

# [C26002-ATVX]

## EscopoVix

EscopoVIX é uma solução baseada em Inteligência Artificial criada para automatizar a análise de solicitações recebidas por e-mail. Utilizando modelos de linguagem avançados, o sistema interpreta o conteúdo das mensagens e seus anexos, comparando-os com a base de serviços oferecidos pela empresa. Dessa forma, ele classifica automaticamente as demandas como dentro ou fora do escopo de atendimento, auxiliando a área comercial na triagem de oportunidades e reduzindo o tempo gasto com análises manuais.

- [C26002-ATVX](#)
- [Estudos](#)
  - [Agents](#)
  - [Tokens em Modelos de Linguagem e sua Importância para Agentes de IA](#)
  - [Pensamento-Ação-Observação: ReAct](#)
- [ATA](#)
  - [ATA - 23/03/2026](#)
  - [ATA - 30-03-2026](#)
  - [ATA - 06/05](#)
  - [ATA 20/05](#)
- [Levantamento de requisitos - Modelo Atual](#)

# C26002-ATVX

## 1. Visão Geral

O objetivo do projeto é validar automaticamente se uma solicitação recebida por e-mail está dentro ou fora do escopo de atendimento da empresa.

Essa validação é realizada por meio da análise do conteúdo do e-mail e de seus anexos, utilizando um modelo de linguagem (LLM) que compara a solicitação com uma base de serviços prestados pela empresa, armazenada em uma planilha estruturada.

O sistema utiliza como base:

- Conteúdo textual do e-mail
  - Anexos enviados pelo solicitante
  - Histórico de padrões de solicitações
  - Base de serviços da empresa (BUs)
- 

## 2. Problema Resolvido

O sistema automatiza a análise preliminar normalmente realizada por um atendente comercial.

Antes da automação, a equipe precisava:

- Ler manualmente cada e-mail recebido
- Analisar anexos
- Comparar a solicitação com o portfólio de serviços da empresa

Esse processo gerava:

- Alto consumo de tempo operacional
- Análise manual repetitiva
- Possível atraso na resposta ao cliente

Com a solução implementada, o sistema realiza uma classificação inicial automática, reduzindo significativamente o tempo gasto na triagem de propostas fora do escopo.

---

# 3. Arquitetura da Solução

A solução é composta por diferentes componentes responsáveis pela ingestão, armazenamento, processamento e análise dos dados.

## Componentes da Arquitetura

### GitHub

Repositório responsável por armazenar os scripts de transformação de arquivos anexos em formato textual.

As transformações incluem:

- `.zip` → descompactação e extração de conteúdo
- `.html` → extração do corpo textual
- `.pdf` → conversão para texto
- `.excel` → conversão para `.csv`

Essa etapa garante que os dados possam ser interpretados pelo modelo de linguagem.

---

### Power Automate

Responsável pela captura automática dos e-mails recebidos na caixa:

`comercial@autvix.com.br`

Ao detectar um novo e-mail, o fluxo automatizado:

1. Cria uma pasta no OneDrive
2. Salva o corpo do e-mail
3. Salva todos os anexos

*A estrutura de armazenamento segue o padrão:*

```
<Hora>_<Minuto>$<Data>$<Remetente>
```

Exemplo:

```
12_47$13_03_2026$petronect@petronect.com.br
```

*Dentro da pasta são armazenados:*

corpo\_email.html

anexos enviados pelo remetente

---

## OneDrive

Funciona como camada de armazenamento intermediária do pipeline.

Todos os e-mails processados ficam armazenados nesta estrutura de pastas, permitindo:

- rastreabilidade
- histórico de análise
- sincronização com o servidor de processamento

---

## Servidor Ubuntu

Servidor responsável por hospedar o modelo de linguagem utilizado para análise dos e-mails.

Funções do servidor:

- sincronizar dados do OneDrive
- realizar transformação de arquivos
- executar inferência do modelo de IA
- classificar a solicitação

A sincronização dos arquivos é realizada através da ferramenta onedrive.

---

## Modelo de Inteligência Artificial

O sistema utiliza o modelo:

gpt-oss:120b

O modelo recebe como entrada:

- conteúdo do e-mail
- conteúdo dos anexos convertidos em texto
- base de serviços da empresa (BUS) em formato `.csv`
- Histórico de solicitações
- Prompt agent, um prompt que descreve com detalhes a forma na qual o modelo de LLM deve operar

A partir dessas informações, o modelo avalia se a solicitação:

- está dentro do escopo de atendimento
  - está fora do escopo
  - não representa uma solicitação válida
- 

## 4. Fluxo do Processo

1. Um e-mail é recebido na caixa [comercial@autvix.com.br](mailto:comercial@autvix.com.br) e [comercial@advixsolucoes.com.br](mailto:comercial@advixsolucoes.com.br)
2. O Power Automate detecta automaticamente o novo e-mail.
3. O fluxo cria uma pasta no OneDrive contendo:
  - corpo do e-mail
  - anexos enviados
4. O servidor Ubuntu sincroniza automaticamente os dados utilizando onedrive.
5. Os arquivos são processados e convertidos para texto.
6. O conteúdo do e-mail e anexos é enviado ao LLM junto com a base de serviços da empresa (BUS), bem como o prompt agent e histórico de solicitações.
7. O modelo analisa a solicitação e realiza a classificação.
8. O sistema retorna uma decisão preliminar:
  - Aceitar prospecção (dentro do escopo)
  - Recusar educadamente (fora do escopo)

## 5. Classificação dos E-mails

Os e-mails são classificados em quatro categorias principais.

