💡 Explicação do Funcionamento do Sistema

O sistema é um **Pipeline Assíncrono Orientado a Eventos** construído sobre o *framework* Django. Ele transforma dados não estruturados (o corpo de um e-mail) em dados estruturados (um JSON validado) e usa essa informação para automatizar ações em serviços externos (Trello, Telegram).

O motor que mantém a automação é o **Django-Q**, que atua como um gerenciador de filas para processar tarefas em *background* sem bloquear a aplicação web principal.

O Papel de Cada Pilar

Pilar	Responsável	Função no Fluxo
Backend Core & APIs	**Jullio**	Fornece a base de dados (MailBox, EmailMessage), a interface de gerenciamento (APIs REST) e a segurança (settings.py *hardened*).
Worker Assíncrono	**Thales**	Orquestra a pipeline: Busca e-mails, chama a IA de Juliano e executa as integrações externas, registrando o processo.
Inteligência (IA)	**Juliano**	Recebe o texto bruto do e-mail, usa o **ChatGPT** para extrair metadados e valida essa saída no formato **JSON (Pydantic)**.

Exportar para as Planilhas

🔄 Fluxo Completo: Do E-mail ao Card no Trello

O processo é dividido em duas fases principais: a **Busca (Fetch)** e o **Processamento Assíncrono (Pipeline)**.

Fase 1: Busca e Enfileiramento (Trigger)

Passo	Ação	Responsável	Componentes
1. Agendamento	O **Django-Q Scheduler** (Beat) é configurado para rodar periodicamente (e.g., a cada 5 minutos).	Jullio	settings.py
2. Conexão IMAP	O *worker* **tasks.fetch_emails** conecta-se à MailBox (usando as credenciais seguras do DB).	Thales	tasks/tasks.py
3. Captura	O *worker* busca e-mails novos ou não processados. Para cada e-mail, ele extrai o corpo do texto, assunto, remetente e o **message_id** (para **idempotência**).	Thales	imapclient

Passo	Ação	Responsável	Componentes
4. Persistência	O *worker* salva o e-mail no DB de Jullio como um novo **EmailMessage** com o status **PENDING**.	Thales	emails/models.py
5. Enfileiramento	O *worker* dispara a próxima tarefa para a fila do Django-Q: **async_task('tasks.process_email', email_id)**.	Thales	django_q

Exportar para as Planilhas

Fase 2: Processamento e Integração (Pipeline Assíncrono)

Passo	Ação	Responsável	Componentes
6. Extração IA	O *worker* **tasks.process_email** chama a função **extract_fields_from_text** de Juliano, passando o corpo do e-mail.	Thales	tasks/tasks.py
7. Inteligência e Validação	**Juliano** envia o texto para o OpenAI, instruindo-o a retornar um **JSON** que adere estritamente ao **Schema Pydantic** (ex: ServiceOrderSchema). Se o JSON for inválido, o sistema tenta o *re-prompt* (**fallback**).	Juliano	ai_wrapper.py, schemas.py
8. Persistência de Dados	Se a extração for bem-sucedida, o *worker* salva o **JSON validado** no campo **extracted_data** do EmailMessage no DB de Jullio.	Thales	emails/models.py
9. Integração Trello	O *worker* chama a função **create_trello_card**, mapeando os campos do JSON (extracted_data) para o título e a descrição do card.	Thales	integrations/trello.py
10. Notificação	O *worker* chama **notify_telegram** para alertar o time sobre o novo processo iniciado.	Thales	integrations/telegram.py
11. Log e Conclusão	A cada chamada de API externa (Trello, Telegram), o **IntegrationLog** é preenchido com o status e a resposta. O status final do **EmailMessage** é atualizado para	Thales	integrations/models.py

Passo	Ação	Responsável	Componentes
	INTEGRATED ou **FAILED**.		

Exportar para as Planilhas

Ponto de Controle de Segurança

- Em qualquer ponto, administradores podem consultar o estado do **EmailMessage** e do **IntegrationLog** através da API de **Jullio** ou pelo Admin do Django.
- Em caso de falha de extração, o status é **REQUIRES_REVIEW**, e a equipe é notificada para intervenção humana.