

Arquitetura e Tecnologias

Stack técnico, hierarquia de classes, dependências e conexões com bancos de dados.

Diagrama de componentes

FullStore HTTP POST getVendas isnapp controle.php PHP 7.3 setSubmit() MySQL OSK PDV, Trocas Omni, filiais SQL Server Linx/BI LX_ZERO_300 W_BI_FAT_VENDEDOR... CNPJ + periodo CTEs + filiais prepared statement JSON success / message / data Figura 1 - Componentes e fluxo de dados da integração.

Stack

Camada	Tecnologia	Detalhe
Linguagem	PHP 7.3	Namespaces, PDO, Traits
Framework	CodeIgniter (MY_Controller)	Base do backend
Banco operacional	MySQL / PDO	PDV, Omni, Trocas
Banco E-commerce	SQL Server / PDO sqlsrv:	Linx LX_ZERO_300
Resposta	JSON via setSubmit()	Envelope padrão success / message / data

Hierarquia de classes

biblioteca base do loader fullStoreService carrega classe por UUID/token utilOsk PDO MySQL OSK WOSK\fullStoreService\WebService valida, consulta canais e monta resposta WOSK\Commons\WebService trait: setSubmit(), logs db_sqlserver PDO sqlsrv para e-commerce usa trait instancia Cadeia principal biblioteca -> fullStoreService / utilOsk -> WOSK\fullStoreService\WebService -> trait + db_sqlserver

Arquivos required

Em `controle.php`, carregamos três dependências via `require_once(APPPATH . '...')`:

UUID	Arquivo	Fornece
4cb18a6c-...	<code>wosk_webservice.php</code>	Trait <code>WOSK\Commons\WebService</code> — formata JSON (<code>setSubmit</code>), valida campos e salva logs no MySQL
592ccfa0-...	<code>util0sk.php</code>	Classe pai — conecta ao MySQL OSK e expõe <code>\$this->conexao</code> (PDO)
0670acec-...	<code>db_sqlserver.php</code>	Gerencia conexão PDO com SQL Server (driver <code>sqlsrv:</code>)

Conexão MySQL

Como configuramos

A classe `util0sk` chama `getConfiguracao('db_osk_prod')` — credenciais armazenadas em Base64 na tabela de configuração do sistema. Em seguida, `_setConexao()` cria a conexão PDO:

```
// connection string:
"mysql:host={host};port={porta};dbname={banco}"
// PDO::ATTR_PERSISTENT = true (conexão persistente)
// PDO::ATTR_TIMEOUT = 120 (timeout em segundos)
// charset = UTF-8
```

Conexão SQL Server

Instanciamos em `buscarDadosEcommercePorLoja()`

```
$sqlserver = new \db_sqlserver(false, 'LX_ZERO_300', true);
// false = produção (não homologação)
// 'LX_ZERO_300' = banco Linx
// true = conecta imediatamente
```

Ambiente	Host	Porta	Banco
Produção	<code><HOST_SQLSERVER_PROD></code>	<code><PORTA_PROD></code>	<code>LX_ZERO_300</code>

Ambiente	Host	Porta	Banco
Homologação	<HOST_SQLSERVER_HML>	<PORTA_HML>	LINUX_HMLG

Credenciais hardcoded As credenciais do SQL Server ficam em `db_sqlserver.php`.

Tabelas MySQL utilizadas

Tabela	Alias	O que fornece
<code>movimentacao</code>	m	Cabeçalho da transação (tipo, data, situação, módulo)
<code>movimentacao_detalle</code>	md	Itens da venda (produto, qtde, preço)
<code>movimentacao_nfe</code>	nf / nfe	Número, série e situação da NF-e
<code>movimentacao_movimentacao</code>	mm	Vínculos entre movimentações (troca↔venda, omni↔orçamento)
<code>pessoa</code>	v	Nome e código do vendedor
<code>entidade</code>	e	Código da filial / loja
<code>juridica</code>	j	CNPJ da empresa

Objetos SQL Server utilizados

Objeto	Tipo	O que fornece
<code>W_BI_FAT_VENDEDOR_VDK_OMNI_V0</code>	View BI	Pedidos e-commerce por vendedor: NF, valor, CPF, filial, pedido site/WMS
<code>FILIAIS</code>	Tabela	Cadastro de filiais — usamos para excluir franquias (<code>TIPO_FILIAL <> 'FRANQUIA'</code>)

Revision #2

Created 2026-06-09 18:23:12 UTC by Mauricio Francelino

Updated 2026-06-09 18:52:47 UTC by Mauricio Francelino