprint)*

## História: [Delivery][Frontend] Desenvolvimento dos gráficos da dashboard da Home Page (COPF)

### Visão de Produto

Acreditamos que, ao **exibir gráficos de análise na Home Page (Dashboard) do COPF**, como **status de equipamentos, agências mais afetadas, aging de ocorrências, motivos mais frequentes e fornecedores e a relação previsão de atendimento e sla**, os times operacionais poderão **identificar rapidamente padrões, tendências e gargalos**, otimizando decisões e melhorando a eficiência da atuação.

Saberemos que isso é verdade quando **os usuários conseguirem tomar decisões ou priorizar ações com base apenas nos gráficos exibidos, sem precisar acessar outras telas ou relatórios externos**.

---

### Descrição

Como time de frontend da COPF, queremos **implementar os gráficos principais da Home Page**, com visual limpo, responsivo e de fácil leitura, para

que os usuários tenham uma **visão analítica resumida** das ocorrências e equipamentos monitorados.

Gráficos a serem desenvolvidos:

1. **Status dos Equipamentos**
2. **Top 10 pontos por volume de ocorrências**
3. **Gráfico de Aging**
4. **Long Tail - Sintomas das Ocorrências**
  - a. Sem o eixo duplo (gráfico de linha acumulada)
5. **Fornecedor x Previsão x SLA**

Cada gráfico deve:

- Ser reativo à atualização manual da página
- 

## **Visão do Usuário**

Usuários operacionais e gestores precisam **entender rapidamente os focos de atenção**, como agências mais impactadas, principais motivos de falhas e o tempo médio de resolução, de forma visual e acessível logo na Home (Dashboard).

---

## **Contexto / Narrativa de Negócio**

Antes desses gráficos, as análises eram feitas manualmente a partir de planilhas ou painéis separados. Ao integrar os gráficos na Home Page do COPF, tornamos o monitoramento mais inteligente e proativo, especialmente em situações críticas.

---

## **Premissas**

1. As bibliotecas gráficas utilizadas são compatíveis com o framework atual do COPF.
  2. Os dados são consistentes com os já utilizados em outras telas da plataforma.
- 

## **Regras de Negócio:**

- **Status dos Equipamentos:**



- Agrupar por tipo de equipamento.
- Contabilizar status "operante" e "inoperante".
- **Top 10 pontos por volume de ocorrências**
  - Listar as 10 agências com maior volume de ocorrências pendentes.
    - São consideradas pendentes as ocorrências com os seguintes status:
      - A iniciar + Em andamento + Com impedimentos
  - Cada ocorrência conta como 1.
  - Incluir marcador para VIPs.
- **Aging das Ocorrências:** Agrupar ocorrências abertas pelas faixas de dias:
  - 0 – 1 dia
  - 1 – 2 dias
  - 2 – 4 dias
  - ≥ 5 dias

O aging é calculado como:  $\text{Agora()} - \text{Data\_Abertura}$

\*No motor de priorização, o aging é calculado usando a Data\_Transferência para o fornecedor, ou seja, é um aging focado na gestão de fornecedores.

\*Nesse caso da dashboard, é um aging com visão cliente, ou seja, quanto tempo a ocorrência está aberta.

- **Long Tail – Motivos:** Agrupar ocorrências por motivo/sintoma e ordenar por frequência decrescente.
  - Os top 3 mais críticos recebem uma cor vermelha (verificar com a designer qual o RGB).
- **Fornecedor x Previsão x SLA:**
  - O gráfico é de barra empilhada:
    1. Eixo x: Agrupar ocorrências por fornecedor.
    2. Eixo y: Quantidade de ocorrências por fornecedor
    3. Barra empilhada:

- Categorizar conforme:
    - SLA vencido
    - Previsão maior que SLA
    - Sem previsão
- 

## Tarefas

1. Implementar componentes de gráficos no frontend.
  2. Integrar com a API de backend (Lambda).
  3. Garantir layout responsivo e acessibilidade.
  4. Realizar testes de usabilidade e performance.
- 

## Cenários para Teste e Homologação

1. Os gráficos carregam corretamente com os dados da API.
  2. Os dados são atualizados ao clicar em "Atualizar".
  3. Os gráficos são responsivos em diferentes dispositivos.
  4. O comportamento visual é consistente com os padrões da plataforma.
- 

## Impacto Esperado

- Redução do tempo de análise das equipes.
  - Aumento da assertividade nas decisões operacionais.
  - Proatividade no tratamento de falhas e gargalos.
- 

## Conclusão

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*

- **Resultado:** *(preencher ao fim da sprint)*

## **História: [Delivery][Backend] Desenvolvimento da Lambda de agregação para os gráficos da Home Page (COPF)**

### **Visão de Produto**

Acreditamos que, ao desenvolver uma **função Lambda dedicada à agregação e disponibilização dos dados para os gráficos do dashboard do COPF**, entregaremos **rapidez, consistência e facilidade de manutenção**, permitindo que o frontend se concentre apenas na visualização.

Saberemos que isso é verdade quando **os gráficos da Home Page forem carregados com alta performance, refletindo dados consistentes e atualizados, sem depender de múltiplas queries dispersas ou lógicas fragmentadas.**

---

### **Descrição**

Como time de backend da COPF, queremos **implementar uma Lambda que agregue e entregue os dados necessários para a renderização dos gráficos da Home Page**, consumida via API Gateway autenticada.

Essa Lambda será responsável por consolidar as regras de negócio dos seguintes gráficos:

1. **Status dos Equipamentos**
2. **Top 10 pontos por volume de ocorrências**
3. **Gráfico de Aging**
4. **Long Tail - Sintomas das Ocorrências**
  - a. Sem o eixo duplo (gráfico de linha acumulada)
5. **Fornecedor x Previsão x SLA**

A Lambda:

- Será acionada sob demanda (via frontend).
- Retornará os dados já agrupados, ordenados e no formato pronto para visualização.

- Será protegida com autenticação e autorizada via API Gateway.
  - Permitirá desacoplamento da lógica de agregação do frontend.
- 

## Visão do Usuário

Embora os usuários não acessem diretamente a Lambda, ela garante **a entrega confiável dos dados** que alimentam os gráficos analíticos da Home Page, **proporcionando uma experiência ágil e fluida** na navegação e análise dos dados.

---

## Premissas

1. Todos os dados necessários estão disponíveis no banco de dados do VO3 - COPF no RDS.
  2. A Lambda será deployada com suporte ao pipeline CI/CD do COPF.
  3. O contrato de resposta da API será definido em conjunto com o time de frontend.
  4. O controle de acesso será feito via autenticação no API Gateway.
- 

## Regras de Negócio

- **Status dos Equipamentos:**
  - Agrupar por tipo de equipamento.
  - Contabilizar status "operante" e "inoperante".
- **Top 10 pontos por volume de ocorrências**
  - Listar as 10 agências com maior volume de ocorrências pendentes.
    - São consideradas pendentes as ocorrências com os seguintes status:
      - A iniciar + Em andamento + Com impedimentos
  - Cada ocorrência conta como 1.
  - Incluir marcador para VIPs.
- **Aging das Ocorrências:** Agrupar ocorrências abertas pelas faixas de dias:
  - 0 – 1 dia

- 1 – 2 dias
- 2 – 4 dias
- $\geq$  5 dias

O aging é calculado como: `Agora()` - `Data_Abertura`

\*No motor de priorização, o aging é calculado usando a `Data_Transferência` para o fornecedor, ou seja, é um aging focado na gestão de fornecedores.

\*Nesse caso da dashboard, é um aging com visão cliente, ou seja, quanto tempo a ocorrência está aberta.

