

Soluções Cross Dados - R3 S2 2025

Seção: Ferramenta Única de Ocorrências - COPF - AA e AB

História: [Banco de Dados][Dados] Criação e População Inicial do Banco de Dados RDS da COPF

- **Visão de Produto:**

Acreditamos que, ao **criar o banco de dados RDS com as tabelas modeladas e realizar a carga inicial dos dados.**

Saberemos que isso é verdade quando **todas as tabelas estiverem criadas com a estrutura definida, constraints aplicadas, dados populados de forma íntegra e disponíveis para consulta e testes do backend.**

- **Descrição:**

Como **engenheiro responsável pela fundação de dados da COPF**, quero **criar o banco de dados no Amazon RDS e popular as tabelas com dados reais e atualizados**, garantindo **disponibilidade para consumo pela aplicação e análises, com rastreabilidade e consistência.**

- **Premissas:**

1. O modelo lógico e físico das tabelas foi finalizado na sprint anterior.
2. O engine utilizado será Aurora, conforme definido na modelagem.

- **Regras de Negócio:**

1. Os dados históricos devem ser populados com integridade entre chaves primárias e estrangeiras.
2. Dados duplicados ou inválidos devem ser tratados ou descartados na carga inicial, conforme critérios definidos com os stakeholders.

- **Tarefas:**

1. Criar instância RDS no ambiente homologado com parâmetros de performance e segurança definidos.

2. Executar os scripts de criação de tabelas com constraints, índices e normalização conforme modelagem.
3. Validar integridade relacional e amostragem da qualidade dos dados inseridos.
4. Garantir versionamento dos scripts de criação e carga no repositório do time.
5. (Opcional) Documentar os principais critérios de limpeza e tratamento utilizados na carga.

- **Critérios de Aceite:**

- Todas as tabelas foram criadas com estrutura compatível com a modelagem aprovada.
- As constraints de chave primária e estrangeira estão funcionando corretamente.
- Scripts versionados no repositório e testáveis em novos ambientes.
- As lambdas BFF já conseguem se conectar ao banco e ler os dados populados.

- **Impacto Esperado:**

- Ambiente de dados operacional disponível para testes, desenvolvimento e análises.
- Redução de retrabalho por inconsistências ou ausências de dados no início do ciclo.
- Aceleração da entrega de valor no frontend com base real.

- **Conclusão:**

- **Início:**

- **Desejado:** R3 S2 2025

- **Real:**

- **Fim:**

- **Desejado:** R3 S2 2025

- **Real:**

- **Resultado:**

História: [Delivery][Frontend] Tela de Homepage com Dashboard Operacional

- **Visão de Produto:**

Acreditamos que, ao **construir uma tela inicial (homepage) com visão consolidada das ocorrências e filtros estratégicos**, para **os analistas operacionais, gestores e stakeholders da COPF**, resultará em **uma tomada de decisão mais rápida, baseada em dados e com visibilidade em tempo real das prioridades da operação**.

Saberemos que isso é verdade quando **a homepage exibir os principais KPIs (ocorrências abertas, dentro/fora do SLA, por tipo e por fornecedor)**, com **filtros interativos que permitam navegar pelas informações e acessar detalhes diretamente a partir dos gráficos**.

- **Descrição:**

Como equipe de frontend da COPF, queremos **desenvolver uma tela de entrada com painel visual (dashboard operacional)** que apresente **os principais indicadores de desempenho, gráficos por categoria e fornecedor, status das ocorrências e alertas de risco**, para **permitir que a operação inicie seu dia com foco nas ocorrências mais críticas e nos pontos de atenção**.

- **Premissas:**

1. Os dados e métricas estarão disponíveis via Lambda de Leitura dos Dados que realizará uma consulta com o banco de dados RDS.
2. As definições de SLA, Agind e Reincidência já foram padronizadas com os stakeholders.
3. A stack utilizada permitirá atualização em tempo real ou quase tempo real dos dados exibidos.

- **Tarefas:**

1. Construir o layout da tela com foco em UX, responsividade e clareza visual.
2. Integrar os gráficos com os dados das APIs.

- **Critérios de Aceite:**

- Tela de homepage acessível como rota principal após login/autenticação.
- Gráficos com visual limpo, responsivo e com tooltips informativos.

- Filtros interativos aplicados sobre todos os indicadores e gráficos.
- Possibilidade de clicar em qualquer gráfico e ser redirecionado para a listagem filtrada.
- Dados atualizados através de um botão de atualizar.
- Validação de que os KPIs e gráficos refletem corretamente os dados do backend.
- **Impacto Esperado:**
 - Melhoria significativa na priorização do trabalho pela operação.
 - Redução no tempo gasto para identificar gargalos ou riscos.
 - Visibilidade estratégica para gestores sem depender de extrações manuais.
 - Fortalecimento do uso da COPF como ferramenta de controle operacional.

História: [Delivery][Frontend] Tela de Listagem de Ocorrências

- **Visão de Produto:**

Acreditamos que, ao **disponibilizar uma tela de listagem de ocorrências na ferramenta COPF**, para **analistas de operação e gestores**, resultará em **uma visão clara, atualizada e filtrável do status das ocorrências em toda a rede**.

Saberemos que isso é verdade quando **a tela permitir localizar, filtrar e acompanhar ocorrências com fluidez e sem sobrecarregar a aplicação**.

- **Descrição:**

Como equipe de frontend da COPF, queremos **construir a tela principal de listagem de ocorrências**, com **colunas-chave como ID, status, tipo, agência, fornecedor, SLA e tempo de tratativa**, além de **filtros dinâmicos e paginação**, para **permitir a gestão e priorização em tempo real pelas áreas operacionais**.

