

Arquitetura do sistema

O módulo é uma **biblioteca ISNAPP** (padrão de extensão do ERP) composta por três arquivos e integrada a múltiplos bancos MySQL.

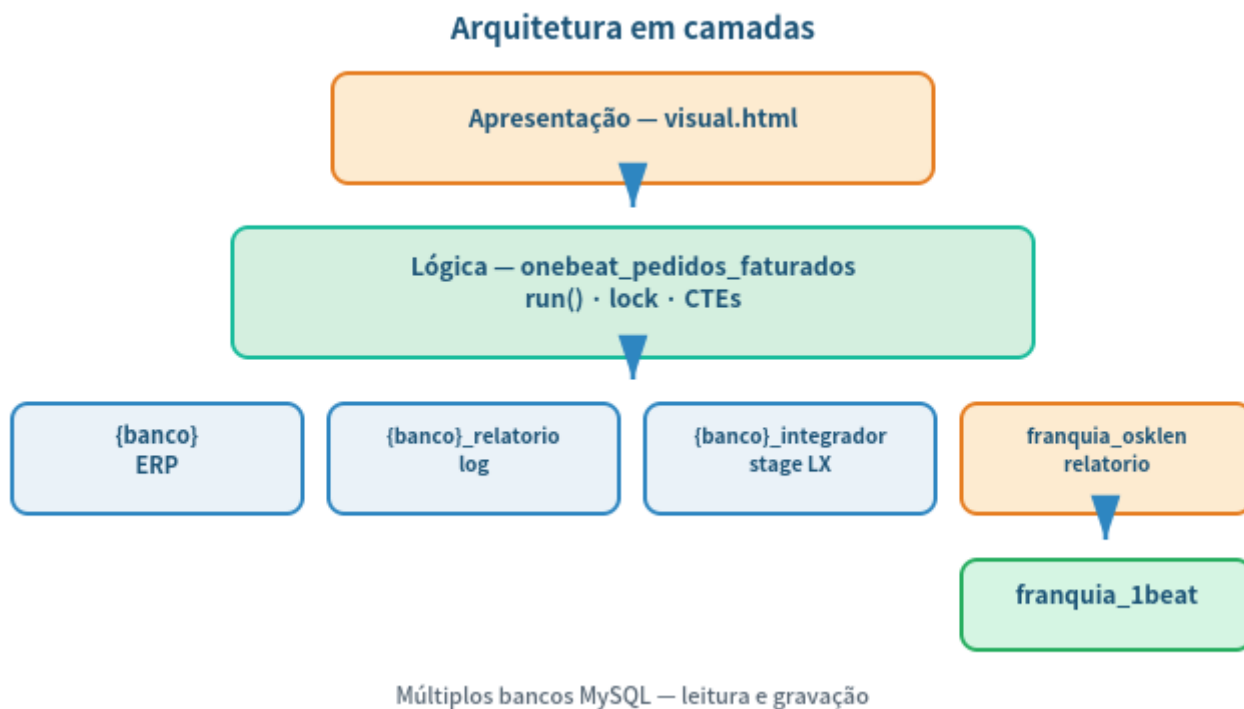


Figura 6 — Arquitetura em camadas: apresentação, lógica PHP e múltiplos bancos MySQL.

Bancos de dados envolvidos

Variável PHP	Nome do banco	Uso
<code>\$banco</code>	Dinâmico (tenant)	Pedidos, itens, entidades, recebimentos
<code>\$bancoRelatorio</code>	<code>{banco}_relatorio</code>	Tabela <code>log</code> com ponteiro de última execução
<code>\$bancoIntegrador</code>	<code>{banco}_integrador</code>	Stage de integração LX (<code>stage_pedido_lx</code>)
<code>\$bancoMasterRelatorio</code>	<code>franquia_osklen_relatorio</code>	Tabela de destino <code>franquia_1beat</code> (compartilhada)

Para entender sem jargão: o sistema lê informações de três "pastas" diferentes no banco de dados e grava o resultado organizado em uma "pasta central" que o OneBeat consulta.

Tabelas de origem (leitura)

Tabela	Banco	Papel
<code>movimentacao</code>	{banco}	Cabeçalho dos pedidos e recebimentos
<code>movimentacao_detalhe</code>	{banco}	Itens (produtos) de cada movimentação
<code>movimentacao_movimentacao</code>	{banco}	Vínculo pai-filho (pedido → recebimento)
<code>entidade</code>	{banco}	Dados da filial (loja)
<code>juridica</code>	{banco}	CNPJ da filial
<code>stage_pedido_lx</code>	{banco}_integrador	Integração com sistema LX (LEFT JOIN)
<code>log</code>	{banco}_relatorio	Controle de ponteiro (<code>procedure = 'franquia_1beat'</code>)

Classe principal

Classe: `onebeat_pedidos_faturados` extends `util`

Token: `6b5b2af8-ab25-4135-9551-b3cb3e055f1a`

CoreVersion: 3

Métodos públicos principais:

- `index()` — exibe visual.html
- `run()` — execução principal (sincronização)
- `verificaGetLock()` — controle de concorrência
- `getPonteiroExecucao()` — última data processada
- `getPedidoUltimaVerificacao()` — data do pedido mais recente
- `getPedidosRecebidos()` — INSERT/UPDATE na tabela destino
- `removePedidosStatus()` — limpeza de registros desativados
- `atualizaPonteiroExecucao()` — grava novo ponteiro

Mecanismo de lock (concorrência)

Para evitar que duas execuções rodem ao mesmo tempo, o módulo usa um **arquivo de lock** no diretório temporário do sistema operacional:

```
/tmp/{banco}.onebeat_pedidos_faturados_run.lock
```

Usa `flock(LOCK_EX | LOCK_NB)` — lock exclusivo não-bloqueante. Se outro processo já estiver rodando, lança exceção: *"Ja tem outro processo em execucao, aguarde o termino"*. O lock é liberado no bloco `finally` via `__releaseLock()`.

Revision #3

Created 2026-06-24 19:51:26 UTC by Gustavo Filgueiras

Updated 2026-06-25 12:34:06 UTC by Gustavo Filgueiras