- **Long Tail – Motivos:** Agrupar ocorrências por motivo/sintoma e ordenar por frequência decrescente.
  - Os top 3 mais críticos recebem uma cor vermelha (verificar com a designer qual o RGB).
- **Fornecedor x Previsão x SLA:**
  - O gráfico é de barra empilhada:
    1. Eixo x: Agrupar ocorrências por fornecedor.
    2. Eixo y: Quantidade de ocorrências por fornecedor
    3. Barra empilhada:
      - Categorizar conforme:
        - SLA vencido
        - Previsão maior que SLA
        - Sem previsão

---

## Informações Técnicas

- Linguagem: Python
- Infraestrutura: AWS Lambda + API Gateway
- Monitoramento: Datadog com rastreamento de erro e performance
- Controle de versão: Repositório COPF no GitHub
- **Exemplo de Output - Sugestão IA:**

```
{
  "aging": [
    { "faixa": "0-1 dias", "quantidade": 32 },
    { "faixa": "1-2 dias", "quantidade": 21 },
    { "faixa": "2-4 dias", "quantidade": 18 },
    { "faixa": "≥5 dias", "quantidade": 9 }
  ],
  "motivos": [
    { "motivo": "ATM Quebrado", "quantidade": 103 },
    { "motivo": "Rede Inoperante", "quantidade": 87 }
  ],
  "status_equipamentos": [
    { "tipo": "Notebook", "operantes": 120, "inoperantes": 12 },
    { "tipo": "ATM", "operantes": 50, "inoperantes": 8 }
  ],
  "top_agencias": [
    { "agencia": "1234", "ocorrencias": 17, "vip": true },
    { "agencia": "5678", "ocorrencias": 15, "vip": false }
  ],
  "fornecedor_sla": [
    { "fornecedor": "ABC", "sla_vencido": 10, "previsao_maior_sla": 5, "sem_p
revisao": 3 }
  ]
}
```

## Tarefas

1. Definir queries SQL ou views para cada tipo de agregação.
2. Implementar a lógica de agregação na Lambda.
3. Garantir logs detalhados para troubleshooting.

## Cenários para Teste e Homologação

1. Lambda retorna os dados esperados.
2. Todos os gráficos do frontend carregam corretamente com os dados da API.

3. Logs de erro e sucesso são registrados corretamente.
  4. Segurança e autenticação funcionam conforme esperado.
- 

## Critérios de Aceite

- ☐ Lambda responde com JSON formatado corretamente.
  - ☐ Dados consistem com o que é exibido nas telas atuais.
  - ☐ Segurança via API Gateway implementada.
  - ☐ Código versionado no repositório COPF.
  - ☐ CI/CD funcionando com sucesso.
- 

## Impacto Esperado

- Redução da complexidade no frontend.
  - Aumento da performance da página de dashboard.
  - Padronização e governança da lógica de agregação.
  - Redução de retrabalho em futuras evoluções dos gráficos.
- 

## Conclusão

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*
- **Resultado:** *(preencher ao fim da sprint)*

## História: [Delivery][Backend] Lambda de cálculo de prioridade de ocorrências (COPF)

### Visão de Produto

Acreditamos que, ao **automatizar o cálculo de prioridade das ocorrências via Lambda**, permitindo atribuir uma pontuação baseada em critérios objetivos e estratégicos, **as áreas operacionais poderão priorizar de forma mais eficaz suas ações**, concentrando esforços onde há maior impacto ou urgência.

Saberemos que isso é verdade quando **as ocorrências forem exibidas com uma pontuação de prioridade clara, consistente e alinhada às regras de negócio definidas — incluindo a classificação de criticidade (Crítica, Alta, Média, Baixa)**.

---

## Descrição

Como time de backend da COPF, queremos **desenvolver uma função Lambda que atribui uma pontuação de prioridade para cada ocorrência**, com base em critérios como localização geográfica, pontos na lista de críticos para o negócio, aging e impacto na operação do ponto.

Essa Lambda será invocada em uma step function que contempla atualmente duas lambdas:

- Lambda de ocorrências pendentes
- Lambda para calculo de priorização

Calcular a **prioridade (score entre 0 e 999)**.

- Atribuir uma **criticidade categórica (Crítica, Alta, Média, Baixa)** com base no score final.
  - Armazenar e retornar esse valor junto à ocorrência.
  - A API ser exposta via API Gateway com autenticação.
- 

## Pontos de atenção

- A primeira versão dessa lambda, será apenas para ocorrências do segmento AA.
  - Para ocorrências do segmento AB, os valores retornados devem ser nulos.
    - ranking = null
    - criticidade = null
- 

## Visão do Usuário



A prioridade e a criticidade calculadas serão usadas para ordenar (ranking), filtrar (criticidade) e destacar ocorrências, permitindo que fornecedor e operação atuem primeiro nas situações mais críticas com base em critérios padronizados e transparentes.

## Contexto / Narrativa de Negócio

Com essa Lambda, trazemos critérios claros e objetivos que permitem **priorização inteligente e categorizada**, mesmo antes de qualquer ação humana. Para o fornecedor, a criticidade será fundamental para determinar quais ocorrências ele irá priorizar.

## Premissas

1. As bases auxiliares (ex: Base 360 e Ceis\_Válidos) estão disponíveis para consulta.
2. As regras de negócio se baseiam no motor de priorização já desenvolvido pela operação da Gimea.

## Regras de Negócio

A pontuação final será a soma de critérios (pontuação total entre 0 e 999):

Exemplo de critérios:

Critério	Descrição	Pontuação
<b>UF = Maranhão</b>	A ocorrência está em agência localizada no Maranhão	+55
<b>Agência VIP</b>	A agência é marcada como VIP	+20
<b>Aging2 &gt; 5 dias</b>	A ocorrência está aberta e foi transferida para o fornecedor há mais de 5 dias	+40
<b>Impacto no ponto &gt; 50%</b>	Mais de 50% dos equipamentos da agência estão inoperantes	+30

## Cálculo do Impacto do Ponto:

- $\text{impacto (\%)} = (\text{qtde equipamentos impactados} / \text{qtde total equipamentos da agência})$
- A quantidade total de equipamentos será consultada na base **CEIS\_Válidos**.
- A quantidade de impactados virá da base atual de ocorrências.

- **Critérios para classificação de criticidade:**

Score de Prioridade	Criticidade
Ocorrências que tiveram Cobrar = Sim	Crítica
600 a 799	Média
400 a 599	Média
0 a 399	Média

Nessa primeira entrega, as pontuações não alteram a criticidade, apenas alimentam a coluna de prioridade.

## Informações Técnicas:

- **Linguagem:** Python 3.11
- **Serviços AWS:** Lambda + API Gateway + DataDog
- **Bases envolvidas:** Banco de Dados do COPF no RDS, CEIS\_VALIDOS
- **Formato de retorno esperado - Sugestão IA:**

```
{
  "ocorrencia_id": "abc123",
  "prioridade": 145,
  "criticidade": "Média",
  "componentes": {
    "ma": 55,
    "vip": 20,
    "aging": 40,
    "impacto": 30
  }
}
```

## Tarefas:

1. Validar regras de pontuação e faixas de criticidade com PM.
2. Codificar a função Lambda.
3. Implementar cálculo de impacto.
4. Testar com ocorrências reais e validar respostas.

5. Expor endpoint via API Gateway.

---

### **Cenários para Teste e Homologação:**

- Lambda retorna score correto com base nas características.
  - Lambda atribui **criticidade correta conforme faixa de score**.
  - Cálculo de impacto da agência retorna valor esperado.
  - API responde em tempo adequado e com segurança aplicada.
- 

### **CrITÉrios de Aceite:**

- ✓ Score de 0 a 999 corretamente calculado.
  - ✓ Criticidade classificada corretamente com base no score. Os status possíveis são: Baixo, Médio, Alto, Crítico.
  - ✓ Endpoint funcional e integrado ao step function.
  - ✓ Código versionado, com logs e testes unitários aplicados.
- 

### **Impacto Esperado:**

- Aumento da assertividade nas ações operacionais.
  - Priorização objetiva e alinhada à estratégia do COPF.
  - Redução de subjetividade e retrabalho em tratativas.
- 

### **Conclusão:**

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*
- **Resultado:** *(preencher ao fim da sprint)*

# História: [Delivery][Backend] Desenvolvimento do ETL de Ocorrências Pendentes - Insert

---

## Visão de Produto

Ter uma base confiável e atualizada de ocorrências em aberto é essencial para priorização e gestão eficiente. Este ETL garantirá a ingestão incremental das ocorrências pendentes diretamente do Signus.

