

# Análise e Fluxo Atualizado de Vinculação entre Agendamentos, Fichas, Sessões e Execuções

---

## 1. Visão Geral

Este documento descreve o fluxo atualizado de dados e os mecanismos implementados para vincular os diferentes registros relacionados aos atendimentos:

- **Agendamentos:** Compromissos registrados no sistema ABA.
- **Fichas:** Fichas de presença físicas digitalizadas (PDFs).
- **Sessões:** Registros individuais de atendimento extraídos das fichas.
- **Execuções:** Registros de atendimento capturados do sistema da Unimed via scraping.

O objetivo principal é estabelecer vínculos claros e confiáveis entre esses elementos para garantir a consistência dos dados, facilitar auditorias e o processo de faturamento.

## 2. Estrutura das Tabelas Principais

### AGENDAMENTOS

- Armazena dados dos agendamentos importados do sistema ABA.
- Campos chave: `id`, `paciente_id`, `data_agendamento`, `procedimento_id`.

### FICHAS

- Armazena informações das fichas de presença digitalizadas.
- Campos chave: `id`, `storage_id`, `codigo_ficha`, `numero_guia`, `data_atendimento`, `total_sesoes`.
- **Campos de Vinculação Adicionados:**
  - `paciente_id` (FK para PACIENTES): Preenchido durante o processamento do PDF para facilitar joins.

### SESSOES

- Representa cada linha (sessão) extraída da ficha de presença.
- Campos chave: `id`, `ficha_id`, `data_sessao`, `possui_assinatura`, `ordem_execucao`.
- **Campos de Vinculação Adicionados:**
  - `agendamento_id` (FK para AGENDAMENTOS): Vincula a sessão ao agendamento correspondente.

### EXECUCOES

- Registros de execuções capturadas no sistema da Unimed.
- Campos chave: `id`, `numero_guia`, `data_execucao`, `ordem_execucao`, `origem`.
- **Campos de Vinculação Adicionados:**
  - `sessao_id` (FK para SESSOES): Vincula a execução à sessão correspondente na ficha.

- `agendamento_id` (FK para AGENDAMENTOS): Vincula a execução ao agendamento correspondente.
- `paciente_id` (FK para PACIENTES): Vincula a execução ao paciente (obtido via guia).
- `link_manual_necessario` (BOOLEAN): Sinaliza execuções que não puderam ser vinculadas automaticamente à sessão com alta confiança.
- `codigo_ficha_temp` (BOOLEAN): Indica se o `codigo_ficha` ainda é temporário (não vinculado à ficha real).

## UNIMED\_SESSOES\_CAPTURADAS

- Tabela intermediária para dados brutos do scraping da Unimed.
- Campos chave: `id`, `numero_guia`, `data_atendimento_completa`, `data_execucao`, `ordem_execucao`, `task_id`.

## UNIMED\_LOG\_PROCESSAMENTO

- Log detalhado do processamento de cada sessão da Unimed, incluindo o resultado da tentativa de vinculação com `sessoes`.

# 3. Fluxo de Dados e Vinculação

## 3.1. Entrada de Dados

1. **Agendamentos:** Importados periodicamente do sistema ABA para a tabela `agendamentos`.

2. **Fichas de Presença (PDF):**

- Upload via interface do sistema.
- Processamento por IA (Claude/Gemini/Mistral) para extrair dados, incluindo `codigo_ficha`, `numero_guia`, `data_atendimento`, `paciente_carteirinha`, e para cada linha: `ordem_execucao`, `data_sessao`, `possui_assinatura`.
- Criação do registro na tabela `fichas` (incluindo busca do `paciente_id` associado à carteirinha).
- Criação de múltiplos registros na tabela `sessoes`, um para cada linha extraída, populando `ficha_id`, `data_sessao`, `possui_assinatura` e `ordem_execucao` (extraída pela IA).

3. **Execuções da Unimed (Scraping):**

- Script `todas_as_fases_adaptado.py` executa o scraping.
- Captura detalhes das execuções (guia, data, **ordem de execução**, dados do profissional) do site da Unimed.
- Dados brutos são salvos na tabela intermediária `unimed_sessoes_capturadas`, incluindo `ordem_execucao`.
- Função SQL `inserir_execucao_unimed` é chamada para cada registro em `unimed_sessoes_capturadas`.

## 3.2. Vinculação Sessão <-> Execução (Função `inserir_execucao_unimed`)

Esta é a principal etapa de vinculação automática, ocorrendo quando uma execução da Unimed é processada:

1. **Busca Guia Principal e Paciente:** Localiza a `guia_id` e o `paciente_id` associado na tabela `guias` usando o `numero_guia`. Se não encontrar, a execução é marcada com erro.
2. **Tentativa de Vinculação com Sessão:**
  - **Nível 1 (Exato):** Busca uma `sessoes` com o mesmo `numero_guia`, mesma `data_execucao` (convertida para `date`) e mesma `ordem_execucao`. Se encontrar uma única correspondência, o `sessoes.id` é usado.
  - **Nível 2 (Tolerância + Ordem):** Se Nível 1 falhar, busca `sessoes` com mesmo `numero_guia`, `data_sessao` +/- 1 dia da `data_execucao`, e mesma `ordem_execucao`. Se encontrar uma única correspondência na janela, o `sessoes.id` é usado.
  - **Nível 3 (Tolerância s/ Ordem - Única):** Se Nível 2 falhar, busca `sessoes` com mesmo `numero_guia` e `data_sessao` +/- 1 dia. **Somente vincula se encontrar exatamente uma `sessoes` candidata** nesta janela.
3. **Criação da Execução:**
  - Insere o registro na tabela `execucoes`, incluindo o `paciente_id` encontrado no passo 1.
  - O campo `sessao_id` é preenchido se a vinculação (Nível 1, 2 ou 3) foi bem-sucedida.
  - O campo `agendamento_id` pode ser preenchido aqui se encontrado via `sessoes` ou buscando diretamente por `paciente_id/data`.
  - O campo `codigo_ficha` é preenchido com o `codigo_ficha` real da `sessoes` vinculada; caso contrário, usa o código temporário (`TEMP_...`). `codigo_ficha_temp` é definido como `FALSE` se vinculado, `TRUE` caso contrário.
  - O campo `link_manual_necessario` é definido como `TRUE` se a vinculação automática falhou ou foi considerada ambígua (ex: Nível 3 encontrou múltiplas candidatas).
4. **Log:** Registra o resultado da tentativa de vinculação em `unimed_log_processamento`.

