



Code Island
CLOUD
2025

IA Generativa em Produção

PATROCINADORES:



DRAGONBD



NEOBITS
SOLUÇÕES EM TECNOLOGIA



KuberLink
CLOUD SOLUTIONS



API4COM

dati →



IA Generativa na Realidade

1. Ponto central

- IA não é fim, é meio.
Só gera valor quando sustentada por processos claros, dados estruturados e arquitetura robusta. Sem isso, o que se automatiza é o caos.

2. Realidade prática da IA

- IA não é mágica → segue as mesmas regras de qualquer software.
- Depende de:
 - Processos bem definidos.
 - Dados confiáveis e organizados.
 - Engenharia e arquitetura robustas.
- A entrega de valor acontece quando tecnologia, dados e negócio estão integrados.

3. Reflexão final

- IA não substitui pessoas, potencializa resultados. Ela organiza, orchestra e acelera. Mas só funciona bem quando os processos estão claros e os dados são confiáveis. Caso contrário, o que se automatiza é o caos.



Camada de Aplicação e Infraestrutura

- Arquitetura Python para IA generativa.
- Componentes principais:
 - FastAPI (API e interface com clientes).
 - Celery Workers/Beat + RabbitMQ (tarefas assíncronas).
 - LangChain + LangGraph (IA multiagente e orquestração).
 - MongoDB vetorial
- Objetivo: resiliência, modularidade e escalabilidade.

- Microserviços no GKE
 - Deploy no Google Kubernetes Engine.
 - Node Pools em multi-regions → tolerância a falhas.
 - Namespaces para isolar soluções (SAC, SAF, Prospect, etc.).
 - Observabilidade com Cloud Logging e Monitoring.

FastAPI – API para IA Generativa

- Framework moderno, leve e performático em Python.
- Construído para alta performance → baseado no Starlette (web) e Pydantic (validação de dados).
- Suporte nativo a async/await, ideal para workloads de IA que dependem de chamadas externas (modelos, vetores, bancos).
- Documentação automática → gera Swagger/OpenAPI sem esforço.
- Base sólida para expor APIs de IA generativa e integrações com clientes, microsserviços e frontends.

Boas práticas de uso em produção:

- Rotas organizadas por domínio (ex.: [/auth](#), [/chat](#), [/admin](#)).
- Autenticação segura → JWT, OAuth2 ou API Keys.
- Uso de variáveis de ambiente (12-Factor App).
- Middlewares para logging, tracing e métricas.
- Versionamento de rotas ([/api/v1](#), [/api/v2](#)) para evolução controlada.





RabbitMQ – Orquestrador de Mensagens

- Broker de mensagens confiável para comunicação entre serviços.
- Suporte a filas, tópicos e roteamento, permitindo diferentes padrões de entrega.
- Projetado para resiliência em cenários de alta carga (ex.: bursts de chamadas de IA).
- Garante desacoplamento → a aplicação continua funcionando mesmo quando um modelo ou serviço demora a responder.
- Amplo ecossistema de plugins e integrações (monitoramento, autenticação, dashboards).

Boas práticas de uso em produção:

- Criar dead-letter queues (DLQ) para mensagens com falha.
- Definir políticas de retry/backoff em caso de erros.
- Monitorar consumo com Prometheus/Grafana.
- Configurar persistência de mensagens para workloads críticos.
- Segregar filas por tipo de workload ou prioridade.



Celery (Workers & Beat)

- Workers → processam tarefas de IA de forma assíncrona.
- Beat → agenda jobs recorrentes (ex.: envio de trilha de follow-up, limpeza de logs, rotinas de manutenção).
- Suporte a escalabilidade horizontal → adicionar mais workers = aumentar capacidade de processamento.
- Integração natural com RabbitMQ.
- Confiável e amplamente utilizado em produção para sistemas distribuídos.