---

## Descrição

Como engenheiro(a) de dados, queremos construir um ETL que alimente as tabelas `cadastro_ocorrencia` e `acompanhamento_ocorrencia`, com foco exclusivo em ocorrências que ainda estão abertas no Signus.

---

## Regras de Negócio

- Inserir novas ocorrências em `cadastro_ocorrencia` com os campos definidos na modelagem.
  - Para cada nova ocorrência, gerar um registro de abertura em `acompanhamento_ocorrencia`, com:
    - `tipo_evento`: "Abertura da ocorrência"
    - `descricao_evento`: "Ocorrência identificada como nova"
    - `data_hora_comunicacao_enviada`: timestamp da data/hora abertura ocorrência
  - O campo `criticidade_ocorrencia` **não permite valores nulos**, conforme a modelagem.
    - Nesta primeira versão, todas as ocorrências pendentes serão inseridas com **criticidade = "Média"**.
      - Utilizar o **código da criticidade "Média"** da tabela `criticidade_ocorrencia` para preencher esse campo.
- 

## Informações Técnicas

- Executado como Lambda.
- A execução da Lambda será feita dentro de uma Step Function, com um trigger iniciado pelo EventBridge.

- Haverá uma step function que contempla:
    - Lambda de Ocorrências Pendentes
    - Lambda de Cálculo de Prioridade
- 

## Tarefas

1. Mapear campos Signus → Modelagem COPF.
  2. Criar função Lambda para ingestão incremental.
  3. Popular `cadastro_ocorrencia` com novas pendências.
  4. Preencher o campo `criticidade_ocorrencia` com código da criticidade "Média" (por que atualmente demanda um valor (futuramente devemos mudar a modelagem para permitir valor nullable)).
  5. Gerar entrada correspondente em `acompanhamento_ocorrencia`.
  6. Validar se dados relacionais (ex: agência, equipamento, sintoma, criticidade) estão consistentes.
  7. Testar execução manual.
  8. Garantir logging básico e rastreabilidade com Datadog.
- 

## Critérios de Aceite

- ☐ Ocorrências novas alimentam corretamente a tabela `cadastro_ocorrencia`.
- ☐ Cada nova ocorrência gera um evento de abertura em `acompanhamento_ocorrencia`.
- ☐ Não há duplicidade em reprocessamentos.
- ☐ Campo `criticidade_ocorrencia` é preenchido corretamente com código da criticidade "Média".
- ☐ Logging no Datadog está ativo.

## História: [Delivery][Backend] Desenvolvimento do ETL de Ocorrências Pendentes - Update

---

### Visão de Produto

Garantir que os dados das ocorrências em aberto estejam sempre atualizados é essencial para refletir a realidade operacional nas ferramentas de gestão. No ETL de Ocorrências Pendentes, há a capacidade de atualizar periodicamente alguns dos campos das ocorrências já inseridas (realizando update), assegurando confiabilidade nas análises e nas decisões.

---

## Descrição

Como engenheiro(a) de dados do Monitora Ops (COPF), queremos implementar uma etapa no ETL de Ocorrências Pendentes que atualize os campos relevantes das ocorrências que continuam abertas no Signus, assegurando que alterações de status, fornecedor, impedimentos, SLA e datas sejam refletidas nas tabelas `cadastro_ocorrencia` e `acompanhamento_ocorrencia`.

---

## Regras de Negócio

- Atualizar apenas ocorrências que já estejam registradas na tabela `cadastro_ocorrencia`.
- Atualizar os seguintes campos:
  - `fornecedor_id`
  - `data_transferencia` e `hora_transferencia`
  - `status_ocorrencia`
  - `sla_ocorrencia`
  - `previsao_atendimento`
  - `possui_impedimento` e `motivo_impedimento`
- Atualizações que geram registro na tabela `acompanhamento_ocorrencia`:
  - Alteração de previsão de atendimento
    - `tipo_evento`: "Atualização Previsão de Atendimento"
    - `descricao_evento`: descrição do(s) campo(s) alterado(s)
    - `data_hora_comunicacao_enviada`: timestamp atual
  - Inclusão de impedimento/Alteração de impedimento/Remoção de impedimento
    - `tipo_evento`: "Inclusão de Impedimento"

- `descricao_evento` : descrição do(s) campo(s) alterado(s). Incluir motivo.
  - `data_hora_comunicacao_enviada` : timestamp atual
- 

## Informações Técnicas

- Executado na Lambda do 'ETL de Ocorrências Pendentes'.
  - Atualização incremental por partição de dados (ano, mês, dia, hora, minuto).
  - Leitura comparativa dos dados atuais no banco versus payload do Signus.
  - Garantir integridade referencial com:
    - `agencia`
    - `equipamento`
    - `sintoma_ocorrencia`
    - `criticidade_ocorrencia`
- 

## Tarefas

1. Mapear campos do Signus que podem sofrer atualização - Validar com PM.
  2. Construir lógica de detecção de alterações campo a campo.
  3. Criar rotina incremental no Lambda com controle por timestamp.
  4. Atualizar os dados em `cadastro_ocorrencia` apenas se houver alteração.
  5. Gerar entrada em `acompanhamento_ocorrencia` com campos alterados.
  6. Validar consistência de chaves estrangeiras.
  7. Implementar logs de alterações para rastreabilidade usando o Datadog.
  8. Testar a execução com casos simulados de atualização.
- 

## Critérios de Aceite

- ☐ Apenas ocorrências existentes e pendentes são atualizadas (sem sobrescrever ocorrências encerradas).
- ☐ Algumas alteração geram eventos de atualizações em `acompanhamento_ocorrencia`.

- ☐ Campos atualizados refletem as mudanças no Signus.
- ☐ Logs permitem rastrear quais ocorrências foram atualizadas e quais campos mudaram.

## História: [Infraestrutura][EventBridge] Parametrização do cron para o gatilho da Step Function de Ocorrências Pendentes

### Visão de Produto

Acreditamos que, ao **orquestrar a execução frequente da Step Function que contém as Lambdas de ingestão de ocorrências em aberto e cálculo de prioridade**, garantiremos **atualização constante da base operacional com baixa latência**, permitindo que os dados estejam sempre sincronizados para análise e decisão.

Saberemos que isso é verdade quando **as ocorrências pendentes e suas prioridades estiverem atualizadas em intervalos regulares sem necessidade de acionamento manual**.

---

### Descrição

Como time de infraestrutura do COPF, queremos **configurar uma regra de cron do EventBridge** para acionar automaticamente a Step Function de ocorrências pendentes a cada 15 minutos, para que o processo de ingestão e priorização rode de forma contínua e confiável.

#### Escopo:

- Criar regra de cron no EventBridge.
  - Acionar a Step Function configurada.
  - Validar execução automatizada e logs de sucesso/erro.
- 

### Visão do Usuário

Usuários operacionais e gestores terão dados de ocorrências e prioridades sempre atualizados sem necessidade de execuções manuais, aumentando confiabilidade e reduzindo atrasos.

---

### Contexto / Narrativa de Negócio



Com o cron, garantimos **execução contínua e governada pela infraestrutura**, com menor risco de falhas humanas.