### 3.3. Vinculação com Agendamentos (Função Batch `vincular_agendamentos`)

Esta vinculação ocorre separadamente, através de uma função batch (`vincular_agendamentos`) que **não é executada automaticamente na criação de cada registro**, mas sim chamada sob demanda (via API/botão) ou periodicamente (ex: job agendado). Ela visa conectar `sessoes` e `execucoes` aos `agendamentos` correspondentes:

1. **Vincular Sessões a Agendamentos:**
  - Busca `sessoes` onde `agendamento_id` é nulo.
  - Para cada `sessao`, procura por um `agendamentos` correspondente usando:
    - `paciente_id` (obtido da tabela `fichas` associada à sessão).
    - `data_sessao` comparada com `data_agendamento`.
    - `procedimento_id` (obtido da tabela `guias` associada à sessão).
  - Se **exatamente um** `agendamentos` for encontrado com essa combinação, atualiza `sessoes.agendamento_id`.
2. **Vincular Execuções a Agendamentos:**
  - Busca `execucoes` onde `agendamento_id` é nulo.
  - Para cada `execucao`, procura por um `agendamentos` correspondente usando:
    - `paciente_id` (obtido via `guias` e `carteirinhas` associadas à execução).
    - `data_execucao` comparada com `data_agendamento`.
    - `procedimento_id` (obtido da tabela `guias` associada à execução).

- Se **exatamente um** `agendamentos` for encontrado com essa combinação, atualiza `execucoes.agendamento_id`.

### 3. Propagação de Vínculo (Sessão -> Execução):

- Verifica `execucoes` que já estão vinculadas a uma `sessoes` (`execucoes.sessao_id` não é nulo).
- Se a `sessoes` vinculada possui um `agendamento_id` (obtido no passo 1 ou anteriormente), mas a `execucao` ainda não (`execucoes.agendamento_id` é nulo), copia o `agendamento_id` da `sessoes` para a `execucao`.

### 4. Propagação de Vínculo (Execução -> Sessão):

- Verifica `sessoes` que já estão vinculadas a uma `execucao` (`execucoes.sessao_id` aponta para a `sessoes`).
- Se a `execucao` vinculada possui um `agendamento_id` (obtido no passo 2 ou anteriormente), mas a `sessoes` ainda não (`sessoes.agendamento_id` é nulo), copia o `agendamento_id` da `execucao` para a `sessoes`.

## 3.4. Revisão Manual (Interface de Vinculação)

- Uma interface permite visualizar `execucoes` marcadas com `link_manual_necessario = true` ou que não possuem `sessao_id`.
- O usuário pode selecionar uma `execucao` e uma `sessao` para vinculá-las manualmente.
- Esta ação atualiza `execucoes.sessao_id`, `execucoes.codigo_ficha`, `execucoes.codigo_ficha_temp = false` e `execucoes.link_manual_necessario = false`.

## 4. Visualização Consolidada (VIEW `vw_agendamentos_com_status_vinculacao`)

- Uma VIEW no banco de dados (`vw_agendamentos_com_status_vinculacao`) consolida as informações.
- Para cada `agendamentos`, ela mostra:
  - Dados do agendamento, nome do paciente e nome do procedimento.
  - Se existe uma `sessoes` vinculada (`possui_sessao_vinculada`).
  - Se existe uma `execucoes` vinculada (`possui_execucao_vinculada`).
  - Um `status_vinculacao` geral ('Pendente', 'Ficha OK', 'Unimed OK', 'Completo', 'Divergência').
- A API (`GET /api/agendamentos`) utiliza esta VIEW implicitamente através da função RPC `func_listar_agendamentos_view` para fornecer os dados à página de Agendamentos no frontend. A função RPC refaz os joins necessários para permitir filtros e paginação dinâmica, e sua definição `RETURNS TABLE` foi cuidadosamente ajustada para corresponder aos tipos de dados reais das tabelas envolvidas.
- A interface do frontend permite filtrar a lista de agendamentos por este `status_vinculacao`.

## 5. Pontos Chave da Implementação Atual

- **Prioridade na Ordem:** A `ordem_execucao` é o principal critério (depois da guia e data) para a vinculação automática Sessão <-> Execução.
- **Tratamento de Ambiguidade:** Casos ambíguos (múltiplas correspondências) não são vinculados automaticamente e são sinalizados para revisão manual (`link_manual_necessario`).

- **Vinculação em Etapas:** A vinculação Sessão <-> Execução ocorre na inserção da Execução. A vinculação com Agendamentos ocorre em um processo batch separado.
- **Flexibilidade:** A vinculação batch de agendamentos pode ser acionada conforme necessário.
- **Interface de Apoio:** A interface de vinculação manual é crucial para resolver casos complexos ou corrigir erros.

Este fluxo atualizado visa maximizar a automação da vinculação com alta confiança, ao mesmo tempo que fornece mecanismos claros para lidar com ambiguidades e intervenção manual quando necessário.