- **Premissas:**

- 1.

- **Tarefas:**

- 1.

- **Critérios de Aceite:**

-

- **Impacto Esperado:**

- Visibilidade operacional centralizada.
 - Melhoria na eficiência da gestão de ocorrências.

História: [Delivery][Frontend] Tela de Comunicação entre Operação e Fornecedor (Bidirecional)

- **Visão de Produto:**

Acreditamos que, ao **oferecer uma interface de comunicação direta entre o time de operação e os fornecedores**, para ambas as partes envolvidas na **resolução das ocorrências**, resultará em **respostas mais rápidas, alinhamento em tempo real e histórico completo das tratativas**.

Saberemos que isso é verdade quando **os usuários conseguirem enviar e receber mensagens vinculadas a uma ocorrência específica, com visualização clara, timestamps e alertas de nova resposta**.

- **Descrição:**

Como time de frontend, queremos **construir uma tela/modal de comunicação bidirecional entre operação e fornecedor**, com **interface semelhante a um chat vinculado à ocorrência e exibição cronológica das mensagens**, respeitando as regras de autenticação e visualização.

- **Premissas:**

1. Os endpoints de envio e leitura de mensagens estão disponíveis no backend.
2. As permissões de acesso foram definidas por perfil (operação x fornecedor).

- **Tarefas:**

- 1.

- **Critérios de Aceite:**

-

- **Impacto Esperado:**

- Redução do tempo de tratativa por falhas de comunicação.
 - Registro completo e auditável das interações com fornecedores.
-

Seção: IBS 360 - Plataforma

Seção: IBS 360 - Frontend

História: [Delivery][Frontend] Pesquisa de Satisfação - Criar Triggers na Jornada do Frontend

- **Visão de Produto:**

Acreditamos que, ao **incluir triggers para ativar a pesquisa de satisfação em pontos-chave da jornada do usuário no IBS 360**, para **usuários finais da plataforma**, resultará em **uma coleta mais natural, contextualizada e eficaz de feedbacks, com maior taxa de resposta e relevância das informações**.

Saberemos que isso é verdade quando **a pesquisa for exibida de forma fluida em momentos estratégicos, sem prejudicar a experiência do usuário e com respostas sendo armazenadas com sucesso**.

- **Descrição:**

Como time responsável pela **experiência do usuário e coleta de feedbacks contínuos**, queremos **implementar triggers no frontend da plataforma**, para exibir a pesquisa:

- Ao concluir tarefas críticas (ex: filtro ou navegação);
- Após tempo de navegação significativo;

- **Premissas:**

1. A pesquisa já está implementada no frontend como componente.
2. Os pontos da jornada para ativação da pesquisa estão mapeados.

- **Informações Técnicas:**

- Lógica condicional para exibição única por sessão.
- Ancoragem por etapa ou componente acessado.

- **Tarefas:**

1. Mapear pontos de ativação da pesquisa.

2. Criar triggers e lógica de exibição por contexto.
3. Garantir que a trigger envia os dados corretamente ao backend.

- **Critérios de Aceite:**

- Pelo menos 2 triggers diferentes implementadas (ex: após tempo de uso e após conclusão de ação).
- A pesquisa aparece no frontend de forma contextual, sem travar a navegação.
- A exibição da pesquisa é limitada a uma vez por sessão.
- A submissão do formulário dispara corretamente a chamada ao backend.
- Experiência testada em diferentes produtos da plataforma (ex: Geocompasso e Scorefy).
- Código revisado e publicado no repositório com comentários explicativos.
- Evento de trigger registrado em log no cloudwatch (opcional).

- **Impacto Esperado:**

- Melhoria da amostragem e qualidade das respostas.
- Integração fluida com a jornada sem atrito.

História: [Delivery][Backend] Pesquisa de Satisfação - Conectar Frontend com Backend

- **Visão de Produto:**

Acreditamos que, ao **implementar a integração entre frontend e backend para a submissão das respostas da pesquisa de satisfação**, para o **time de engenharia e governança**, resultará em **uma jornada de feedback contínua, com dados recebidos e armazenados de forma segura, auditável e utilizável para decisões de produto**.

Saberemos que isso é verdade quando **as respostas forem enviadas corretamente via API e armazenadas no DynamoDB com validação de schema e logging de erros**.

- **Descrição:**

Como time de engenharia da plataforma IBS 360, queremos **desenvolver os endpoints que recebem as respostas da pesquisa vinda do frontend**, com validação e envio para o DynamoDB, garantindo:

- Segurança de acesso e autenticação mínima (Auth0);
- Validação do payload para garantir integridade dos dados;
- Logging para auditoria e troubleshooting.
- **Premissas:**
 1. O DynamoDB já está criado com os atributos necessários.
 2. O frontend envia os dados via JSON estruturado.
 3. O FastAPI já está estruturado para suportar esse endpoint.
- **Informações Técnicas:**
 - Criação de rota POST `/feedback`
 - Validação com Pydantic e logs em CloudWatch
 - Escrita no DynamoDB com boto3
- **Tarefas:**
 1. Criar rota de recebimento de respostas.
 2. Validar payloads e autenticar solicitações.
 3. Escrever no DynamoDB com estrutura correta.
 4. Testar integração ponta-a-ponta com frontend.
- **Critérios de Aceite:**
 - Endpoint POST criado (ex: `/feedback`) e documentado.
 - Payload da pesquisa validado com Pydantic, incluindo campos obrigatórios e tipos.
 - Respostas são armazenadas corretamente no DynamoDB com logging de sucesso.
 - Chamadas inválidas retornam erro com mensagens claras de validação.
 - Endpoint protegido com autenticação mínima via Auth0 ou API Gateway.

