BIUD Talks

Integração de Sistemas

Gabriel Regis - Desenvolvedor Full Stack

Cronograma do Programa

22/04	**Wanderson**	Inteligência Artificial para Dev
29/04	**Praxedes**	Testes Unitários
06/05	**Luk**	Arquitetura de Software
13/05	**Regis**	Integração de Sistemas
20/05	**Rogério**	Root Cause Analysis (RCA)
⚡ 27/05	**Gustavo**	Metodologia Ágil
⚡ 03/06	**Gabriel Pereira**	Banco de Dados

Estrutura de cada Sessão

Apresentação

20-30 minutos

Conteúdo e expertise do palestrante

Debate & Discussão

30 minutos

Dúvidas, insights e networking

? Moderador: Rogério

Responsável por mediar o debate e garantir engajamento de todos os participantes

Integração de Sistemas

- **Documentação do ERP do cliente**

  Mapeamento de endpoints REST ou schema do banco (ex: TOTVS Protheus, Bling, Linx Microvix, Shopify, etc.)

  Ex: Mapeamento de tabelas e endpoints para a montar o json, como tabelas de vendas, clientes e produtos.

- **Estratégias de integração: API REST vs. banco direto**

  Critérios: disponibilidade de API, volume de dados, latência, indisponibilidade, inconsistência e permissões do cliente.

  Ex: Cliente com Bling → usamos API REST com paginação; cliente com Protheus sem API habilitada → conexão direta via SQL Server com usuário read-only.

- **Coleta e normalização dos dados de venda**

  Padronização de campos (produto, SKU, valor, data) em schema consolidado para modelo de padronização dos dados requeridos.

  Ex: Bling retorna "valor_bruto" e o SAP retorna "NetAmount" — ambos mapeados para o campo sale_amount no nosso modelo unificado.

Integração de Sistemas

- **Atualização em tempo real e sincronização contínua**

Jobs agendados via cron ou webhooks do ERP disparam a coleta incremental buscando apenas registros alterados desde a última execução (delta sync). Estratégias como timestamp de updated_at, e filas (RabbitMQ) garantem que a base reflita o estado atual com latência controlada, sem reprocessar todo o histórico.

Ex: cron diária busca pedidos com updated_at > last_sync; em clientes com alto volume, usamos CDC no SQL Server com captura de log de transações para latência abaixo de 5 minutos.

- **Consistência e integridade dos dados na nossa base**

Uso de idempotência, deduplicação e logs de sincronização para evitar dados duplicados.

Ex: usamos o order_id do ERP como chave de upsert — se o pedido já existe, atualizamos, se não, inserimos. Log registra cada execução com status e total de registros afetados.

- **Segurança e conformidade com a LGPD**

Anonimato de CPF/e-mail, criptografia em trânsito (TLS) e controle de acesso por escopo.

Ex: CPF armazenado como rasura (***) para fins de anonimato; e-mail mascarado em logs (j*@gmail.com); credenciais do ERP guardadas no Vault (ou rancher), nunca em .env commitado.**

De extrema a importância um script pronto para detonação dos dados sensíveis coletados de base externa a fim de conformidade com a Lei n° 13.709/2018 (Lei Geral de Proteção de Dados Pessoais).

Abrindo o Debate

❓ Questões para reflexão:

1. Qual foi o principal aprendizado para você?
2. Como você pode aplicar isso em seu dia a dia?
3. Quais foram as dúvidas geradas?
4. Que conexões você vê com outros tópicos?

Obrigado!

Até o próximo episódio da BIUD Talks

 Inovando juntos | Crescendo juntos