### Dentro do Escopo

Solicitações compatíveis com os serviços oferecidos pela empresa.

Exemplos:

- pedidos de proposta
- solicitações técnicas relacionadas aos serviços da empresa

Resultado:

Prosseguimento da prospecção comercial.

---

## Fora do Escopo

Solicitações que não fazem parte do portfólio de serviços da empresa.

Exemplos:

- serviços que a empresa não oferece
- demandas fora da área de atuação

Resultado:

Resposta automática recusando educadamente a solicitação.

---

## Não é Solicitação

E-mails que não representam uma demanda comercial.

Exemplos:

- convites
- newsletters
- comunicações institucionais

Resultado:

Ignorar ou arquivar.

---

## Oportunidade Externa

Quando a solicitação não está diretamente no e-mail, mas sim em:

- links
- portais externos
- plataformas de licitação

Resultado:

Encaminhamento para análise manual.

---

# 6. Tecnologias Utilizadas

## Infraestrutura

- Ubuntu Server
- Microsoft 365
- OneDrive

## Automação

- Power Automate

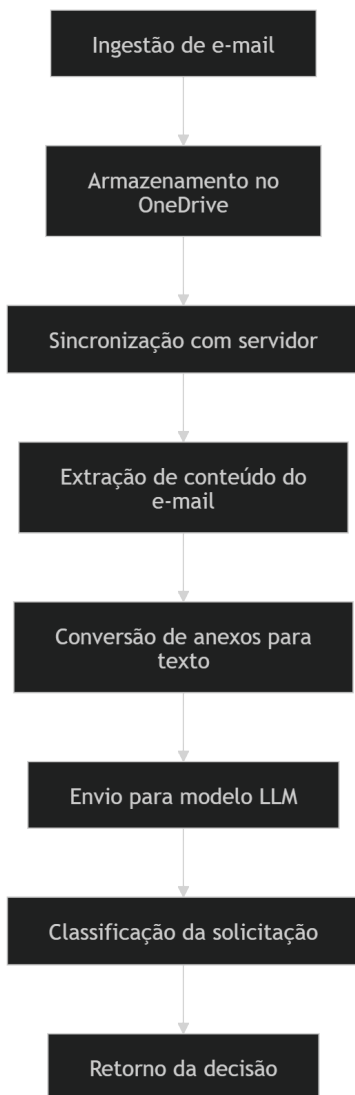
## Repositório de código

- GitHub

## Inteligência Artificial

- gpt-oss:120b
- 

# 7. Pipeline de Processamento



---

## 8. Pontos de Atenção / Limitações

Algumas limitações do sistema incluem:

- Dependência da qualidade do texto extraído dos anexos
- Possível ambiguidade na interpretação de solicitações
- Necessidade de atualização constante da base de serviços (BUS)
- Carga computacional elevada para execução de modelos de grande porte
- Casos complexos podem exigir validação humana

---

## 9. Melhorias Futuras

Possíveis evoluções da solução:

- Implementação de RAG (Retrieval Augmented Generation) para melhorar a consulta da base de serviços
- Criação de dashboard de monitoramento das classificações
- Treinamento ou ajuste fino do modelo com histórico de e-mails da empresa
- Integração com CRM comercial (Sankya)



# Estudos

Estudos

# Agents

# Tokens em Modelos de Linguagem e sua Importância para Agentes de IA

## 1. O que são tokens

Tokens são as menores unidades de texto que um modelo de linguagem consegue processar.

Um texto como:

```
"Aprender IA é incrível!"
```

Pode ser transformado em algo como:

```
["Aprender", " IA", " é", " incrível", " !"]
```

Ou até em pedaços menores, dependendo do modelo.

### **Importante:**

O modelo não entende palavras diretamente — ele entende **tokens**.

---

## 2. Tipos de tokens

### 2.1 Tokens comuns (tokens de texto)

São os tokens que representam conteúdo normal:

- palavras
- partes de palavras
- pontuação

Exemplo:

```
"inteligência" → ["inteli", "gência"]
```

## 2.2 Tokens especiais

São tokens que **não representam texto comum**, mas sim **estrutura ou controle**.

Eles funcionam como instruções internas para o modelo.

# 3. Principais tipos de tokens especiais

## 3.1 BOS (Beginning Of Sequence)

Indica o início da entrada.

```
<bos>
```

Serve para avisar ao modelo:

“Aqui começa o conteúdo”

## 3.2 EOS (End Of Sequence)

Indica o fim da geração.

```
<eos>
```

Função:

- Diz ao modelo quando parar
- Evita geração infinita

**Analogia:**

É como o ponto final de uma frase, mas com poder de encerrar completamente a resposta.

---

## 3.3 PAD (Padding)

Usado para completar sequências até um tamanho fixo.

```
<pad>
```

Muito usado em treinamento.

---

## 3.4 Tokens de papel (role tokens)

Usados em chats:

```
<user><assistant><system>
```

Servem para indicar:

- quem está falando
  - contexto da conversa
- 

## 3.5 Tokens delimitadores

Alguns modelos usam formatos próprios:

```
[INST] ... [/INST]
```

ou

```
<s> ... </s>
```

Eles delimitam blocos de instrução.

---

## 4. Como isso funciona em um modelo de chat

Por trás de uma conversa simples, o modelo recebe algo como:

```
<system> Você é um assistente útil </system><user> O que é IA? </user><assistant>
```

O modelo então completa:

```
IA é o campo da computação que... <eos>
```

---

## 5. Por que tokens especiais são importantes

### 5.1 Estrutura da conversa

Sem tokens especiais:

```
Usuário: Oi Assistente: Olá Usuário: Tudo bem?
```

Para o modelo isso vira um texto confuso.

Com tokens:

```
<user> Oi </user><assistant> Olá </assistant><user> Tudo bem? </user>
```

Agora há estrutura clara.

---

### 5.2 Controle da geração

O token `<eos>` permite:

- parar respostas automaticamente
- evitar loops infinitos
- melhorar performance

---

## 5.3 Definição de comportamento

O token `<system>` pode mudar completamente o modelo:

```
<system> Seja formal </system>
```

Isso altera o estilo da resposta.

---

## 5.4 Compatibilidade entre modelos

Cada modelo tem seu próprio formato.

Exemplo:

- Llama → usa `[INST]`
- OpenAI → usa roles (system/user/assistant)
- Outros → usam JSON interno

Por isso existem **chat templates**:

“ eles adaptam a entrada para o formato correto do modelo

---

# 6. Importância para agentes de IA

Agentes de IA dependem fortemente desses tokens porque:

## 6.1 Mantêm contexto

Permitem separar:

- instruções
  - histórico
  - respostas
- 

## 6.2 Controlam comportamento

O agente pode:

- mudar personalidade
- seguir regras
- executar tarefas específicas

Tudo via estrutura de tokens.

---

## 6.3 Evitam ambiguidade

Sem tokens especiais, o modelo pode:

- confundir quem está falando
  - responder de forma incoerente
- 

## 6.4 Permitem automação

Ferramentas, memória e raciocínio estruturado dependem de:

- delimitação clara
  - início/fim de blocos
  - controle de geração
- 

# 7. Insight avançado

Tokens especiais são uma forma de:

“programar o modelo usando texto”

Eles funcionam como uma camada de controle sem precisar alterar o código do modelo.

---

## 8. Conclusão

- Tokens são a base do funcionamento dos LLMs
- Tokens especiais organizam e controlam o comportamento
- O token EOS é essencial para indicar o fim da resposta
- Em agentes de IA, eles são fundamentais para:
  - contexto
  - estrutura
  - controle
  - previsibilidade

Estudos

# Pensamento-Ação- Observação: ReAct

# ATA

Este capítulo contém as ATAS das reuniões realizadas com os stakeholders do projeto.

# ATA - 23/03/2026

## Ata de Reunião

Data da Reunião: 23/03/2026

Pauta: Definição do fluxo inicial do projeto, requisitos técnicos, classificação de e-mails e organização do processo comercial

Participantes: Lucas C. Arruda, Gustavo Schwartz, Marcos Monfardini, Vinícius Caetano, Wanderson Silva e Romeu Roveda

---

### ☐ Objetivos

- Definir os requisitos mínimos para início do desenvolvimento do projeto
  - Estruturar o fluxo de processamento de e-mails e oportunidades
  - Alinhar a estratégia de implementação local e segurança dos dados
  - Demonstrar a evolução do projeto ao longo das semanas
- 

### ☐ Acompanhamento e Definições

- O projeto será desenvolvido 100% local, visando maior segurança e controle de dados
- O fluxo inicial parte da caixa de e-mails, com automação para coleta e organização das mensagens
- Cada e-mail será armazenado em uma estrutura de pastas com identificação por data, horário e remetente
- Os e-mails poderão conter documentos (ex: editais), geralmente com:
  - Resumo
  - Documento técnico (quando disponível)
- Será implementado um processo de classificação automática de e-mails, distinguindo:
  - Oportunidades (propostas/prospecções)
  - Mensagens informativas ou atualizações
- Alguns portais (ex: Petronet) possuem características específicas e poderão ter fluxos diferenciados
- O sistema irá gerar uma fila de oportunidades qualificadas, reduzindo o trabalho manual de análise
- O processo atual será gradualmente automatizado, priorizando o fluxo inicial
- Foi identificado que diferentes usuários possuem formas distintas de trabalho, dificultando padronização total
- O foco inicial será nos principais canais e fluxos mais recorrentes
- O time está remodelando o fluxo comercial para um formato:
  - Mais simples

- Sequencial
- 

## ☐ Encaminhamentos

- Code - Definir regras de extração de dados dos documentos (editais/resumos)
  - Code - Avaliar fluxos específicos para portais com comportamento diferenciado
  - Todos - Realizar reuniões semanais com registro de atas e gravações
  - Equipe - Conversar com stakeholders (Victor, Juliana, etc.) e registrar informações para base de conhecimento
  - Romeu - Disponibilizar as BUs da ADVIX
  - Marcos - Fornecer a base de clientes
  - Wanderson - Liberar o acesso para os portais da petronet
- 

## ☐ Observações Finais

- Reuniões semanais foram sugeridas para refinamento contínuo do projeto
- É importante registrar todas as discussões e decisões, inclusive gravações
  - As informações coletadas com usuários devem ser analisadas com cautela
- O projeto está em fase de estruturação inicial, com foco em gerar valor rapidamente com automações simples
- Comunicação contínua via WhatsApp foi recomendada para dúvidas e alinhamentos rápidos

# ATA - 30-03-2026

Ata de Reunião

**Data da Reunião:** 30-03-2026

**Pauta:** Validação de fluxograma, análise de requisitos para dashboard e definição de acessos aos portais

**Participantes:** Lucas C. Arruda, Victor, Wanderson, Marcos, Gustavo e Romeu

---

## ☐ **Objetivos**

- Validar o novo fluxograma do processo comercial
  - Levantar e analisar requisitos para implementação do dashboard
  - Definir data para alinhamento de acesso aos portais e sistemas
- 