Boas práticas em produção:

- Definir retries automáticos com backoff para tarefas que falham.
- Usar timeouts para evitar tasks presas indefinidamente.
- Dividir filas de tasks por prioridade (ex.: chamadas críticas vs. batch).
- Garantir idempotência das tarefas (não causar efeitos duplicados em caso de reprocessamento).



LangChain

- Framework projetado para acelerar o desenvolvimento de soluções de IA.
- Permite construir aplicações modulares e reutilizáveis.

Abstrai complexidades como:

- Chamadas a modelos (LLMs) de diferentes provedores.
- RAG (Retrieval Augmented Generation) → integração com bancos vetoriais e bases de conhecimento.
- Ferramentas externas → APIs, bancos de dados, sistemas corporativos.
- Evita “reinventar a roda” e libera o time para focar na lógica de negócio e orquestração dos agentes.

Boas práticas em produção:

- Encapsular LLMs em chains bem definidos.
- Usar retrievers vetoriais para enriquecer prompts com contexto.
- Monitorar custo e latência das chamadas a modelos.
- Implementar fallback automático para provedores alternativos.
- Controlar versões de chains/agents para manter consistência.



LangGraph - Orquestração Multiagente

- Evolução sobre o LangChain: foca em fluxos complexos de execução.
- Representa pipelines de IA como grafos de execução em vez de sequências lineares.

Permite:

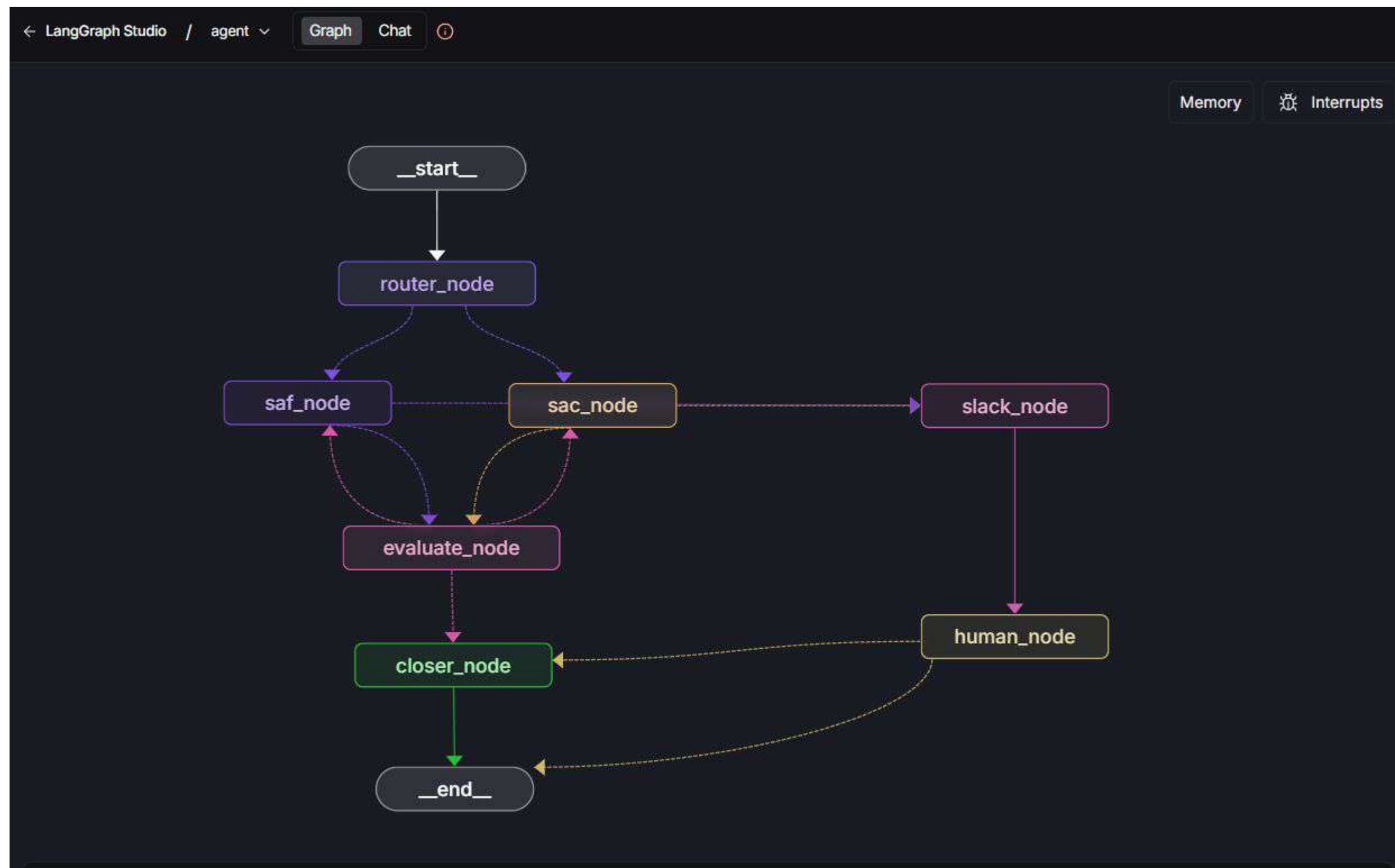
- Controle de estado → memória persistente entre passos.
- Loops → voltar e refinar respostas automaticamente.
- Fallback → redirecionar para outro modelo/agente se algo falhar.
- Ideal para arquiteturas de IA multiagente.