---

## Premissas

1. Lambda de ocorrências pendentes já publicada.
  2. Lambda de ranking já publicada;
  3. Permissões IAM já criadas para execução da Step Function.
  4. Logs de execução disponíveis no Datadog/CloudWatch.
- 

## Regras de Negócio

1. A Step Function deve ser acionada a cada 15 minutos.
  2. Execuções devem registrar logs completos (início, fim, falha, sucesso).
  3. Em caso de falha, notificação deve ser enviada para o canal de monitoramento.
- 

## Informações Técnicas

- **Serviço:** AWS EventBridge + Step Functions.
  - **Agendamento:** cron(0/15 \* \* \* ? \*).
  - **Observabilidade:** CloudWatch Logs + Datadog.
  - **Segurança:** IAM Role vinculada ao EventBridge com permissão para invocar a Step Function.
- 

## Tarefas

1. Criar regra de cron no EventBridge.
  2. Configurar destino como Step Function de ocorrências pendentes.
  3. Validar execução periódica em ambiente de teste.
  4. Garantir geração de logs e rastreabilidade.
- 

## Cenários para Teste e Homologação

1. Step Function é acionada automaticamente a cada 15 minutos.

2. Logs de execução aparecem no CloudWatch/Datadog.
  3. Em caso de erro, logs permitem identificar falha.
  4. Dados de ocorrências são atualizados conforme esperado no intervalo.
- 

## Critérios de Aceite

- ☐ A **Step Function de Ocorrências Pendentes** é acionada automaticamente pelo EventBridge a cada **15 minutos**, sem necessidade de disparo manual.
  - ☐ Os **logs de execução** ficam disponíveis no **CloudWatch** e integrados ao **Datadog**, registrando início, fim, sucesso e falhas.
  - ☐ Em caso de falha na execução, um **alerta é disparado para o canal de monitoramento** definido (ex.: Datadog alert) - **Sugestão IA**.
  - ☐ A execução da Step Function resulta na **atualização das tabelas de ocorrências pendentes** dentro do SLA.
  - ☐ Teste de homologação comprova que, após pelo menos **3 execuções consecutivas**, os dados são atualizados corretamente e os logs estão íntegros - **Sugestão IA**.
- 

## Impacto Esperado

- Atualização constante da base de ocorrências.
  - Redução de latência entre evento no Signus e disponibilidade no COPF.
  - Menor dependência de execuções manuais.
- 

## Conclusão

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*
- **Resultado:** *(preencher ao fim da sprint)*

# História: [Delivery][Backend] Desenvolvimento do ETL de Ocorrências Encerradas

## Visão de Produto

Acreditamos que, ao **disponibilizar um pipeline diário com as ocorrências encerradas**, será possível oferecer **histórico consistente para análises retroativas, evolução de performance e indicadores de encerramento no COPF**.

Saberemos que isso é verdade quando **os dados de ocorrências encerradas estiverem disponíveis diariamente e forem utilizados em análises históricas e relatórios de performance**.

---

## Descrição

Como **time de dados do COPF**, queremos:

1. **Criar um ETL** para ingestão das ocorrências encerradas no **Signus**, com execução diária às **07h da manhã**, agendada via **EventBridge**.
  2. As ocorrências encerradas podem já estar constando no banco de dados do RDS COPF (pois estavam anteriormente pendentes).
  3. **Atualizar os dados** em estrutura compatível com histórico analítico e dashboards do COPF.
- 

## Visão do Usuário

Gestores e analistas precisam ter acesso a **dados históricos confiáveis de encerramento** para:

- mensurar eficiência operacional (ex: percentual de ocorrências encerradas por dia);
  - calcular tempo médio de resolução (ex: MTTR);
  - identificar gargalos (ex: há algum tipo de fornecedor ou equipamento que tem maior MTTR?);
  - gerar indicadores estratégicos de performance.
- 

## Contexto / Narrativa de Negócio

Atualmente, o COPF não possui um histórico consolidado de ocorrências encerradas, dificultando análises retroativas e comparações de performance ao longo do tempo. O ETL permitirá que o histórico seja construído e atualizado diariamente, reforçando a governança e qualidade das análises.

---

## Premissas

1. O **Signus** possui uma tabela democratizada de ocorrências encerradas.
  2. A execução via **EventBridge (07h)** já está validada pela arquitetura de soluções.
  3. A modelagem analítica do COPF aceita a atualização das ocorrências, com a operação de upsert de ocorrências encerrados.
- 

## Regras de Negócio

1. Devem ser considerados **apenas registros com status "Encerrado"** no Signus.
  - a. Utilização da tabela do Data Mesh:  
histórico\_ocorrências\_fechadas\_\_thf\_\_
2. Campos obrigatórios a serem atualizados:
  - **Status da ocorrência**
    - Encerrada
  - **Data de encerramento;**
  - **SLA;**
  - **Previsão de atendimento**
    - Último status de previsão de atendimento que estava ativo antes da ocorrência ser encerrada
  - **Responsável**
    - Última classe que estava antes da ocorrência ser encerrada
3. A carga deve ser **diária e incremental**, evitando duplicidades.
4. Quando uma ocorrência for encerrada, um evento deve ser enviado para a tabela `acompanhamento_ocorrencia` com:
  - `tipo_evento` : "Encerramento da ocorrência"

- `descricao_evento` : "Ocorrência identificada como encerrada"
- `data_hora_comunicacao_enviada` : timestamp da data/hora encerramento ocorrência

Validar com o TechLead como deve se comportar a inclusão do evento na Tabela de Acompanhamento de Ocorrência - Ocorrência Encerrada

---

## Informações Técnicas

- **Execução:** Lambda + Step Function acionada pelo EventBridge às 07h.
  - **Fonte:** API/Tabela de Ocorrências do Signus (Mesh).
  - **Destino:** Tabela `cadastro_ocorrencia` no RDS COPF.
  - **Observabilidade:** Logs e métricas no Datadog + CloudWatch.
- 

## Tarefas

1. Desenvolver o pipeline para ingestão das ocorrências encerradas.
  2. Configurar agendamento via EventBridge às 07h.
  3. Criar tabela/estrutura analítica para histórico de encerramento.
  4. Validar volumetria, consistência e esquema da base.
  5. Implementar logs e monitoramento do processo.
- 

## Cenários para Teste e Homologação

- ☐ Ocorrências encerradas são carregadas diariamente sem falhas.
  - ☐ Dados refletem corretamente os status de encerramento do Signus.
  - ☐ Não há duplicidade de registros em cargas sucessivas.
  - ☐ Estrutura analítica permite consulta em relatórios e dashboards.
  - ☐ Logs registram sucesso e falhas de execução.
- 

## Critérios de Aceite

- ✓ ETL executado diariamente às 07h via EventBridge.
- ✓ Dados encerrados disponíveis com esquema consistente.
- ✓ Dados consumíveis por painéis e relatórios históricos.

- ✓ Monitoramento e logs ativos para rastreabilidade.
- 

## Impacto Esperado

- Histórico confiável para análises retroativas.
- Melhoria na governança de dados do COPF.
- Redução da dependência de extrações manuais.
- Base sólida para evolução de indicadores estratégicos.

## História: [Delivery][Frontend] Alteração de Taxonomia – “Fornecedor” para “Responsável”

### Visão de Produto

Acreditamos que, ao **alterar a taxonomia da aplicação substituindo o termo “Fornecedor” por “Responsável” em todas as páginas do Frontend do COPF**, os usuários terão **mais clareza sobre o papel da entidade exibida, alinhado com a narrativa de negócio atualizada**.

Saberemos que isso é verdade quando **todas as páginas e componentes exibirem o termo “Responsável” de forma consistente, sem menções residuais a “Fornecedor”**.

---

### Descrição

Como time de frontend do COPF, queremos **atualizar a nomenclatura de “Fornecedor” para “Responsável” em todas as telas, componentes e dashboards**, garantindo consistência de taxonomia e clareza para os usuários.

**Observação:** Embora o ideal seja que essa alteração também ocorra no **backend** (lambdas, API Gateway e demais serviços), nesta sprint o foco será **exclusivamente no frontend**. A evolução para ajustes no backend será planejada em uma **história futura**.

#### Escopo:

- Substituir “Fornecedor” → “Responsável” em labels, títulos, colunas de tabelas, tooltips, filtros e gráficos.
- Garantir que a mudança seja aplicada em **todas as páginas onde o termo aparece** (Home, Dashboard, Ocorrências, Filtros, Relatórios).

- A alteração é **exclusiva do Frontend**, sem impacto em banco de dados ou backend.
- 

## Visão do Usuário

Usuários operacionais e gestores terão **maior clareza sem confundir "Fornecedor" com parceiros externos**, entendendo que o campo representa o **responsável direto pela tratativa da ocorrência**.

---

## Contexto / Narrativa de Negócio

A nomenclatura "Fornecedor" gerava ambiguidades, pois remetia a terceiros. A mudança para "Responsável" reflete melhor a realidade do COPF, em que o campo designa a unidade ou pessoa responsável pela tratativa.

---

## Premissas

1. A alteração será feita **apenas no Frontend** (sem ajustes em backend, banco de dados ou APIs).
  2. O campo **mantém a mesma lógica de negócio** (apenas alteração textual).
  3. A mudança deve abranger **100% das telas e componentes existentes**.
- 

## Regras de Negócio

1. Onde aparecer "Fornecedor" em labels, títulos, cabeçalhos de tabelas ou gráficos → substituir por "Responsável".
  2. Filtros e tooltips também devem refletir a nova taxonomia.
  3. Garantir que não existam duplicidades (ex.: "Fornecedor/Responsável").
- 

## Informações Técnicas

- **Framework:** mesmo usado atualmente no COPF.
  - Alterações restritas a arquivos de UI/labels.
  - Realizar varredura de código para identificar todas as ocorrências do termo.
  - Testar em diferentes resoluções e navegadores para garantir consistência visual.
-

## Tarefas

1. Mapear todas as ocorrências do termo "Fornecedor" no frontend.
  2. Substituir por "Responsável" em labels, colunas, tooltips, gráficos, filtros.
  3. Revisar impacto visual e semântico nas telas.
  4. Validar com time de negócio a consistência da mudança.
- 

## Cenários para Teste e Homologação

1. Todas as páginas exibem "Responsável" no lugar de "Fornecedor".
  2. Filtros e relatórios usam "Responsável" sem falhas de busca.
  3. Nenhuma menção a "Fornecedor" residual no frontend.
  4. Layout e responsividade mantidos após alteração.
- 

## Critérios de Aceite

- Onde aparecer "Fornecedor" em labels, títulos, cabeçalhos de tabelas ou gráficos → substituir por "Responsável".
  - Todo o funcionamento do COPF continua ocorrendo normalmente, seja filtros, seja listagem de dados.
- 

## Impacto Esperado

- Melhoria na **clareza da taxonomia** usada pelo COPF.
  - Redução de ambiguidades na interpretação de relatórios.
  - Maior alinhamento entre usuários operacionais e stakeholders de negócio.
- 

## Conclusão

- **Início:**
  - **Desejado:** R3 S3 2025
  - **Real:** R3 S3 2025
- **Fim:**
  - **Desejado:** R3 S3 2025
  - **Real:** *(preencher ao fim da sprint)*



- **Resultado:** *(preencher ao fim da sprint)*

## História: [Delivery][Backend + Frontend] Sistema de Alertas no COPF para Escalations e Mensagens

### Visão de Produto

Para garantir agilidade e governança nas ocorrências tratadas via COPF, o sistema de alertas deve notificar os usuários (operação e fornecedores) sempre que houver:

- Uma **priorização manual (escalation)** da ocorrência.
- Uma **nova mensagem registrada na ocorrência**.

O objetivo é fornecer **visibilidade imediata** para os envolvidos, com persistência dos alertas no sistema, indicadores de leitura e performance otimizada.

---

### Descrição

#### Fluxo – Usuário Fornecedor

Como **fornecedor** do COPF, quero visualizar um painel de alertas que:

1. Mostre quando uma **ocorrência for escalada** pela operação (ação de priorização).
2. Exiba mensagens enviadas pela operação.
3. Me permita indicar quais os alertas foram lidos.
4. Possua **limitação inteligente** (últimos x dias ou últimos y eventos) para manter boa performance – *valor a ser definido pelo TechLead*.

#### Fluxo – Usuário Operação

Como **operação** do COPF, quero visualizar um painel de alertas que:

1. Exiba mensagens enviadas pelo fornecedor.
  2. Me permita indicar quais os alertas foram lidos.
  3. Possua **limitação inteligente** (últimos x dias ou últimos y eventos) para manter boa performance – *valor a ser definido pelo TechLead*.
-

## Visão do Usuário

- **Operação:** quer garantir que o fornecedor receba alertas de escalations e mensagens importantes, além de ser notificado sobre mensagens recebidas.
  - **Fornecedor:** quer ser alertado de escalations e mensagens importantes da operação.
- 

## Contexto / Narrativa de Negócio

- O processo de **priorização (escalation)** é uma ação manual realizada pela operação.
  - A troca de mensagens precisa ser registrada e centralizada na ocorrência.
  - Alertas não podem depender de canais externos (e-mail, WhatsApp).
  - Todos os alertas precisam ser rastreáveis, com status de leitura e design intuitivo.
- 

## Premissas

1. O sistema já possui autenticação com perfis distintos (operação e fornecedor).
  2. O evento de escalation já está implementado como tipo de ação em `acompanhamento_ocorrencia`.
  3. O sistema de envio de mensagens já está implementado.
- 

## Regras de Negócio

- Todos os alertas (escalation ou mensagem) aparecem em painel.
  - O usuário pode marcar um alerta como "lido".
  - Alertas permanecem visíveis mesmo após leitura, mas com indicador visual.
  - Exibir apenas os **últimos x dias** ou **últimos y eventos** (*a definir pelo TechLead*).
  - Alertas devem ser exibidos em ordem decrescente por `data_evento`.
  - Somente eventos pertinentes ao **perfil logado** devem aparecer.
-

## Informações Técnicas

- **Backend:**

- View materializada/unificada de eventos (mensagens e escalations).
- Tabela auxiliar `alertas_lidos` com `alerta_id`, `mensagem_id`, `escalation_id`, `usuario_id` – *Sugestão IA*.
- Endpoint: `GET /alertas/` – *Sugestão IA*.
- Endpoint: `POST /alertas/{id}/marcar-lido` – *Sugestão IA*.
- Contrato de retorno (JSON) – *Sugestão IA*:

```
[
  {
    "id": "123",
    "tipo": "escalation",
    "mensagem": "Ocorrência 456 escalada",
    "data_evento": "2025-09-08T15:30:00Z",
    "lido": false},
  {
    "id": "124",
    "tipo": "mensagem",
    "mensagem": "Fornecedor respondeu ocorrência 456",
    "data_evento": "2025-09-08T16:10:00Z",
    "lido": true}
]
```

- **Frontend:**

- Componente visual com listagem por ordem decrescente.
- Botão “marcar como lido”.
- Filtro automático por perfil logado.

---

## Tarefas

1. Criar view de eventos recentes (últimos x dias ou y eventos) – *valor definido pelo TechLead*.
2. Criar endpoint `GET /alertas/` com status de leitura – *Sugestão IA*.

3. Criar endpoint `POST /alertas/{id}/marcar-lido` – *Sugestão IA*.
  4. Criar componente frontend de listagem de alertas.
  5. Validar visualização e comportamento por perfil logado (operação ou fornecedor).
- 

## Cenários para Teste e Homologação

- ☐ Fornecedor visualiza alerta de escalation feito pela operação.
  - ☐ Mensagem enviada pela operação gera alerta para fornecedor.
  - ☐ Mensagem enviada pelo fornecedor gera alerta para operação.
  - ☐ Alertas marcados como "lido" continuam visíveis com status atualizado.
  - ☐ Painel mostra no máximo os últimos x dias ou y eventos – *definição do TechLead*.
  - ☐ Sistema mantém performance com alta carga de alertas.
- 

## Critérios de Aceite

- ☐ Endpoint `GET /alertas/` retorna alertas de acordo com perfil logado.
  - ☐ Endpoint `POST /alertas/{id}/marcar-lido` atualiza status corretamente.
  - ☐ Frontend exibe painel de alertas em ordem decrescente.
  - ☐ Badges e status de leitura funcionam conforme esperado.
  - ☐ Performance validada com volume de teste.
- 

## Impacto Esperado

- Aumento da visibilidade sobre ações críticas (escalations).
  - Redução da dependência de canais paralelos.
  - Melhor tempo de resposta e engajamento dos fornecedores.
  - Maior governança e rastreabilidade no tratamento de ocorrências.
- 

## Conclusão

Este sistema de alertas reforça a robustez do COPF como plataforma de gestão de ocorrências, promovendo comunicação efetiva entre operação e

fornecedor, com rastreabilidade e agilidade.

## História: [Discovery][Backend] Levantamento e Definição das Bases Complementares para o COPF

### Visão de Produto

Acreditamos que, ao **realizar um discovery para mapear e validar as fontes das bases complementares que alimentarão o COPF**, conseguiremos garantir **consistência, rastreabilidade e governança dos dados de apoio**, preparando o terreno para a futura implementação dos ETLs.

Saberemos que isso é verdade quando tivermos **documentação clara das fontes oficiais, entendimento da periodicidade necessária, stakeholders responsáveis e regras de transformação preliminares para cada base**.

---

### Descrição

Como **time de dados do COPF**, queremos:

- Mapear e validar a origem das **bases complementares** necessárias ao funcionamento do COPF.
- Conduzir estudos com os times responsáveis para definir as fontes oficiais e periodicidades.
- Consolidar **documentação funcional e técnica** que embasará a fase de delivery dos ETLs.

#### Bases no escopo do discovery:

1. **Base de Agências** → Origem já definida: IBS 360 (Base 360).
  2. **Base de Equipamentos** → Estudo em conjunto com o **time de Engenharia de Dados** e o **time de Planejamento Operacional** para identificar a fonte oficial.
  3. **Lista de Fornecedores** → Estudo em conjunto com o **time da GIMEA** para identificar a base de referência confiável.
- 

### Visão do Usuário

Analistas e gestores do COPF terão, ao final deste discovery, **clareza sobre a cobertura das bases complementares e confiabilidade dos dados**, garantindo

que as futuras análises e indicadores sejam sustentados por dados corretos e rastreáveis.

---

## Contexto / Narrativa de Negócio

Hoje, parte das análises do COPF depende de cruzamentos manuais ou dados externos sem rastreabilidade. Antes de construir os ETLs, é fundamental **confirmar com as áreas responsáveis (Engenharia de Dados, Planejamento Operacional, GIMEA)** quais serão as fontes oficiais para cada base, reduzindo riscos de retrabalho ou inconsistência.

---

## Premissas

1. A **Base de Agências** virá do IBS 360 (Base 360).
  2. As fontes de **Equipamentos** e **Fornecedores** ainda precisam ser definidas em conjunto com os times de negócio e engenharia.
  3. O TechLead apoiará na definição da **periodicidade de atualização** necessária.
  4. As regras de transformação serão discutidas e registradas, mas não implementadas nesta fase.
- 

## Regras de Negócio (Discovery)

- As definições devem considerar **consistência, custo de atualização e impacto nos usuários**.
  - A periodicidade de atualização deve ser definida junto ao TechLead e stakeholders.
  - Cada base deve ter **fonte única oficial** para evitar divergências.
  - O discovery deve registrar **gaps de dados** e propor soluções ou próximos passos.
- 

## Informações Técnicas (Discovery)

- Não há ETL a ser construído nesta fase.
- Os estudos incluirão:
  - **Análise de fontes existentes** (IBS 360, catálogos internos, bases operacionais).

- **Validação de integridade e granularidade.**
  - **Mapeamento de stakeholders responsáveis.**
  - **Registro de decisões em repositório do COPF (GitHub/Equipe SCD).**
- 

## **Tarefas**

1. Mapear fontes atuais da Base de Equipamentos (com Eng. de Dados + Planejamento Operacional).
  2. Mapear fontes atuais da Base de Fornecedores (com GIMEA).
  3. Documentar periodicidade e requisitos de atualização junto ao TechLead.
  4. Registrar gaps e limitações das fontes levantadas.
  5. Consolidar proposta final de fontes oficiais para as três bases complementares.
  6. Validar documentação com stakeholders de negócio e engenharia.
- 

## **Cenários para Teste e Homologação (Discovery)**

- ☐ Fontes da Base de Agências, Equipamentos e Fornecedores documentadas.
  - ☐ Periodicidade de atualização definida ou registrada como decisão pendente.
  - ☐ Gaps e riscos levantados e comunicados.
  - ☐ Stakeholders validaram a proposta final.
- 

## **Critérios de Aceite**

- ☒ Fontes oficiais das bases complementares mapeadas e documentadas.
  - ☒ Periodicidade de atualização definida junto ao TechLead.
  - ☒ Gaps e riscos identificados e registrados.
  - ☒ Aprovação da documentação pelos stakeholders (Engenharia de Dados, Planejamento Operacional, GIMEA).
- 

## **Impacto Esperado**

- Clareza e alinhamento sobre fontes de dados que alimentarão o COPF.

- Redução de riscos e retrabalho na fase de delivery.
  - Maior confiabilidade e governança das futuras análises e indicadores.
- 

## **Seção: IBS 360 - Plataforma**

---

## **Seção: IBS 360 - Frontend**

---

## **Seção: IBS 360 - Gestão do Parque**

---

## **Seção: IBS 360 - Gestão do Parque - Acompanhamento de Esteiras**

### **História: [Delivery][Deploy] Deploy da Tela de Acompanhamento de Esteiras no Produto Frame do Agências (IBS 360)**

---

#### **Visão de Produto**

Acreditamos que, ao realizar o deploy da nova tela de Acompanhamento de Esteiras no produto Frame do Agências (IBS 360), para o ambiente oficial utilizado por diretores, superintendentes e usuários estratégicos, resultará em acesso contínuo, seguro e validado a um recurso crítico para decisões estruturais da rede de agências. Sabemos que isso é verdade quando a tela estiver publicada no ambiente de produção com segurança, versionamento e monitoramento ativo.

---

#### **Descrição**

O objetivo desta história é **operacionalizar o deploy da funcionalidade desenvolvida no Frontend** (Streamlit), garantindo que:

- A aplicação esteja publicada no ambiente **oficial do IBS 360 (Frame do Agências)**.



- O código esteja **versionado e revisado** conforme os padrões da squad.
  - O processo de deploy seja **documentado, rastreável e seguro**.
  - O monitoramento esteja habilitado (CloudWatch), garantindo observabilidade e rápida resposta em caso de falhas.
  - Os usuários autorizados consigam **acessar a tela e consumir os dados via interface**.
- 

## Contexto / Narrativa de Negócio

- A funcionalidade já foi validada com stakeholders do planejamento e áreas comerciais.
  - O deploy precisa ocorrer sem causar impacto em outras funcionalidades do IBS.
  - A funcionalidade está alinhada com os objetivos estratégicos de ocupação, otimização de parque e governança de rede.
  - A tela será acessada diretamente pelo produto "Frame do Agências", mantendo a coesão da experiência do usuário final.
- 

## Premissas

1. A tela deve ser homologada em ambiente de homologação com usuários chaves do time de Planejamento Comercial e time SETA de Planejamento.
  2. Os dados da base integradora estão disponíveis em S3 e atualizados periodicamente.
  3. O processo de deploy segue os pipelines definidos no IBS 360.
  4. As permissões de acesso seguem o modelo de autenticação do IBS via grupos de acessos cadastrados no POIAM.
- 

## Regras de Negócio

1. O deploy só deve ocorrer após checklist de QA (dados, design, performance).
2. Logs devem estar ativados para rastreamento de erros em tempo real.
3. Somente usuários com acesso devem acessar a funcionalidade.

- a. O comportamento de acesso deve respeitar
    - i. Usuários com o grupo de acesso e que não são comerciais: visualizam todas as agências.
    - ii. Usuários com o grupo de acesso e que são comerciais: visualizam apenas a sua respectiva dineg ou superintendência.
      - 1. Ex: Dineg 5, visualiza apenas as agências da diretoria comercial 5.
      - 2. Ex: Supt 51, visualiza apenas as agências da superintendência comercial 51.
- 

## Tarefas

- ☐ Revisar branch final com testes homologados e pull request aprovado.
  - ☐ Executar pipeline de deploy.
  - ☐ Verificar publicação da tela no frame de agências.
  - ☐ Validar acessibilidade e visibilidade para os usuários autorizados.
  - ☐ Acionar time de suporte para monitoramento nos primeiros dias de uso.
  - ☐ Atualizar documentação de produto com link e instruções de uso.
  - ☐ Comunicar stakeholders e líderes comerciais sobre a nova funcionalidade.
- 

## Cenários para Teste e Homologação

1. Acesso autorizado ao link da nova tela por um diretor e por um Supt.
  2. Verificação dos filtros e drill-downs em ambiente de produção.
  3. Testes de performance no carregamento de dados e resposta da interface.
  4. Simulação de erro de carregamento para validar mensagens e logs.
  5. Verificação de logs de acesso e comportamento via painel de observabilidade.
- 

## Impacto Esperado

- Liberação oficial da funcionalidade de Acompanhamento de Esteiras para toda a rede.

- Maior autonomia dos usuários na visualização da evolução das ações físicas.
  - Redução de demanda de suporte para envio de relatórios manuais.
  - Garantia de rastreabilidade e governança da entrega.
- 

## Conclusão

- **Início:**
    - **Desejado:** R3 S1 2025
    - **Real:** R3 S1 2025
  - **Fim:**
    - **Desejado:** R3 S1 2025
    - **Real:**
  - **Resultado:**
- 

## Seção: IBS 360 - Geocompasso

### História: [Delivery][Backend] Função para Conversão de Endereço/CEP em Coordenadas Geográficas

- **Visão de Produto:**

Acreditamos que, ao **implementar uma função no backend do Geocompasso que converta endereços ou CEPs em coordenadas geográficas (latitude e longitude)**, para **usuários da funcionalidade de prospecção do time de Folhas**, resultará em **maior flexibilidade na entrada de dados e automatização do processo de matching geográfico**.

Saberemos que isso é verdade quando **os dados inseridos (endereços ou CEPs) forem automaticamente convertidos em coordenadas, permitindo cálculos de raio e visualização no mapa sem necessidade**.

- **Descrição:**

Como time técnico do Geocompasso, queremos **construir uma função backend que, ao receber um endereço ou CEP, retorne sua latitude e**

**longitude**, com tratamento de erros, **para permitir a posterior análise geográfica desses pontos.**

- **Premissas:**

1. A função será consumida pelas funcionalidades do Geocompasso.

- **Tarefas:**

1. Implementar funcionalidade.
2. Testar amostragem real com dados do time de folhas.

- **Critérios de Aceite:**

- Conversão de endereço ou CEP para latitude e longitude com acurácia mínima de 85%.
- Tratamento de exceções e logging de erros disponíveis para análise.

- **Impacto Esperado:**

- Ganho de escala e agilidade para novas análises.
- Confiabilidade no posicionamento dos pontos de entrada.

- **Conclusão**

- **Início:**

- **Desejado:** R3 S1 2025
    - **Real:** R3 S1 2025

- **Fim:**

- **Desejado:** R3 S1 2025
    - **Real:**

- **Resultado:**

- R3 S2 2025 - Repriorizado para R3 S4 2025. A colaboradora Beatriz foi remanejada para a frente de Inteligência Imobiliária - Planejamento de Negociação.

## **História: [Delivery][Frontend] Entrada Manual de Endereço/CEP para Busca Direta no Geocompasso**

- **Visão de Produto:**

Acreditamos que, ao **permitir que o usuário do Geocompasso insira manualmente endereços ou CEPs para buscar pontos de interesse no mapa**, para **usuários como o time de Folhas e analistas territoriais**, resultará em **uma experiência mais dinâmica, acessível e adaptável para análises pontuais sem depender da Base 360**.

Saberemos que isso é verdade quando **usuários forem capazes de digitar um endereço ou CEP e visualizar diretamente no mapa o ponto buscado e os equipamentos no entorno, com base no raio selecionado**.

- **Descrição:**

Como **usuário do Geocompasso**, quero **digitar um endereço ou CEP diretamente na interface**, selecionar um raio de busca (ex: 2km, 4km) e visualizar **quantos TecBans, PABs, PAEs ou agências existem ao redor**, sem depender de um código de agência ou da entrada via Base 360.

- **Premissas:**

1. A função de geocodificação já está implementada no backend.

- **Tarefas:**

1. Criar componente de input para endereço/CEP com validação.
2. Conectar com a função de geocodificação.
3. Apresentar o ponto buscado no mapa com marcador.
4. Permitir ao usuário seguir o fluxo do geocompasso.

- **Critérios de Aceite:**

- Campo funcional para entrada de endereço ou CEP.
- Resultado exibido corretamente no mapa com marcador e raio.

- **Impacto Esperado:**

- Expansão do uso do Geocompasso para análises pontuais e não estruturadas.
- Redução de dependência da Base 360 para análises geográficas.
- Maior autonomia e agilidade para áreas como Folhas e Logística.

- **Conclusão**

- **Início:**

- **Desejado:** R3 S1 2025
- **Real:** R3 S1 2025
- **Fim:**
  - **Desejado:** R3 S1 2025
  - **Real:**
- **Resultado:**
  - R3 S2 2025 - Repriorizado para R3 S4 2025. A colaboradora Beatriz foi remanejada para a frente de Inteligência Imobiliária - Planejamento de Negociação.

## História: [Delivery][Frontend] Funcionalidades – Prospecção de Folhas no Geocompasso

### Visão de Produto

Acreditamos que, ao **permitir que o time de Folhas realize prospecções geográficas a partir de endereços ou CEPs diretamente no Geocompasso, com enriquecimento completo dos dados**, resultará em **maior agilidade, precisão e autonomia na avaliação de oportunidades (ex.: prospecção de folhas INSS)**.

Saberemos que isso é verdade quando **os critérios de prospecção hoje tratados manualmente em planilhas puderem ser automatizados via inserção de uma lista de ceps/endereços manualmente ou upload de planilha, com outputs enriquecidos e exportáveis**.

### Descrição

Como **usuário estratégico do time de Folhas**, quero **subir uma lista de endereços ou CEPs no Geocompasso**, para que a ferramenta **enriqueça automaticamente esses dados e retorne uma tabela consolidada (sem interface de mapa neste primeiro momento)**, contendo:

- **Cidade de referência.**
- **Se há ou não agência Itaú na cidade (sim/não).**
- Para cada entidade (Agência Itaú, TecBan, PIX Saque e Troco, Concorrentes, Parceiros Itaú):

1. Distância até a unidade mais próxima (em km).
2. Quantidade de unidades no raio de 5 km, 10 km e 15 km.
3. Lista de unidades no raio de 5 km, 10 km e 15 km (IDs/nome/endereço).

Assim, **posso avaliar a viabilidade de prospecção de novas folhas (ex.: Folha da Sabesp, Bunge) sem depender de análises manuais em Excel.**

#### **Escopo:**

- Inserção de CEP/Endereço manualmente
- Upload de planilha contendo endereços/CEPs.
- Enriquecimento automático da tabela com as métricas acima.
- Visualização apenas em tabela enriquecida (futuramente deveremos abranger visão mapa).
- Exportação da tabela enriquecida com todos os campos calculados.

---

### **Visão do Usuário**

Usuários estratégicos do time de Folhas terão autonomia para executar **análises geográficas enriquecidas e comparativas** diretamente no Geocompasso, reduzindo dependência de análises manuais e acelerando a prospecção.

---

### **Contexto / Narrativa de Negócio**

Hoje, o time de Folhas realiza cálculos de prospecção manualmente via planilhas, com alto tempo gasto e risco de inconsistência. A nova funcionalidade transforma esse processo em uma **análise enriquecida e automática**, padronizada para múltiplos pontos em lote.

---

### **Premissas**

1. Base geográfica de **agências Itaú, TecBans, PIX, concorrentes e parceiros Itaú** já disponível na Base 360.
2. Backend será capaz de realizar **matching geográfico, contagem e cálculo de distâncias**.
3. O enriquecimento será realizado via IBS 360, já preparada para ser consumida pelo frontend.

## Regras de Negócio

1. **Entrada:** planilha com endereços ou CEPs OU inserção manual do usuário.
  2. **Saída:** tabela enriquecida exportável em xlsx.
  3. Campos obrigatórios no enriquecimento:
    - Cidade.
    - Existe agência Itaú na cidade (sim/não).
    - Distância (km) da unidade mais próxima de cada tipo (Itaú, TecBan, PIX, Concorrente, Parceiro Itaú).
    - Quantidade de unidades nos raios de 5 km, 10 km e 15 km.
    - Lista de unidades nos raios de 5 km, 10 km e 15 km (detalhes: ID, nome, endereço).
  4. Os resultados devem estar disponíveis em **até y segundos** para planilhas de até *n* registros (**definir valor com TechLead**).
  5. Exportação deve conter todos os campos acima.
- 

## Informações Técnicas

- **Frontend:** Geocompasso (IBS 360 - Streamlit).
  - **Output:** Tabela enriquecida + exportação xlsx.
  - **Integração:** Backend para cálculos geográficos (matching + distâncias).
  - **Observabilidade:** Logs no CloudWatch.
- 

## Tarefas

1. Implementar upload de planilha com endereços/CEPs (já há outra história focada nisso em:)
  2. Conectar funções de backend para enriquecimento.
  3. Renderizar tabela enriquecida com todos os campos definidos.
  4. Implementar exportação da tabela enriquecida.
  5. Validar experiência de uso e performance com stakeholders.
- 

## Cenários para Teste e Homologação



- ☐ Upload de planilha gera tabela enriquecida com todos os campos obrigatórios.
  - ☐ Campo "Agência Itaú na cidade" retorna sim/não corretamente.
  - ☐ Distância mínima até a unidade mais próxima é calculada corretamente.
  - ☐ Quantidade de unidades em 5 km, 10 km e 15 km é exibida corretamente para cada tipo.
  - ☐ Exportação contém todos os campos definidos.
  - ☐ Layout é responsivo e suporta grandes volumes de dados.
- 

## **Critérios de Aceite**

- ☐ Enriquecimento automático gera todos os campos definidos.
  - ☐ Resultados apresentados em até y segundos para lotes médios (*n* registros – a definir).
  - ☐ Exportação de resultados em xlsx está disponível.
  - ☐ Dados consistentes com consultas validadas na Base 360 e Google Maps.
  - ☐ Homologação validada com time de Folhas.
- 

## **Impacto Esperado**

- Redução significativa do tempo gasto em análises manuais.
  - Maior confiabilidade e padronização nas prospecções.
  - Geocompasso consolidado como ferramenta central de decisão territorial para Folhas.
- 

## **Conclusão**

- **Início:**
  - **Desejado:** R3 S4 2025
  - **Real:** R3 S4 2025
- **Fim:**
  - **Desejado:** R3 S4 2025
  - **Real:** *(preencher ao fim da sprint)*

- **Resultado:** *(preencher ao fim da sprint)*
- 

## Seção: Radar Imobiliário

---

## Seção: Inteligência Imobiliária

**História: [Delivery][Backend] Orquestração em AWS Lambda do ETL de Base Filtrada a partir do Planejamento de Negociações**

### Visão de Produto

Acreditamos que, ao **orquestrar o processo de geração da base filtrada em uma Lambda na AWS, consumindo diretamente os parâmetros preenchidos no frontend do Planejamento de Negociações**, resultará em **um fluxo mais escalável, automatizado e seguro**, garantindo que cada simulação rode de ponta a ponta sem intervenção manual.

Saberemos que isso é verdade quando os **planejamentos realizadas pelo time de Advanced Analytics utilizarem a base filtrada gerada pela Lambda, com rastreabilidade garantida via UUID e performance adequada em múltiplos cenários**.

---

### Descrição

Como **time de engenharia do IBS 360**, queremos **migrar o fluxo de geração da base filtrada (hoje em Python standalone) para execução em uma Lambda AWS**, que:

1. Receba os **filtros preenchidos pelo frontend**.
  2. Leia os dados da **base consolidada pelo ETL1 (GPA + Osiris)**.
  3. Aplique os filtros de elegibilidade e gere uma **base reduzida e segura**.
  4. Salve o resultado em **estrutura versionada, vinculada ao UUID da simulação**, para uso pelo time de Advanced Analytics.
- 

### Visão do Usuário

O usuário final não interage diretamente com a Lambda, mas se beneficia de:

- **Simulações mais rápidas.**
- **Resultados coerentes com os filtros aplicados.**
- **Garantia de rastreabilidade** entre filtros aplicados → base gerada → resultados da simulação.

---

## Contexto / Narrativa de Negócio

Esse avanço elimina a dependência de scripts locais em Python, criando um **pipeline 100% cloud-native**:

- Automação ponta a ponta (frontend → Lambda → base filtrada).
- Escalabilidade para múltiplas simulações concorrentes.
- Governança com logs centralizados e versionamento automático.

---

## Premissas

1. A base consolidada de imóveis (ETL1) está disponível em storage corporativo (S3/RDS).
2. O frontend já envia os parâmetros de filtro ao backend.
3. O ambiente AWS do IBS 360 tem suporte a Lambda + API Gateway + S3.
4. UUID de simulação será gerado pelo backend e propagado até a Lambda.

---

## Regras de Negócio

1. A base final deve conter apenas imóveis **elegíveis conforme filtros** do usuário.
2. Cada execução gera um **arquivo exclusivo**, identificado pelo UUID.
3. Os arquivos devem ser armazenados em bucket S3 com **padrão de nomenclatura**:
  - `simulacoes/{uuid}/base_filtrada.parquet`
  - Futuramente esse base será democratizada via fluxo Glue - usando o Lake Formation.
4. Logs da execução devem registrar parâmetros aplicados e quantidade de registros filtrados.

5. A Lambda deve suportar **execuções concorrentes** (um ETL por simulação).

---

## Informações Técnicas

- **Linguagem:** Python.
  - **Infraestrutura:** AWS Lambda + API Gateway + S3.
  - **Entrada:** JSON com filtros enviados pelo frontend + UUID.
  - **Processamento:** Cálculo dos dados respeitando as regras de negócio enviadas pelo Stakeholder.
  - **Saída:** Arquivo Parquet/CSV no S3.
  - **Observabilidade:** CloudWatch logs estruturados.
- 

## Tarefas

1. Criar Lambda em Python para execução do fluxo de filtragem.
  2. Expor endpoint via API Gateway para receber filtros do frontend.
  3. Salvar base filtrada em bucket S3 com versionamento (UUID).
  4. Implementar logs detalhados (parâmetros, contagem de registros).
  5. Testar execuções paralelas com múltiplos UUIDs.
  6. Documentar arquitetura e fluxo (frontend → API → Lambda → S3).
- 

## Cenários para Teste e Homologação

- ☐ Execução da Lambda com filtros diferentes gera bases distintas (UUIDs únicos).
  - ☐ Apenas imóveis elegíveis aparecem na base final.
  - ☐ Arquivos são gravados corretamente no S3 com nomenclatura esperada.
  - ☐ Logs mostram parâmetros aplicados e número de registros processados.
  - ☐ Lambda suporta execuções concorrentes sem degradação de performance.
  - ☐ Advanced Analytics consegue consumir a base filtrada em simulações.
- 

## Critérios de Aceite

- ✓ Lambda recebe filtros do frontend e executa ETL corretamente.
  - ✓ Arquivos gerados no S3 estão vinculados a UUIDs únicos.
  - ✓ Logs estruturados disponíveis em CloudWatch.
  - ✓ Dados consistentes com filtros aplicados.
  - ✓ Base consumível pelos modelos de Advanced Analytics.
- 

## Impacto Esperado

- Automação ponta a ponta entre frontend → backend → simulação.
  - Redução do tempo de execução de simulações.
  - Rastreabilidade e versionamento garantidos.
  - Adoção de arquitetura cloud-native com governança.
- 

## Conclusão

- **Início:**
    - **Desejado:** R3 S4 2025
    - **Real:** R3 S4 2025
  - **Fim:**
    - **Desejado:** R3 S4 2025
    - **Real:** *(preencher ao fim da sprint)*
  - **Resultado:** *(preencher ao fim da sprint)*
- 

## Seção: Score de Agências

---

## Seção: App Planejamento de Pessoas

---

## Seção: Arquitetura Cross

---

## **Seção: Monitoramento e Métricas de Produto**

---

### **Seção: AWSCloudBridge**