## ☐ **Acompanhamento e Definições**

- Foi apresentado um novo fluxograma, dividido entre Autvix e ADVix, visando maior clareza nas particularidades de cada área
  - Definido que a análise de oportunidades externas deve ocorrer antes da leitura completa dos anexos dos e-mails
  - Identificada a necessidade de explicitar o conhecimento tácito do time (principalmente do analista comercial) para alimentar corretamente o modelo de IA
  - Destacado que o processo atual depende fortemente da experiência individual (ex: Victor), sendo necessário documentar critérios de análise (escopo, palavras-chave, tipo de solicitação, etc.)
  - Reforçada a importância de criar contexto detalhado para evitar divergência entre expectativa e entrega, principalmente no uso de IA
  - Sugerido uso inicial de filtros simples (palavras-chave) antes de avançar para modelos mais complexos
  - Definido que futuramente haverá um banco de oportunidades para comparação e análise por similaridade
  - Apresentada arquitetura técnica utilizando grafos (LangGraph), processamento em JSON e integração com OneDrive
  - Confirmado uso de modelos de IA (GPT e outros especializados) com possibilidade de refinamento conforme necessidade
  - Identificada limitação de execução local devido ao uso de modelos pesados, sendo necessário ambiente em servidor
- 

## ☐ **Encaminhamentos**

- **Victor** – Documentar e compartilhar o processo de análise de e-mails e critérios de decisão
  - **Wanderson** – Realizar explicação detalhada do fluxo completo e regras de negócio para o time
  - **Equipe Técnica** – Estruturar regras e contexto para treinamento da IA com base no conhecimento do time
  - **Equipe Geral** – Agendar nova reunião para aprofundamento do fluxo completo e alinhamento técnico
  - **Code** - Criar banco de dados para alimentação do dashboard
  - **Code** - Criar dashboard
- 

## ☐ **Observações Finais**

- O sucesso da automação depende diretamente da externalização do conhecimento dos especialistas
- Há necessidade de transformar conhecimento tácito em documentação estruturada
- O projeto evoluirá de regras simples para modelos mais avançados com base em histórico de dados
- Próxima reunião deverá focar na explicação completa do fluxo e detalhamento das regras de negócio para implementação da IA

# ATA - 06/05

## ATA DE REUNIÃO

**Projeto:** C26002-ATVX - Implantação de IA no Comercial

Campo	Informação
Data	06 de maio de 2026
Horário	17h01
Duração	2h08min28s
Formato	Reunião remota com compartilhamento de tela
Participantes identificados	Lucas C. Arruda - Autvix Group; Gustavo L. Schwartz - Autvix Group
Fonte	Transcrição da gravação da reunião

### 1. Objetivo da reunião

Revisar o estado atual do sistema de implantação de IA no Comercial, incluindo arquitetura, requisitos funcionais e não funcionais, modelo de dados, fluxo de ingestão de e-mails, observabilidade, dashboards e evolução dos filtros de classificação. A reunião também teve como objetivo definir ajustes técnicos, prioridades de implementação e próximos prazos do projeto.

### 2. Pauta tratada

- Apresentação do fluxo atual de captura, sincronização, validação e armazenamento de e-mails.
- Revisão dos requisitos funcionais e não funcionais já documentados ou implementados.
- Discussão sobre a complexidade atual envolvendo OneDrive, Fuse Poller, Watch/Inotify, Power Automate, MySQL, Metabase e Grafana.

- Análise de inconsistências na base, especialmente duplicidade de IDs de mensagens e definição do `folder_path` como referência única.
- Revisão do modelo de dados, com foco na tabela principal de e-mails, contatos, clientes e classificação.
- Avaliação dos testes de classificação de IA, métricas de acurácia e matriz de confusão.
- Definição dos próximos passos, responsáveis e prazos para migração, testes e entregáveis.

## 3. Principais discussões

### 3.1 Arquitetura e fluxo atual

Foi revisado que o fluxo atual envolve captura de e-mails, geração de pastas, sincronização via OneDrive, detecção por mecanismo baseado em Inotify/Fuse Poller, validação dos dados e inserção em banco MySQL. Também foram citados Metabase e Grafana para visualização e observabilidade. Foi destacado que o fluxo atual funciona, mas possui uma cadeia de serviços extensa e complexa.

### 3.2 Complexidade técnica e necessidade de simplificação

Lucas destacou que o projeto apresenta profundidade técnica em pontos como logs, migrations e observabilidade, mas ainda possui pendências em elementos essenciais do entregável, como estruturação definitiva da base e do filtro principal. Foi defendida a priorização do esqueleto funcional e entregável ao cliente antes de novas camadas de refinamento técnico.

### 3.3 Migração para Microsoft Graph API / Power Automate

A equipe convergiu que a migração do fluxo atual para uma abordagem baseada em Microsoft Graph API, possivelmente via Power Automate, deve reduzir a complexidade lógica e operacional. A substituição reduziria dependências de OneDrive, Fuse Poller e Watch/Inotify, além de facilitar rastreabilidade e confiabilidade do processamento.

### 3.4 Revisão do banco de dados

A tabela de e-mails foi tratada como principal por valor agregado, pois concentra as informações recebidas e serve como base para auditoria, dashboards e relacionamento com classificações. Foi reforçado que o banco não deve ser o mecanismo principal de processamento do modelo, mas sim um repositório histórico, de auditoria e análise futura.

## 3.5 Campos da tabela de e-mails

Foi discutida a manutenção de campos como ID, `contato_id`, `classificacao_id`, destinatário, `subject`, data de recebimento, data de processamento, `folder_path`, quantidade de anexos, nomes dos anexos e ID da mensagem. Foi sugerida a remoção de `body_html` e `body_text`, pois os arquivos correspondentes já ficam salvos na pasta, evitando duplicidade e ambiguidade.

## 3.6 Contatos, clientes e domínios

Foi diferenciada a entidade contato da entidade cliente: contato representa quem envia e-mails, enquanto cliente representa a base de clientes cadastrados. Ficou indicado que os domínios de remetentes devem ser relacionados posteriormente a clientes, com apoio de pessoas da área comercial, já que a equipe técnica não possui domínio completo para essa vinculação.

## 3.7 Classificação e filtros de IA

Lucas apresentou testes com regressão logística, banco vetorial, LLM e combinação dos modelos. Foi mencionada acurácia combinada em torno de 91% em amostra de teste e acurácia de 96% na regressão logística sobre 1.097 dados. Também foi reforçada a necessidade de testar com dados não usados no treinamento para validar a real capacidade de generalização.

## 3.8 Métricas, dashboards e observabilidade

A dashboard é considerada requisito do sistema para visualização dos e-mails e indicadores. Foram discutidas métricas como quantidade de anexos, nomes/extensões de arquivos, status de processamento e saúde dos serviços. A observabilidade atual usa Grafana e registros em tabela de status dos serviços.

## 3.9 Integrações externas e limitações

Foi comentada a limitação relacionada a APIs de portais externos, especialmente Petrobras/Petronet. A equipe reconheceu que parte desses problemas depende de terceiros, mas

que será necessário acompanhar e cobrar resolução para viabilizar etapas futuras.

## 4. Decisões e alinhamentos

1. Priorizar a entrega funcional do projeto antes de aprofundar novas camadas de refinamento técnico.
2. Migrar o fluxo atual para Microsoft Graph API/Power Automate, substituindo dependências excessivas de OneDrive, Fuse Poller e Watch/Inotify sempre que possível.
3. Tratar a tabela de e-mails como a tabela principal para histórico, auditoria, dashboards e relacionamento com classificações.
4. Manter o `folder_path` como referência para localização dos arquivos do e-mail e para consultas futuras, especialmente após classificação.
5. Revisar a base de dados para eliminar inconsistências, especialmente duplicidade de `message_id`, reconstruindo a base se necessário a partir dos dados salvos.
6. Remover ou deixar de utilizar campos redundantes como `body_html` e `body_text` enquanto não houver necessidade clara de uso no banco.
7. Distinguir contato de cliente no modelo de dados, mantendo contato como remetente/domínio e cliente como entidade de cadastro comercial.
8. Delegar a vinculação de domínios a clientes para pessoas com conhecimento comercial, como Juliana e/ou Vidal, por meio de planilha de validação.
9. Focar inicialmente na classificação “solicitação” versus “não solicitação”, antes de aprofundar o filtro de dentro/fora do escopo.
10. Reexecutar testes da pipeline de classificação com dados não utilizados no treinamento, para medir acurácia de forma mais confiável.

## 5. Encaminhamentos e responsáveis

#	Ação	Responsável	Prazo
1	Implementar a migração do fluxo atual para Graph API/Power Automate, reduzindo dependências de OneDrive/Fuse/Watch.	Gustavo	Até 07/05/2026, 17h
2	Recarregar ou reconstruir a base de e-mails, validando duplicidades e consistência do <code>message_id</code> e do <code>folder_path</code> .	Gustavo	Após migração para Graph API

#	Ação	Responsável	Prazo
3	Revisar o modelo de dados da tabela de e-mails, removendo campos redundantes e mantendo os campos essenciais para auditoria e dashboard.	Lucas e Gustavo	Curto prazo
4	Exportar lista de domínios/contatos e solicitar validação de associação com clientes à área comercial.	Gustavo, com apoio de Juliana/Vidal	A definir
5	Revisar prompt/system prompt e repetir testes da pipeline de classificação.	Lucas	Antes da próxima validação técnica
6	Separar dados não utilizados no treinamento para nova rodada de testes de acurácia.	Lucas, com apoio operacional a definir	Curto prazo
7	Iniciar desenvolvimento de novo filtro após estabilização da ingestão e validação inicial da classificação.	Lucas	Após testes e base ajustada
8	Buscar bios/dados da Davis/Multimix ou focar inicialmente em uma única empresa para simplificar o recorte do teste.	Lucas/Romeu	Até 11/05/2026, se possível
9	Acompanhar pendências com APIs externas, especialmente Petronet/Petrobras.	Lucas	Contínuo
10	Preparar dashboard robusta para apresentação, com base nos dados disponíveis.	Gustavo	Até 11/05/2026

## 6. Prazos combinados

Data	Entrega/Marco
07/05/2026, até 17h	Conclusão estimada da migração inicial para Graph API/Power Automate e ajustes de base.
11/05/2026, até 17h	Meta de ter pré-classificador em conjunto com a base ajustada; possibilidade de dashboard robusta.

Data	Entrega/Marco
29/05/2026	Meta indicativa para ter algo rodando no Comercial. O prazo foi reconhecido como desafiador, especialmente para a parte de IA.

## 7. Pontos em aberto

- Confirmar o desenho final da arquitetura após a migração para Graph API/Power Automate.
- Validar se o ID da mensagem será único em todos os cenários e como será tratado quando houver múltiplos destinatários.
- Definir com precisão quais campos permanecerão nas tabelas de e-mails, contatos, clientes e classificação.
- Confirmar o mecanismo final de entrega dos resultados classificados ao cliente: CRM, plataforma futura, dashboard ou outro canal.
- Avaliar a viabilidade de integração com portais externos e contornar limitações de APIs de terceiros.
- Definir quem fará a rotulagem/validação de dados não treinados para os próximos testes.

## 8. Encerramento

A reunião foi encerrada após a definição dos próximos passos técnicos, com foco na simplificação da arquitetura, revisão do modelo de dados, melhoria da confiabilidade da ingestão de e-mails e continuidade dos testes de classificação. A próxima etapa prática é a migração para Graph API/Power Automate, seguida da validação da base e avanço do pré-classificador.

---

Ata elaborada a partir da transcrição da reunião.

# ATA 20/05

## Ata de Reunião — Implantação de IA no Comercial

**Projeto:** C26002-ATVX — Implantação de IA no Comercial

**Data:** 20/05/2026

**Duração:** 1h13min35s

**Participantes:**

- Lucas C. Arruda — Autvix Group
- Gustavo L. Schwartz — Autvix Group

**Fonte:** Transcrição da reunião “Reunião em [C26002-ATVX] Implantação de IA no Comercial”.  
:contentReference[oaicite:0]{index=0}

---

## 1. Objetivo da reunião

Alinhar o andamento do projeto de IA aplicado ao processo comercial, com foco em:

- processamento e classificação de e-mails;
  - uso de LLM e regressão logística;
  - tratamento das solicitações vindas da Petronect;
  - diferenciação entre oportunidades públicas, convites e dispensas;
  - definição do escopo mínimo viável do MVP;
  - próximos passos técnicos para implementação.
-

## 2. Pontos discutidos

### 2.1 Benchmark e avaliação dos modelos

Foi discutida a necessidade de comparar diferentes modelos utilizados no processamento das solicitações.

O foco da prova de conceito, neste momento, não é avaliar se a resposta está “bonita” ou bem redigida, mas sim se ela é **verídica e correta** em relação ao conteúdo analisado.

Foi mencionado que as saídas devem seguir um formato padronizado, preferencialmente em **JSON**, para facilitar o tratamento posterior via código, regex ou integração com outros componentes do sistema.

Também foi levantada a necessidade de separar os resultados por modelo, criando pastas ou estruturas distintas para permitir comparação entre as execuções.

---

### 2.2 Fluxo atual de processamento de e-mails

Foi explicado que o sistema atual possui um endpoint responsável por receber o caminho da pasta do e-mail processado.

O fluxo descrito foi:

1. receber o caminho da pasta do e-mail;
2. localizar o arquivo HTML;
3. extrair o texto do e-mail;
4. aplicar filtros iniciais;
5. verificar se é e-mail interno;
6. verificar se é resposta;
7. classificar se é ou não uma solicitação;
8. aplicar modelos de classificação;
9. gerar confiança consolidada com base nas classificações.

A classificação atual utiliza uma combinação de abordagens, incluindo:

- regressão logística;
- vetorização com TF-IDF;
- análise de remetente;

- análise de cliente;
  - análise de tipos de anexos;
  - chamada para LLM;
  - cálculo de confiança ponderada.
- 

## 2.3 Uso de LLM e regressão logística

Foi discutido que a regressão logística depende de um dataset treinado especificamente para a tarefa desejada.

O dataset atual classifica se um e-mail é ou não uma solicitação. Para classificar se uma solicitação Petronect é pública ou convite/dispensa, seria necessário outro dataset, pois se trata de uma classificação diferente.

Também foi discutida a possibilidade de criar uma chamada para LLM com prompt específico para essa nova classificação.

No entanto, para o caso específico da Petronect, foi entendido que talvez não seja necessário iniciar com modelo de IA, já que os e-mails possuem padrões textuais que podem ser tratados por regra simples.

---

# 3. Petronect — classificação de oportunidades públicas e convites

## 3.1 Problema identificado

Inicialmente, havia a hipótese de usar a API da Petronect para diferenciar oportunidades públicas de convites ou dispensas.

A lógica proposta era:

- se a API retornasse dados, a oportunidade seria pública;
- se a API retornasse vazio, nulo ou 404, a oportunidade seria tratada como convite ou dispensa.

Durante a reunião, foi identificado que essa lógica não era suficientemente segura, pois houve casos em que uma oportunidade aparentemente pública não retornou corretamente pela API.

Com isso, concluiu-se que a API **não deve ser a única fonte de decisão** para classificar a oportunidade.

---

## 3.2 Decisão sobre o filtro Petronect

Foi decidido que a classificação entre **pública** e **convite/dispensa** deve considerar o **corpo do e-mail**.

A regra inicial definida foi:

- se o corpo do e-mail contiver termos como **“licitação pública”** ou **“oportunidade pública”**, classificar como:
  - Petronect - Oportunidade Pública
- se o corpo do e-mail contiver termos como **“sua empresa foi convidada”**, classificar como:
  - Petronect - Convite
  - ou Petronect - Dispensa, conforme nomenclatura final a ser adotada.

Foi comentado que essa verificação pode ser implementada com uma lógica simples, como busca textual, regex ou consulta no banco, desde que o corpo do e-mail esteja salvo.

---

## 3.3 Uso da API da Petronect

A API continuará sendo considerada útil para buscar informações adicionais da oportunidade pública.

Entretanto, ela não será usada isoladamente para definir se a oportunidade é pública ou convite.

Fluxo acordado:

1. identificar se o e-mail é uma solicitação;
  2. verificar se é Petronect;
  3. analisar o corpo do e-mail;
  4. classificar como oportunidade pública, convite ou dispensa;
  5. quando for pública e houver retorno da API, buscar dados complementares;
  6. quando não houver retorno da API, não assumir automaticamente que não é pública;
  7. salvar a classificação para consulta posterior.
- 

## 4. MVP e limites do escopo

Foi reforçado que o projeto deve seguir com uma abordagem de **MVP**, priorizando a entrega funcional antes de buscar refinamentos excessivos.

A decisão foi trabalhar com as informações atualmente disponíveis, mesmo que a base da empresa ainda não esteja totalmente estruturada.

Foram citados alguns pontos que podem gerar limitações:

- BU genérica demais;
- ausência de documentação detalhada de famílias Petronect;
- falta de mapeamento completo de quais famílias a empresa atende;
- ausência de portfólio técnico detalhado;
- requisitos de qualidade por família ainda não tratados;
- possibilidade de classificações incorretas por falta de informação de negócio.

A orientação foi: primeiro entregar uma solução funcional e, quando os gargalos aparecerem em produção, tratar as melhorias com base em evidências.

---

## 5. Famílias Petronect

Foi discutido que a Petronect trabalha com famílias de fornecimento e que a empresa pode estar cadastrada em diferentes famílias.

Foi levantada a possibilidade de, futuramente, existir uma rotina para verificar ou atualizar automaticamente o cadastro da empresa nas famílias da Petronect.

No entanto, foi definido que esse tema **não entrará como prioridade do MVP**, pois a empresa ainda não possui documentação interna suficiente indicando:

- quais famílias atende;
- quais requisitos técnicos cumpre;
- quais famílias não fazem sentido para o negócio;
- quais produtos ou serviços estão associados a cada família.

A conclusão foi que esse tipo de refinamento depende de informações do negócio e poderá ser tratado futuramente.

---

## 6. BU e qualidade das classificações

Foi identificado que a BU atual pode estar genérica demais.

Exemplo discutido: uma BU que menciona “projetos elétricos” pode levar o modelo a concluir que a empresa atende uma solicitação de painel elétrico, mesmo que esse item não faça parte do escopo real da empresa.

Foi entendido que esse problema não é exclusivamente do software, mas também da qualidade da informação fornecida ao modelo.

A melhoria da BU foi considerada importante, mas não impeditiva para o MVP.

---

## 7. Requisitos de qualidade e SGQ

Foi discutido que algumas solicitações podem exigir requisitos específicos de qualidade, associados ao SGQ ou a categorias técnicas.

Foi citado que a empresa pode atender determinado item, mas não atender um requisito de qualidade específico exigido naquela oportunidade.

Apesar disso, ficou decidido que, no MVP, o sistema não fará essa validação detalhada.

A classificação inicial será baseada principalmente em aderência ao escopo da empresa.

Validações mais específicas, como requisitos de qualidade, famílias, tipos de fornecimento ou restrições técnicas, poderão ser adicionadas em fases futuras.

---

## 8. Interface, dashboard e acompanhamento

Foi discutida a existência de uma interface para visualização das classificações e acompanhamento dos e-mails.

Foi levantada a comparação com Metabase, mas foi apontado que a necessidade principal não é apenas visualizar gráficos, e sim permitir que o usuário veja os e-mails classificados, filtre por status e acompanhe o processo de triagem.

Também foi mencionado que ferramentas como Plums ou sistemas comerciais podem receber apenas aquilo que já está dentro do escopo, enquanto o sistema de IA precisa lidar também com:

- fora do escopo;
- não solicitação;
- solicitação em análise;
- solicitação Petronect;
- oportunidades públicas;
- convites;
- respostas e e-mails internos.

---

## 9. Decisões tomadas

1. A classificação Petronect não dependerá exclusivamente da API.
2. O corpo do e-mail será usado para identificar se a solicitação é pública ou convite.
3. Termos como “licitação pública” e “oportunidade pública” indicarão oportunidade pública.
4. Termos como “sua empresa foi convidada” indicarão convite ou dispensa.
5. A API será usada para buscar dados complementares, mas não como única regra de classificação.
6. O MVP não tratará, neste primeiro momento, regras avançadas de famílias Petronect.
7. O MVP não tratará, neste primeiro momento, requisitos específicos de qualidade/SGQ.
8. O sistema será desenvolvido com base na realidade atual das informações disponíveis.
9. Melhorias de BU, famílias e requisitos técnicos serão tratadas futuramente, conforme surgirem gargalos.
10. A classificação deverá ser salva para consulta posterior em banco, dashboard ou interface de acompanhamento.

---

## 10. Encaminhamentos

Item	Responsável	Encaminhamento
1	Gustavo	Implementar filtro para identificar e-mails Petronect.
2	Gustavo	Criar regra para classificar oportunidade pública com base no corpo do e-mail.

Item	Responsável	Encaminhamento
3	Gustavo	Criar regra para classificar convite/dispensa com base no corpo do e-mail.
4	Gustavo	Avaliar se a regra será feita por regex, busca textual ou lógica no banco.
5	Lucas	Apoiar na definição do fluxo geral e validação da lógica.
6	Lucas/Gustavo	Definir nomenclatura final para as categorias Petronect.
7	Lucas/Gustavo	Manter a API como fonte complementar para busca de informações da oportunidade.
8	Lucas/Gustavo	Deixar melhorias de famílias, BU e SGQ para etapas futuras.