- Integração com frontend testada ponta a ponta (dados reais salvos no banco).
 - Logs visíveis no CloudWatch ou ferramenta de observabilidade utilizada.
 - **Impacto Esperado:**
 - Pipeline completo de captação de feedbacks ativo.
 - Monitoramento completo da jornada da resposta.
 - Fundamento para dashboards e análises contínuas.
-

Seção: IBS 360 - Gestão do Parque

História: [Delivery][Dados] Inclusão dos Pontos PAE na Base 360

- **Visão de Produto:**

Acreditamos que, ao **incluir os pontos PAE na Base 360**, resultará em **uma visão mais abrangente e completa dos ativos físicos do banco**, permitindo **decisões mais coerentes sobre manutenção, investimentos, reestruturações e monitoramento de performance**.

Saberemos que isso é verdade quando **os dados dos pontos PAE estiverem integrados e disponíveis na Base 360**, podendo ser usados nas análises e dashboards da comunidade.

- **Descrição:**

Como **time responsável pela governança e evolução da Base 360**, queremos **remover o filtro que atualmente exclui os pontos PAE da base**, e garantir que **esses pontos passem a ser incluídos na estrutura e atualização da Base 360**, usando as mesmas fontes de dados já utilizadas para agências físicas.

Essa alteração permitirá **uma visão mais granular e completa da rede**, facilitando comparações, segmentações e priorizações operacionais, especialmente em estudos de footprint, planejamento de capacidade e evolução do parque tecnológico.

- **Visão do Usuário:**

Os **usuários da Base 360** terão agora **visibilidade total dos pontos PAE**, o que viabiliza **ações mais integradas de análise e planejamento**, além de evitar

inconsistências e lacunas em estudos que antes excluía esses pontos automaticamente.

- **Contexto/Narrativa de Negócio:**

Os pontos PAE fazem parte da rede física do banco, mas **até o momento estavam excluídos da Base 360 por um filtro técnico na ingestão**. Isso gerava **incompletude em algumas análises** e dificultava a **avaliação holística da ocupação e da performance da rede**. A inclusão desses pontos **corrige essa lacuna** e alinha a base à realidade operacional do banco.

- **Premissas:**

1. Os dados dos pontos PAE já existem na base de origem (ex: base de agências) utilizada na ingestão atual.
2. A estrutura da Base 360 é compatível com a incorporação dos pontos PAE sem necessidade de refatorações.
3. O time responsável pela ingestão possui controle sobre o filtro de exclusão atual.

- **Regras de Negócio:**

1. O campo `tipo_ponto` deve identificar corretamente os pontos PAE.
2. Nenhum dado de agência já existente deve ser impactado pela mudança.
3. Os pontos PAE devem ser carregados com o mesmo nível de qualidade, granularidade e chaves de identificação que os demais pontos.
 - a. Exceto pelas informações que só são aplicáveis às agências.
4. A inclusão deve ser transparente aos consumidores, sem impacto negativo nas análises atuais.

- **Informações Técnicas:**

- Campo relevante: `tipo_ponto` (ex: `Agência`, `PAE`, `PAB`)
- Fonte de dados: Base já utilizada na ingestão atual
- Alteração: remoção de filtro que exclui `tipo_ponto = PAE`
- Atualização: mesma periodicidade da atualização padrão da Base 360
- Chave de junção: `codigo_agencia`

- **Tarefas:**

1. Localizar o ponto do código responsável pelo filtro que exclui PAE.
 2. Ajustar o pipeline para remover o filtro e permitir ingestão dos pontos PAE.
 3. Validar o carregamento inicial com amostras conhecidas de PAE.
 4. Atualizar documentação técnica da Base 360, incluindo a nova lógica.
 5. Comunicar aos stakeholders a presença dos pontos PAE na base.
 6. Acompanhar uso e eventuais dúvidas ou impactos com os consumidores.
- **Cenários para Teste e Homologação:**
 1. Validar que os pontos PAE estão aparecendo na Base 360 com `tipo_ponto = PAE`.
 2. Verificar se não houve impacto em registros existentes de agências ou PABs.
 3. Validar com consumidores se a nova segmentação está funcionando conforme o esperado.
 - **Critérios de Aceite:**
 - Os pontos PAE foram carregados corretamente na Base 360.
 - Campo `tipo_ponto` diferencia corretamente os tipos de ponto (Agência, PAB, PAE).
 - Filtro antigo foi removido do pipeline de ingestão.
 - Dicionário de dados atualizado com a definição do novo escopo da base.
 - Stakeholders foram comunicados e validaram a nova estrutura.
 - Nenhum campo da base foi corrompido ou perdeu dados anteriores.
 - Nenhum produto teve impacto negativo.
 - **Impacto Esperado:**
 - Visão completa da rede física do banco.
 - Maior capacidade de análise de footprint, planejamento e performance.
 - Correção de inconsistências em dashboards e estudos que excluía PAE por padrão.