Uso prático em produção:

- Agente roteador decide qual agente especialista chamar:
 - Análise de imagem.
 - Python/SQL agent para cálculos e queries.
- Garante escalabilidade (cada agente pode evoluir de forma independente).
- Favorece modularidade → trocar um agente não quebra todo o fluxo.



LangGraph





LangGraph

ia_sacsaf_alertas

Messages Files Sem título +

Today

Transferir

10:16 Olá! Será necessária uma intervenção humana, pois a informação solicitada não está disponível na minha base de conhecimento.

Ver Chamado

Informações do chamado sac

ID: 264238

Categoria: Sou Cliente market4u

Franqueado/Usuário: Jorge Luis Delbin

Pergunta reformulada: Como posso acompanhar o status do meu pedido #37792165 e obter mais informações sobre o desconto aplicado?

Resumo: O usuário enviou uma imagem de um resumo de compras de um aplicativo de celular. A imagem mostra detalhes como o número do pedido, produtos comprados, preços, descontos e formas de pagamento. A conversa indica que o caso do usuário foi encaminhado para a equipe responsável, que dará retorno sobre a solicitação.

Você pode definir uma data de expiração para a resposta ao adicioná-la à base de conhecimento. Caso não deseje expirar, deixe esse campo em branco!

Resposta ao usuário

Write something

Selecione a data Seleccione o horário **Adicionar Resposta**

Transferir

B I | | | | | | | |

Message B ia_sacsaf_alertas

+ Aa | | | | |





MongoDB

- Armazenamento vetorial para RAG (Retrieval Augmented Generation) → perguntas/respostas com contexto.
- Suporte a embeddings de múltiplos modelos (OpenAI, Gemini, Llama etc.).
- Garante continuidade e resiliência da aplicação de IA em produção.
- Flexibilidade para lidar com diferentes tipos de dados (texto, imagens, metadados).

Boas práticas em produção:

- Indexar embeddings em coleções dedicadas.
- Usar filtros de metadados para refinar buscas (ex.: por cliente, domínio, data).
- Definir TTLs ou rotinas de limpeza para embeddings desatualizados.
- Monitorar latência de consultas → RAG precisa ser rápido para não travar experiência.



MCP

- Hub de ferramentas para agentes de IA.
- Centraliza acesso a APIs, bancos de dados e integrações externas.
- Evita dependência hardcoded → agentes não precisam conhecer diretamente cada serviço.
- Garante consistência e governança no uso de recursos compartilhados.