# 11. Pendências

- Definir a nomenclatura final entre:
  - Petronect - Convite ;
  - Petronect - Dispensa ;
  - Petronect - Oportunidade Fechada .
- Validar mais exemplos reais de e-mails Petronect para confirmar os padrões textuais.
- Verificar se o corpo do e-mail está sendo salvo no banco para permitir consultas futuras.
- Definir onde a classificação será exibida:
  - dashboard;
  - caixa de entrada da interface;
  - banco de dados;
  - Metabase;
  - outra ferramenta de acompanhamento.
- Avaliar futuramente a necessidade de:
  - scraping para convites;
  - atualização automática de famílias;
  - documentação técnica por família;
  - validação de requisitos de qualidade;
  - refinamento da BU.

# 12. Conclusão

A reunião consolidou que o projeto deve seguir com foco em entrega de MVP, evitando aprofundar neste momento em regras complexas de negócio que dependem de documentação ainda inexistente ou incompleta.

Para as solicitações da Petronect, ficou definido que a classificação entre oportunidade pública e convite/dispensa deve ser feita inicialmente por análise do corpo do e-mail, usando padrões textuais identificáveis. A API será mantida como apoio para buscar informações adicionais, mas não como única fonte de decisão.

O objetivo imediato é colocar o fluxo em produção com regras simples, validar o comportamento com dados reais e evoluir o sistema conforme os gargalos aparecerem.

# Levantamento de requisitos

## - Modelo Atual

### 1. Visão Geral e Escopo

**Objetivo:** Automatizar a triagem de e-mails comerciais das empresas **Autvix** e **Advix Soluções**, utilizando LLM para classificar demandas contra uma base de serviços (BUS).

**Principais Metas:**

- Triagem automática de solicitações via e-mail e anexos.
- Redução do tempo operacional e atrasos na resposta ao cliente.
- Persistência de dados para auditoria e análise via Metabase.

### 2. Requisitos Funcionais (RF)

ID	Descrição	Referência Técnica
RF01	Capturar e-mails e anexos das caixas comerciais e salvar no OneDrive.	Power Automate
RF02	Identificar o remetente e validar se o domínio já existe na base de contatos.	Tabela <code>contacts</code>
RF03	Converter anexos (.pdf, .xlsx, .zip) para texto para processamento pelo LLM.	Scripts Python
RF04	Classificar e-mails em categorias pré-definidas com nível de confiança e justificativa.	LLM gpt-oss:120b
RF05	Monitorar a saúde dos serviços de ingestão e processamento.	Tabela <code>service_heartbeat</code>

<b>RF06</b>	Armazenar o corpo do e-mail em HTML e texto plano para rastreabilidade.	Tabela <code>emails</code>
-------------	---	----------------------------

## 3. Requisitos Não Funcionais (RNF)

- **RNF01 - Integridade Referencial:** O sistema deve garantir que nenhum e-mail seja órfão de um contato ou de uma classificação através de chaves estrangeiras (FK).
- **RNF02 - Desempenho de Busca:** Utilização de índices (KEY) em campos críticos como `tipo`, `recipient`, `domain` e `received_at` para otimizar os dashboards.
- **RNF03 - Padronização de Caracteres:** Todo o banco deve utilizar o charset `utf8mb4_unicode_ci` para suportar caracteres especiais e emojis de e-mails.
- **RNF04 - Disponibilidade:** O status do sistema deve ser reportado via *heartbeat* em intervalos regulares.

## 4. Modelo de Dados (Arquitetura do Banco)

O banco de dados MySQL é o coração da persistência do EscopoVix, estruturado da seguinte forma:

### A. Núcleo de Comunicação

- **contacts:** Gerencia os remetentes. Classifica o `tipo` entre `CLIENTE`, `PORTAL` ou `DESCONHECIDO` e marca se o contato já está cadastrado no sistema legiado (`is_cadastrado`).
- **emails:** Armazena os metadados da mensagem, o caminho da pasta no OneDrive (`folder_path`), a contagem de anexos e o conteúdo textual.

### B. Inteligência e Classificação

- **classifications:** Armazena o veredito do modelo `gpt-oss:120b`.
  - **Tipos permitidos:** `DENTRO_ESCOPO`, `FORA_ESCOPO`, `ANALISE_TECNICA`, `NAO_SOLICITACAO`.
  - **Campos de IA:** `confidence` (0 a 1) e `reasoning` (explicação lógica da IA para aquela decisão).

### C. Apoio e Monitoramento

- **cliente:** Tabela de referência para cruzamento de dados de domínio, CNPJ e Razão Social.

- **service\_heartbeat**: Registra se o serviço `watcher` ou `processor` está `running`, `stopped` ou em `error`, incluindo metadados em JSON para diagnóstico.
- 

## 5. Fluxo de Dados e Pipeline

1. **Ingestão**: Power Automate salva e-mail no OneDrive.
2. **Detecção**: O `watcher` detecta a nova pasta.
3. **Identificação**: O sistema busca ou cria o registro em `contacts` via domínio do e-mail.
4. **Processamento**: O LLM analisa o texto e gera um registro em `classifications` com o `reasoning`.
5. **Finalização**: O registro em `emails` é atualizado com o `classificacao_id` e marcado como processado.