- Alinhamento entre os dados e a operação real da rede.
 - **Conclusão**
 - **Início:**
 - **Desejado:** R3 S2 2025
 - **Real:**
 - **Fim:**
 - **Desejado:** R3 S2 2025
 - **Real:**
 - **Resultado:**
-

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

Seção: IBS 360 - Geocompasso

História: [Discovery] Funcionalidades - Prospeção de Folhas

- **Visão de Produto:**

Acreditamos que, ao **explorar uma solução para permitir que o time de folhas realize prospecções geográficas com base em endereços, CEPs ou agências, diretamente no Geocompasso, resultará em maior agilidade, precisão e autonomia na avaliação de oportunidades para o time de Folhas (como a prospecção de folhas INSS).**

Saberemos que isso é verdade quando **validarmos junto ao time de folhas que os critérios de prospecção atualmente respondidos manualmente via planilhas possam ser automatizados via upload de planilha ou seleção interativa de localizações, com outputs dinâmicos no mapa e exportáveis.**

- **Descrição:**

Atualmente o que sabemos:

Como **usuário estratégico do time de folhas, quero subir uma lista de endereços, CEPs ou selecionar agências diretamente no Geocompasso, escolher um raio (ex: 2km, 4km) e receber como retorno a quantidade de agências, TecBans, PABs e PAEs no entorno, para que eu possa avaliar a**

viabilidade de prospecção de novas folhas (ex: Folha da Sabesp) sem depender de análises manuais em Excel.

- **Premissas:**

1. A base de dados com localização geográfica de Tecbans, PABs, PAEs e agências já está disponível na Base 360.
2. O time de folhas realiza hoje esse trabalho manualmente a partir de bases entregues por e-mail.
3. O Geocompasso já possui interface de mapa com possibilidade de seleção e visualização de pontos.

- **Regras de Negócio (a validar no discovery):**

1. Entrada pode ser uma planilha com endereços, CEPs ou Código das agências.
2. Saída esperada: tabela e mapa com a contagem das camadas em raio definido (ex: até 2km, 4km).
3. Deve ser possível escolher múltiplos tipos de camadas a serem considerados.
4. Deve ser possível selecionar se deve considerar apenas o raio ou também incluir no mesmo município.
5. Exportação em planilha deve manter as colunas: ID/Endereço/CEP | Raio (km) | Qtd Agências | Qtd Tecbans | Qtd PABs | Qtd PAEs, considerando o raio, considerando o município.

- **Tarefas:**

1. Realizar entrevista com o time de folhas para detalhar a jornada atual e critérios de decisão.
2. Mapear os formatos de entrada mais comuns (endereços, CEPs, Código das Agências).
3. Validar com o time técnico a viabilidade de matching geográfico e performance do cálculo por batch.
4. Construir protótipo da experiência no Geocompasso (upload + retorno dos dados).
5. Definir modelo de output e formato de exportação desejado.

6. Levantar gaps de dados ou melhorias necessárias na base de equipamentos da Base 360.

- **Critérios de Aceite do Discovery:**

- Jornada de uso atual documentada com exemplos reais de planilhas utilizadas pelo time de folhas.
- Protótipo de fluxo validado com stakeholders.
- Definição clara dos tipos de filtros e raios que deverão ser implementados.
- Definição do escopo mínimo para início da fase de delivery na próxima sprint.

- **Impacto Esperado:**

- Redução drástica do tempo gasto em análises manuais.
- Maior confiabilidade e padronização na prospecção de novas folhas.
- Integração do processo ao fluxo contínuo do Geocompasso, fortalecendo a plataforma como ferramenta central de tomada de decisão territorial.

- **Conclusão**

- **Início:**

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

- **Fim:**

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

- **Resultado:**

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.

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.

Seção: Radar Imobiliário

Tarefa: [Delivery][Documentação] Criação da TechDoc do Radar Imobiliário

Seção: Inteligência Imobiliária

História: [Delivery][Infraestrutura] Criação e Configuração do DynamoDB para Armazenamento de Dados do Planejamento de Negociações

- **Visão de Produto:**

Acreditamos que, ao **criar e configurar a base de dados no DynamoDB para armazenar os dados preenchidos pelo usuário no Frontend do Radar Imobiliário e os tickets de requisições entre APIs**, para **garantir rastreabilidade, persistência e disponibilidade das simulações do modelo de Planejamento de Negociações**, resultará em **uma solução de backend consistente, com histórico acessível e preparada para consulta e auditoria das execuções**. Sabemos que isso é verdade quando **os dados preenchidos e os IDs das requisições forem salvos corretamente no DynamoDB, com chave primária bem definida, e puderem ser consultados, atualizados e correlacionados com logs das Lambdas**.

- **Descrição:**

Como time responsável pelo desenvolvimento técnico do modelo de Planejamento de Negociações, queremos **implementar o banco de dados em DynamoDB**, garantindo que ele **armazene os dados inseridos pelos usuários via Frontend e mantenha o vínculo com os tickets UUID de chamadas às APIs do modelo**, promovendo **rastreabilidade ponta-a-ponta e flexibilidade de acesso no futuro**.

- **Visão do Usuário:**

O usuário não interage diretamente com a base, mas **a experiência será mais fluida e segura, com preenchimentos armazenados automaticamente, simulações rastreáveis e possibilidade de reuso ou auditoria posterior**, sem perda de dados.

- **Contexto/Narrativa de Negócio:**

Com as funções Lambda e o API Gateway já operacionais, a ausência de um banco estruturado representa **risco de perda de dados, dificuldade de debugging e limitação em features futuras como histórico de simulações ou reprocessamento por ticket**. A adoção do DynamoDB permite **escalabilidade, rapidez e fácil integração com os demais componentes AWS do modelo**.

- **Premissas:**

1. As Lambdas que executam o modelo já estão criadas e registram UUIDs por chamada.
2. O Frontend será responsável por enviar os dados preenchidos via chamada autenticada.
3. As requisições conterão um identificador único (UUID) para cada simulação.

- **Regras de Negócio:**

1. Cada item no DynamoDB deve conter: UUID da simulação, dados preenchidos, timestamp, status da requisição.
2. O status da simulação pode ser: `pendente`, `em_processamento`, `concluído`, `erro`.
3. Atualizações devem ser atômicas e controladas apenas pelas Lambdas com permissão específica.

- **Informações Técnicas:**

1. Criação de uma tabela no **DynamoDB** com chave primária `UUID` e índices secundários (ex: por usuário ou timestamp, se necessário).
2. Configuração de **TTL (Time to Live)** para expurgar dados antigos após X dias, se aplicável.
3. Permissões via **IAM Role** apenas para Lambdas autorizadas.
4. Integração com as Lambdas de entrada e processamento para leitura/escrita.
5. Logs de escrita e leitura para rastreabilidade em **CloudWatch Logs**.

- **Tarefas:**

1. **Criação da Tabela DynamoDB**

- Definir estrutura de chave: `UUID` como primária.
- Incluir atributos: dados da simulação, timestamps, status, usuário (opcional).

2. **Configuração de Segurança e Acesso**

- Criar policy IAM exclusiva com permissões de `PutItem`, `GetItem`, `UpdateItem`.
- Atribuir IAM Role às Lambdas que acessam a base.

3. Integração com Lambdas Existentes

- Adaptar Lambda de entrada para salvar dados no DynamoDB.
- Adaptar Lambda de resposta para atualizar status.

4. Testes de Leitura e Escrita

- Simular envio de dados do Frontend → salvar no DynamoDB.
- Simular execução de modelo → atualizar status no DynamoDB.

5. Monitoramento e Logs

- Garantir que logs de acesso sejam registrados.
- Validar integridade dos dados salvos e atualizados.

6. Documentação

- Especificar estrutura da tabela e campos obrigatórios.
- Mapear fluxos de leitura e escrita com exemplos de payloads.

• Cenários para Teste e Homologação:

1. Envio de simulação via painel → persistência no DynamoDB com UUID correto.
2. Atualização do status pela Lambda → visível e consistente na tabela.
3. Consulta por UUID retorna dados completos e atualizados.
4. Testes com erro intencional garantem consistência dos dados (ex: status "erro").
5. Logs no CloudWatch mostram ações completas com rastreabilidade.

• Impacto Esperado:

- Persistência segura e estruturada dos dados de simulação.
- Rastreabilidade ponta-a-ponta entre Frontend, API e execução do modelo.
- Facilidade para reprocessar, auditar e evoluir funcionalidades futuras.
- Conformidade com boas práticas de arquitetura serverless escalável e desacoplada.

• Conclusão

- **Início:**

- **Desejado:** R2 S4 2025
- **Real:** R3 S1 20253
- **Fim:**
 - **Desejado:** R3 S1 2025
 - **Real:**
- **Resultado:**

História: [Delivery][Backend] Enviar Dados de Output para a Lambda POST

- **Visão de Produto:**

Acreditamos que, ao **criar um processo de envio de dados filtrados e tratados para uma Lambda POST**, para **integração com o modelo de otimização do time Advanced Analytics**, resultará em **gatilho automatizado e integração eficiente entre front e motor de decisão**.

- **Descrição:**

Queremos **construir uma Lambda que envie via POST os dados parametrizados pelo usuário**, após filtragem e enriquecimento, para **acionar o modelo de otimização matemática** hospedado no Advanced Analytics.

- **Visão do Usuário:**

O usuário configura os parâmetros, clica em "Executar" e espera o retorno com agilidade e confiabilidade.

- **Contexto/Narrativa de Negócio:**

A comunicação entre nossa plataforma e o modelo externo deve ser confiável, segura e ágil.

- **Premissas:**

- O endpoint da API externa já está definido.
- O modelo espera dados num payload específico.

- **Regras de Negócio:**

- Campos obrigatórios devem estar presentes.
- Em caso de erro, deve haver retry e logging.

- **Informações Técnicas:**

- Lambda em Python.
- API autenticada com token.
- Comunicação HTTPS segura.

- **Tarefas:**

- Criar função Lambda.
- Mapear estrutura do payload.
- Tratar falhas com retry e log.

- **Cenários para Teste e Homologação:**

- Testar integração com endpoint real.
- Simular erros de conexão.
- Validar sucesso com payload mínimo.

- **Impacto Esperado:**

Execução ágil e rastreável de integrações com o modelo.

- **Conclusão**

- **Início:**

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

- **Fim:**

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

- **Resultado:**

História: [Delivery][Backend] Criação do ETL para Geração da Base Filtrada a partir dos Parâmetros do Planejamento de Negociações

- **Visão de Produto:**

Acreditamos que, ao **criar um processo de ETL para filtrar a base original de dados imobiliários com base nos parâmetros preenchidos no Frontend do**

Planejamento de Negociações, para **preparar uma base específica e otimizada para uso pelo time de Advanced Analytics**, resultará em **um processo mais eficiente, seguro e rastreável para execução do modelo de otimização**.

- **Descrição:**

Como time responsável pela engenharia do IBS 360, queremos **desenvolver e operacionalizar um ETL que leia os dados recebidos pelo ETL1 (dados da GPA e Osiris)**, filtre a base imobiliária consolidada no ETL1 e gere **uma base reduzida, segura e rastreável**, que será utilizada nas simulações feitas pelo time de Advanced Analytics.

- **Visão do Usuário:**

O usuário não visualiza diretamente a base filtrada, mas se beneficia de **simulações mais rápidas, coerentes com os filtros aplicados no painel, com rastreabilidade clara entre os dados preenchidos e os resultados retornados**.

- **Contexto/Narrativa de Negócio:**

O ETL criará **bases de trabalho específicas por simulação, permitindo rastreabilidade via UUID e promovendo um fluxo limpo para execução dos modelos**.

- **Premissas:**

1. A base consolidada de imóveis está disponível via ETL1.
2. Os filtros preenchidos estão no frontend.
3. A arquitetura suporta a geração de arquivos versionados por simulação.

- **Regras de Negócio:**

1. A base deve conter somente os imóveis elegíveis conforme os filtros informados.
2. O arquivo final deve estar associado a um UUID de simulação.

- **Informações Técnicas:**

1. Leitura da base ETL1 consolidada.
2. Aplicação dos filtros preenchidos pelo usuário.
3. Logging estruturado no CloudWatch.

- **Tarefas:**

1. Criar script de filtragem.
 2. Escrever base filtrada.
 3. Automatizar a execução com.
 4. Testar fluxos de ponta a ponta com múltiplos cenários.
 5. Documentar estrutura de diretórios, nomenclatura de arquivos e regras aplicadas.
- **Cenários para Teste e Homologação:**
 1. Testar execução com diferentes conjuntos de filtros.
 2. Validar que apenas registros elegíveis estão na base final.
 3. Verificar se o UUID aparece corretamente no nome/pasta.
 4. Confirmar que arquivos são legíveis por processos posteriores (ex: modelos em Python).
 - **Impacto Esperado:**
 - Base otimizada e segura para execução de modelos.
 - Redução de tempo e custo de processamento nas simulações.
 - Garantia de rastreabilidade entre preenchimento → base gerada → resultado.
 - **Conclusão:**
 - **Início:**
 - **Desejado:** R3 S1 2025
 - **Real:**
 - **Fim:**
 - **Desejado:** R3 S1 2025
 - **Real:**
 - **Resultado:**

História: [Delivery][Frontend] Tela de Visualização de Tickets Enviados ao Advanced Analytics

- **Visão de Produto:**

Acreditamos que, ao **criar uma tela de visualização dos envios (tickets) realizados para o modelo de otimização**, para permitir aos usuários acompanhar o status e histórico, resultará em **transparência, rastreabilidade e controle sobre as simulações feitas**.

- **Descrição:**

Construir uma nova aba ou seção no painel exibindo o histórico de execuções, com data, parâmetros utilizados e status de resposta.

- **Visão do Usuário:**

O usuário quer consultar simulações anteriores e entender o que já foi enviado.

- **Contexto/Narrativa de Negócio:**

A rastreabilidade das execuções é essencial para justificar decisões, evitar duplicidade e garantir governança.

- **Premissas:**

- As execuções estão sendo registradas no DynamoDB.
- A API de leitura está disponível.

- **Regras de Negócio:**

- Exibir simulações do usuário autenticado.
- Mostrar status (pendente, processado, erro).

- **Informações Técnicas:**

- Tela em Streamlit conectada ao DynamoDB.
- Leitura com paginação e filtros.

- **Tarefas:**

1. Criar endpoint de leitura.
2. Desenvolver tabela de visualização.
3. Aplicar ordenação por data.

- **Cenários para Teste e Homologação:**

- Testar com usuários diferentes.
- Verificar exibição de tickets reais.

- **Impacto Esperado:**

Transparência total nas simulações e histórico acessível.

- **Conclusão:**

- **Início:**

- **Desejado:** R3 S2 2025

- **Real:**

- **Fim:**

- **Desejado:** R3 S2 2025

- **Real:**

- **Resultado:**

Seção: Score de Agências

História: [Delivery][Infraestrutura] Separação dos Outputs das Bases do Score de Agências em Ambientes Dev, Homologação e Produção

- **Visão de Produto:**

Acreditamos que, ao **separar fisicamente os outputs da Base 360 em ambientes distintos (Desenvolvimento, Homologação e Produção)**, para os **times que desenvolvem, testam e consomem a base em diferentes estágios de maturidade**, resultará em **maior segurança operacional, redução de riscos em publicações acidentais e maior controle sobre o ciclo de vida dos dados**. Sabemos que isso é verdade quando **os dados forem disponibilizados separadamente em três ambientes controlados e rastreáveis, garantindo que cada etapa (dev, hom, prod) tenha seu próprio repositório de outputs**.

- **Descrição:**

Como time responsável pela **gestão e governança da Base 360**, queremos **separar fisicamente os outputs dos dados gerados pela base**, garantindo que:

- **Desenvolvimento** tenha seu próprio repositório de testes e validações técnicas.
- **Homologação** tenha um ambiente estável para validações funcionais e de negócio.

- **Produção** tenha apenas dados aprovados e prontos para uso oficial.

Essa separação será feita inicialmente **por repositórios distintos no SharePoint**, com **uma pasta dedicada para cada ambiente**.

- **Visão do Usuário:**

Os **desenvolvedores, analistas de qualidade e consumidores finais da base** terão **clareza e segurança ao acessar apenas o ambiente correspondente ao seu estágio de trabalho**, evitando **o risco de trabalhar com dados errados ou não validados**.

- **Contexto/Narrativa de Negócio:**

Hoje, **os outputs da Base 360 são gerados em um único repositório**, o que **mistura dados de desenvolvimento, homologação e produção**, aumentando **o risco de exposição de dados não validados** e dificultando **o controle do ciclo de vida da base**. Com a separação em **ambientes dedicados**, será possível **mitigar esses riscos e aumentar a governança sobre a disponibilização dos dados**.

- **Premissas:**

1. A equipe já possui **pasta ou site no SharePoint estruturado para a Base 360**.
2. É possível **criar e gerenciar pastas separadas para dev, hom e prod no SharePoint**.
3. Os processos de geração da Base 360 podem **direcionar os outputs para pastas distintas conforme o ambiente executado**.

- **Regras de Negócio:**

1. Cada ambiente deve **ter um repositório próprio e exclusivo para armazenar os dados**.
2. Os dados de produção só devem ser **gerados e disponibilizados após validação em homologação**.
3. **Apenas usuários autorizados** devem ter acesso ao ambiente de produção.

- **Informações Técnicas:**

1. Configuração de **três repositórios/pastas no SharePoint**:

- `/Score de Agências/Performance/Dev`

- /Score de Agências/Performance/Homologação
 - /Score de Agências/Performance/Produção
 - /Score de Agências/Ecoeficiência/Dev
 - /Score de Agências/Ecoeficiência/Homologação
 - /Score de Agências/Ecoeficiência/Produção
2. Atualização dos **pipelines ou scripts de geração de output**, direcionando para o repositório correto com base no ambiente de execução.
 3. **Governança de acesso**, garantindo que os acessos sejam segregados conforme o ambiente.
- **Tarefas:**
 1. **Configuração das Pastas no SharePoint**
 - Criar ou validar a existência das três pastas (Dev, Homologação, Produção).
 2. **Ajuste nos Pipelines de Geração**
 - Atualizar os scripts e pipelines para direcionar os arquivos para o ambiente correto.
 3. **Governança de Acesso**
 - Definir e aplicar as permissões de acesso para cada pasta.
 4. **Testes de Publicação**
 - Validar que os outputs estão sendo gerados e salvos nas pastas corretas.
 5. **Documentação e Comunicação**
 - Documentar o processo e comunicar o time sobre a nova estrutura.
 - **Cenários para Teste e Homologação:**
 1. Validar que o pipeline de **desenvolvimento salva os outputs apenas na pasta Dev.**
 2. Validar que o pipeline de **homologação salva os outputs apenas na pasta Homologação.**

3. Validar que o pipeline de **produção salva os outputs apenas na pasta Produção.**
 4. Garantir que **os acessos estão segregados conforme o ambiente.**
- **Impacto Esperado:**
 - **Redução de riscos de publicação incorreta de dados.**
 - **Maior controle e governança sobre o ciclo de vida da Base 360.**
 - **Melhor organização dos outputs por ambiente, facilitando validações e homologações.**
 - **Segurança reforçada no acesso aos dados de produção.**
 - **Conclusão**
 - **Início:**
 - **Desejado:** R2 S2 2025
 - **Real:** R2 S2 2025
 - **Fim:**
 - **Desejado:** R2 S2 2025
 - **Real:** R2 S2 2025
 - **Resultado:** A separação foi realizada.
 - Criou-se um sharepoint apenas para as bases do IBS 360.
 - O modelo já começou a ser replicado para as bases do Score.

História: [Delivery][Backend] Aplicação de Regras de Data Quality nos Fluxos ETL do Score de Agências

- **Visão de Produto:**

Acreditamos que, ao **aplicar regras de qualidade de dados (Data Quality) nos pipelines de ETL do Score de Agências**, para **monitorar e mitigar inconsistências e ausência de dados críticos**, resultará em **um Score mais confiável, com maior credibilidade para as áreas de Performance, Eficiência e Governança**, além de **reduzir o esforço manual de validação e retrabalho.**

Saberemos que isso é verdade quando **as quebras forem detectadas automaticamente, registradas em logs ou dashboards de monitoramento, e**

acionarem alertas preventivos antes de impactar o cálculo dos scores ou a visualização para o usuário final.

- **Descrição:**

Como time responsável pelo **Score de Agências**, queremos **implementar validações de qualidade de dados diretamente nos fluxos de ingestão e transformação**, identificando:

- Dados faltantes por KPI (ex: ausência de medição de ATM, energia, consumo de água)
- Dados fora de range esperado (ex: valores nulos, negativos ou discrepantes)
- Ausência de atualização dentro da janela esperada (ex: KPI que deveria ser diário, mas não se atualiza há X dias)
- Inconsistências por agência (ex: agência ativa sem dados de operação)

Essa validação deve ser **automatizada, extensível e integrada aos processos existentes**, com visibilidade técnica e funcional.

- **Visão do Usuário:**

O time de dados e o time de produto **terão visibilidade clara sobre a integridade dos dados do Score antes da publicação**. Os times de Performance e EcoEficiência serão impactados **indiretamente**, com maior confiabilidade no produto e menor exposição a análises equivocadas baseadas em dados corrompidos ou ausentes.

- **Contexto / Narrativa de Negócio:**

Em análises recentes e sprints anteriores, foram identificados **vários casos de scores calculados incorretamente por ausência ou erro de dados em KPIs específicos**. A falta de alertas e rastreabilidade dificultou a correção proativa e causou **dúvidas por parte dos stakeholders**. Esta entrega busca **iniciar uma cultura de data quality nativa no produto**, permitindo detecção antecipada e evolução do monitoramento técnico.

- **Premissas:**

1. Os pipelines de ingestão/transformation estão sob gestão do time técnico.
2. Existe entendimento técnico e de negócio sobre o que configura um dado "válido" para cada KPI.

3. Os fluxos permitem inclusão de etapas intermediárias para validação e logging sem comprometer performance.

- **Regras de Negócio:**

1. Cada KPI deve ter regras mínimas de validade: presença, formato, range, frequência de atualização.
2. Quebras devem ser registradas com:
 - Nome do KPI
 - Agência impactada
 - Tipo de erro
 - Timestamp da verificação
3. Os erros devem ser consolidados em um painel interno ou exportados para logs do observability.
4. A pipeline não deve falhar por erro de dados, mas o erro deve ser sinalizado.

- **Informações Técnicas:**

1. As validações serão aplicadas em camada intermediária nos scripts do ETL.
2. Os resultados poderão ser registrados em:
 - Tabela de controle localmente.
 - Futuramente deseja-se armazenar os logs no CloudWatch.
3. Alertas críticos (ex: quebra de mais de 20% das agências para um KPI) poderão acionar notificações automatizadas (e-mail, Teams, Slack).

- **Tarefas:**

1. Mapear KPIs críticos e suas regras de qualidade.
2. Criar camada de validação nos pipelines.
3. Implementar mecanismo de logging e registro das quebras.
4. Realizar testes com dados reais e cenários de quebra simulados.
5. Publicar documentação das regras implementadas.

6. Apresentar ao time de governança e negócio os resultados do monitoramento.

- **Cenários para Teste e Homologação:**

1. Simular dados faltantes para uma agência e validar detecção.
2. Injetar valores fora de range e verificar se são sinalizados.
3. Interromper envio de dados por mais de X dias e verificar alertas.
4. Validar que a pipeline continua funcional mesmo com erros detectados.

- **Impacto Esperado:**

- Redução de erros silenciosos nos scores.
- Aumento da confiança de stakeholders nos dados exibidos.
- Capacidade de ação rápida sobre problemas sistêmicos de ingestão.
- Evolução contínua da maturidade de qualidade de dados no produto.
- Prevenção de incidentes e retrabalhos associados à análise de dados incorretos.

- **Conclusão:**

- **Início:**
 - **Desejado:** R2 S4 2025
 - **Real:**
- **Fim:**
 - **Desejado:** R2 S4 2025
 - **Real:**
- **Resultado:**

História: [Delivery][Dados] Reprocessamento do Score de Agências com Base nos Dados de Backup

- **Visão de Produto:**

Acreditamos que, ao **reprocessar o Score de Agências utilizando os dados de backup da base histórica validada**, para **corrigir erros identificados nos dados oriundos do Saúde das Agências (referentes à contabilização incorreta de semanas)**, resultará em **recuperação da confiabilidade dos**

resultados e mitigação de impactos em análises e decisões baseadas nesses scores.

Saberemos que isso é verdade quando **o reproprocessamento for concluído com sucesso, os scores forem recalculados com os dados corretos, e os outputs forem atualizados nos repositórios oficiais (Dev, Homologação, Produção).**

- **Descrição:**

Como time responsável pelo **Score de Agências**, queremos **realizar um reproprocessamento completo dos scores, utilizando como insumo os dados de backup validados previamente**, corrigindo um erro identificado no input do Saúde das Agências, onde **as semanas estavam sendo contabilizadas de forma incorreta, afetando os cálculos de desempenho e disponibilidade.**

- **Contexto / Narrativa de Negócio:**

Durante análises internas, foi identificado que os **dados provenientes do Saúde das Agências estavam incorretos, com erro na contagem de semanas**, gerando **distorsões nos scores consolidados**. A existência de um **backup validado da base histórica** permite reverter o impacto por meio de um reproprocessamento completo e controlado.

- **Premissas:**

1. O backup utilizado está armazenado em ambiente seguro e já foi validado previamente.
2. As pipelines de cálculo permitem reproprocessamento com nova fonte de dados sem alterar a lógica de cálculo.
3. O reproprocessamento pode ser realizado em ambiente isolado antes da publicação.

- **Regras de Negócio:**

1. O reproprocessamento deve ser realizado apenas com dados de backup previamente aprovados.
2. Os resultados reproprocessados devem substituir os dados com erro nos repositórios oficiais.
3. Todos os dados antigos com erro devem ser mantidos em backup, com rastreabilidade.

- **Informações Técnicas:**

1. As pipelines devem ser executadas apontando para o backup como nova fonte de input.
2. Após validação técnica e funcional, os dados devem ser movidos para as pastas de output (**Dev** , **Hom** , **Prod**) com versionamento.

- **Tarefas:**

1. Validar integridade dos dados de backup.
2. Reconfigurar pipelines para reproprocessamento com base no backup.
3. Executar os cálculos de score com nova fonte.
4. Realizar validações técnicas e funcionais dos resultados.
5. Atualizar os repositórios com os novos outputs.
6. Comunicar as áreas impactadas sobre a atualização e correção.

- **Cenários para Teste e Homologação:**

1. Verificar se os novos scores corrigem os desvios causados pelo erro de semanas.
2. Comparar resultados antes/depois e validar com analistas de negócio.
3. Garantir que os dados corrigidos não impactam negativamente as pipelines de produção.
4. Validar que os dados antigos foram arquivados com rastreabilidade.

- **Impacto Esperado:**

- Recuperação da confiança dos usuários no Score de Agências.
- Correção de distorções nas análises e decisões geradas por dados errados.
- Reforço na governança e rastreabilidade dos dados.
- Redução de riscos de exposição de dados inconsistentes.

- **Conclusão:**

- **Início:**

- **Desejado:** R3 S2 2025

- **Real:**

- **Fim:**

- **Desejado:** R3 S2 2025
 - **Real:**
 - **Resultado:**
-

Seção: App Planejamento de Pessoas

Seção: Arquitetura Cross

Seção: Monitoramento e Métricas de Produto

Seção: AWS CloudBridge