Boas práticas em produção:

- Expor ferramentas via protocolos padronizados → fácil plugar/remover integrações.
- Definir permissões claras por agente (evitar acesso desnecessário).
- Monitorar uso das ferramentas → métricas de consumo, latência, falhas.
- Versionar integrações críticas (API externa, DB, serviços internos).

Por que MCP importa?

- Padronização: um protocolo único para todos os agentes.
- Agilidade: novas ferramentas podem ser plugadas sem alterar o core dos agentes.
- Governança: controle centralizado de permissões e acessos.
- Escalabilidade real: agentes diferentes usam as mesmas ferramentas de forma segura.



Avaliação Contínua dos Modelos

- Riscos a monitorar:
 - Alucinação → respostas inventadas ou incoerentes.
 - Conteúdo sensível → hate speech, viés, linguagem inadequada.
 - Consistência → manter qualidade mesmo com aumento de uso.
- Estratégias de mitigação:
 - Testes automáticos com datasets de avaliação.
 - Métricas específicas: precisão, relevância, taxa de fallback.
 - Monitoramento em tempo real + feedback humano.
 - Camada de segurança → filtros, moderadores de conteúdo e regras de negócio.
- Reflexão:
 - IA em produção é ciclo contínuo de aprendizado + correção.
 - Não é estática, evolui junto com dados, usuários e contexto.

CI/CD e Segurança


DRAGONBD


NEOBITS
SOLUÇÕES EM TECNOLOGIA


KuberLink
CLOUD SOLUTIONS


API4COM


dati

- GitHub Actions + Workload Identity Federation → deploy sem chaves locais.
- Pipelines automatizados com testes + lint + validação.
- Segredos no Secret Manager.
- Princípio do mínimo acesso em IAM.

Métricas de Desempenho

Dashboard de Cobrança Produção De: Até: Filtrar 7 dias 30 dias 90 dias

Total de Tickets 314 No período	Tickets Pagos 227 72.3% pagos/total de tickets	Conversão de Pago 75.7% pagos/total de tickets finalizados	Valor Recuperado R\$ 5.027,16 Total Comunicado: R\$ 7.890,99
Esforço Humano Equivalente (Pagos) 15 h Média: 4 msgs IA/ticket Período conversa: 6306.4 h	Esforço Humano Equivalente Total 24.8 h Média: 4.7 msgs IA/ticket Período conversa: 14181.3 h	Mensagens da IA 1.490 No período selecionado	
Conversas Ativas 14 Em andamento	Taxa de Resposta 48.7% Clientes que responderam	Transferidos 72 Para humano	Intervenções 0 Via Controlhub



Métricas de Desempenho



market4u

AIControlHUB

Home

Dashboard

Prompts IA

Chat IA Homolog

Templates WhatsApp

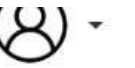
Gerenciamento

Maia RH

Maia Cobrança

Maia Comercial Franquias

Maia Comercial Cardápio



Dashboard SAC/SAF

Todos (SAC/SAF)

Banco de Produção

Últimos 30 dias

Total de Tickets

7.435

No período

Finalizado pela IA

3.030

Status: F

Transferido p/ Analista

3.514

Status: TA

Transferido pelo Analista

701

Status: TAF

Respondido pela IA

183

Status: IAR

Aguardando IA

7

Status: IAN

Taxa de Finalização IA

42%

3.030 de 7.245 tratativas

Transferidos para Analista

49%

3.514 de 7.245 tratativas

Transferidos pelo Analista

10%

701 de 7.245 tratativas



Conclusão

- IA generativa em produção = engenharia de software de ponta.
- Não é mágica: requer
 - Aplicação robusta (FastAPI, Celery, RabbitMQ).
 - Orquestração de agentes (LangChain/LangGraph).
 - Dados confiáveis (MongoDB vetorial).
 - Infra resiliente (GKE, multi-region, CI/CD).
- Resultado: IA escalável, governada e confiável.

Muito Obrigado

Linkedin: <https://www.linkedin.com/in/nicolasjoseit>

Kuberlink: <https://kuberlink.com.br>



PATROCINADORES:

