



Sistema de Controle de Impressão

Venturize Hotelaria

Autor: Ferraz Tecnologia

Versão 1.0 | Novembro 2025

1. Visão Geral

Este documento descreve a implementação do **Sistema de Controle de Impressão** para o sistema Venturize Hotelaria. A solução permite rastreamento completo do processo de impressão de pedidos, incluindo histórico, status e integração com agentes de impressão externos.

Objetivo: Implementar um sistema robusto de controle de impressão que permita rastrear todas as tentativas de impressão, identificar pedidos pendentes e fornecer uma API completa para integração com agentes de impressão externos.

2. Arquitetura da Solução

2.1 Componentes Principais

- **Tabela de Impressões:** Armazena histórico completo de todas as tentativas de impressão
- **Model ImpressaoPedido:** Gerencia os registros de impressão com relacionamentos e scopes
- **API REST:** Endpoints para integração com agentes de impressão externos
- **Controller PrintController:** Lógica de negócio para gerenciamento de impressões

2.2 Fluxo de Trabalho

1. Usuário solicita impressão de cupom parcial através da interface web
2. Sistema cria registro na tabela `impressoes_pedidos` com status "pendente"
3. Dados do pedido são disponibilizados via API em formato JSON
4. Agente de impressão externo monitora e processa pedidos pendentes
5. Agente atualiza status da impressão (sucesso/erro) via API
6. Sistema mantém histórico completo para auditoria

3. Estrutura do Banco de Dados

3.1 Tabela: `impressoes_pedidos`

A tabela `impressoes_pedidos` armazena todas as tentativas de impressão de pedidos.

Campo	Tipo	Descrição
<code>id</code>	BIGINT UNSIGNED	Chave primária auto-incrementável
<code>pedido_id</code>	BIGINT UNSIGNED	Foreign key para tabela <code>pedidos</code> (CASCADE DELETE)

Campo	Tipo	Descrição
agente_impressao	VARCHAR(255)	Identificação do agente que processou a impressão (padrão: 'sistema')
ip_origem	VARCHAR(255)	IP de origem da solicitação (nullable)
status_impressao	ENUM	Status: 'pendente', 'sucesso', 'erro' (padrão: 'pendente')
detalhes_erro	TEXT	Detalhes do erro caso status seja 'erro' (nullable)
dados_impressao	JSON	Dados extras como impressora usada, timestamps, etc. (nullable)
created_at	TIMESTAMP	Data/hora de criação do registro
updated_at	TIMESTAMP	Data/hora da última atualização

3.2 Índices

Para otimizar consultas, foram criados os seguintes índices:

- Índice composto: `(pedido_id, status_impressao)`
- Índice simples: `created_at`
- Índice simples: `status_impressao`

4. Modelos e Relacionamentos

4.1 Model: ImpressaoPedido

Localização: app/Models/Bar/ImpressaoPedido.php

Relacionamentos:

- pedido() - BelongsTo Pedido

Scopes Disponíveis:

- pendentes() - Filtra impressões com status 'pendente'
- sucesso() - Filtra impressões com status 'sucesso'
- comErro() - Filtra impressões com status 'erro'
- hoje() - Filtra impressões criadas hoje
- porAgente(\$agente) - Filtra por agente de impressão

Métodos Úteis:

- marcarComoSucesso(\$dadosExtras) - Marca impressão como sucesso
- marcarComoErro(\$detalhesErro, \$dadosExtras) - Marca impressão como erro

4.2 Model: Pedido (Atualizações)

Localização: app/Models/Bar/Pedido.php

Novos Relacionamentos:

- impressoes() - HasMany ImpressaoPedido
- ultimaImpressao() - HasOne ImpressaoPedido (latest)

Novos Métodos:

- foiImpresso() - Retorna true se pedido foi impresso com sucesso
- temImpressaoPendente() - Retorna true se há impressão pendente
- totalImpressoes() - Retorna total de impressões bem-sucedidas

Novos Scopes:

- `impressos()` - Filtra pedidos já impressos
- `naoImpressos()` - Filtra pedidos não impressos

5. API de Impressão

5.1 Endpoints Disponíveis

5.1.1 Verificar Status de Impressão

GET /api/print/pedido/{pedidoId}/status

Verifica o status de impressão de um pedido específico.

Resposta de Sucesso:

```
{  
    "success": true,  
    "data": {  
        "pedido_id": 123,  
        "foi_impresso": true,  
        "tem_impressao_pendente": false,  
        "pode_imprimir_diretamente": false,  
        "requer_confirmacao": true,  
        "ultima_impressao": {  
            "id": 1,  
            "status": "sucesso",  
            "agente": "AgentePrint_v1.0",  
            "data": "15/01/2025 16:30:00"  
        },  
        "total_impressoes": 1  
    },  
    "message": "Status de impressão verificado com sucesso"  
}
```

5.1.2 Obter Dados do Pedido para Impressão

GET /api/print/pedido/{pedidoId}

Retorna todos os dados estruturados de um pedido em formato JSON para impressão.

Parâmetros Opcionais (Query String):

- `forcar_impressao` - Força impressão mesmo se já foi impresso (true/false)
- `impressao_id` - ID específico da impressão para atualizar
- `agente` - Nome do agente de impressão

Resposta de Sucesso:

```
{  
    "success": true,  
    "data": {  
        "pedido": {  
            "id": 174,  
            "status": "aberto",  
            "total": 45.50,  
            "taxa_servico": 4.55,  
            "total_com_taxa": 50.05,  
            "pedido_apartamento": false,  
            "observacoes": "Sem gelo",  
            "created_at": "15/01/2025 14:30:00",  
            "updated_at": "15/01/2025 15:45:00"  
        },  
        "mesa": {  
            "numero": "05",  
            "status": "ocupada"  
        },  
        "cliente": {  
            "id": 123,  
            "nome": "João Silva",  
            "cpf": "123.456.789-00",  
            "telefone": "(11) 99999-9999"  
        },  
        "reserva": {  
            "id": 456,  
            "quarto": {  
                "numero": "101",  
                "tipo": "Standard"  
            }  
        },  
        "itens": [...],  
        "empresa": {...},  
        "timestamp_impressao": "15/01/2025 15:45:30",  
        "tipo_cupom": "parcial"  
    },  
    "impressao_id": 1,  
    "message": "Dados do pedido gerados com sucesso para impressão"  
}
```

5.1.3 Listar Pedidos Pendentes

GET /api/print/pedidos-pendentes

Retorna lista de pedidos abertos que possuem impressões pendentes.

Resposta de Sucesso:

```
{  
    "success": true,  
    "data": [  
        {  
            "id": 174,  
            "mesa_numero": "05",  
            "cliente_nome": "João Silva",  
            "quarto_numero": "101",  
            "total": 45.50,  
            "pedido_apartamento": false,  
            "updated_at": "15/01/2025 15:45:00",  
            "itens_count": 3,  
            "tem_impressao_pendente": true,  
            "ultima_tentativa_impressao": "15/01/2025 15:45:00"  
        }  
    ],  
    "count": 1,  
    "message": "Lista de pedidos pendentes obtida com sucesso"  
}
```

5.1.4 Marcar Pedido como Impresso

POST /api/print/pedido/{pedidoId}/impresso

Marca um pedido como impresso com sucesso.

Body (JSON):

```
{  
    "agente_impressao": "AgentePrint_v1.0",  
    "impressao_id": 1,  
    "impressora": "Impressora_Cozinha_01",  
    "metodo": "api"  
}
```

Resposta de Sucesso:

```
{  
    "success": true,  
    "message": "Pedido marcado como impresso com sucesso",  
    "pedido_id": 174,  
    "impressao_id": 1,  
    "timestamp": "15/01/2025 16:30:00",  
    "total_impressoes": 1  
}
```

5.1.5 Registrar Tentativa de Impressão

POST /api/print/pedido/{pedidoId}/tentativa

Registra quando o agente inicia o processamento de uma impressão (status: 'processando').

Body (JSON):

```
{  
    "agente_impressao": "AgentePrint_v1.0",  
    "impressao_id": 1,  
    "impressora": "Impressora_Cozinha_01"  
}
```

5.1.6 Marcar Erro na Impressão

POST /api/print/pedido/{pedidoId}/erro

Registra erro ocorrido durante a impressão.

Body (JSON):

```
{  
    "agente_impressao": "AgentePrint_v1.0",  
    "impressao_id": 1,  
    "erro": "Impressora offline",  
    "detalhes_tecnicos": "Conexão recusada na porta 9100"  
}
```

5.1.7 Histórico de Impressões

GET /api/print/pedido/{pedidoId}/historico

Retorna histórico completo de todas as tentativas de impressão de um pedido.

Resposta de Sucesso:

```
{  
    "success": true,  
    "data": [  
        {  
            "id": 1,  
            "agente": "AgentePrint_v1.0",  
            "status": "sucesso",  
            "ip_origem": "192.168.1.100",  
            "detalhes_erro": null,  
            "dados_impressao": {...},  
            "created_at": "15/01/2025 16:30:00",  
            "updated_at": "15/01/2025 16:30:15"  
        }  
    ],  
    "total_impressoes": 1,  
    "impressoes_sucesso": 1,  
    "impressoes_erro": 0,  
    "foi_impresso": true  
}
```

5.1.8 Estatísticas de Impressão

GET /api/print/estatisticas

Retorna estatísticas gerais do sistema de impressão.

Resposta de Sucesso:

```
{  
    "success": true,  
    "data": {  
        "pedidos_pendentes": 5,  
        "pedidos_impressos_hoje": 42,  
        "total_impressoes_hoje": 45,  
        "impressoes_com_erro_hoje": 3,  
        "agentes_ativos": ["AgentePrint_v1.0", "AgentePrint_v2.0"]  
    },  
    "timestamp": "15/01/2025 18:00:00"  
}
```

6. Passo a Passo de Instalação

Importante: Antes de iniciar a instalação, certifique-se de ter um backup completo do banco de dados e dos arquivos do sistema.

6.1 Pré-requisitos

- PHP 8.2 ou superior
- Composer instalado
- Laravel 11
- Banco de dados MySQL/PostgreSQL configurado
- Acesso SSH ou FTP ao servidor
- Permissões de escrita nas pastas `storage` e `bootstrap/cache`

6.2 Passo 1: Backup do Sistema

Execute um backup completo antes de prosseguir:

```
# Backup do banco de dados
mysqldump -u usuario -p nome_banco > backup_antes_instalacao.sql

# Backup dos arquivos (se necessário)
tar -czf backup_arquivos.tar.gz /caminho/do/projeto
```

6.3 Passo 2: Verificar Arquivos da Implementação

Certifique-se de que os seguintes arquivos estão presentes no projeto:

- `database/migrations/2025_01_15_160000_create_impressoes_pedidos_table.php`
- `app/Models/Bar/ImpressaoPedido.php`
- `app/Http/Controllers/Api/PrintController.php` (**atualizado**)
- `app/Models/Bar/Pedido.php` (**atualizado**)
- `routes/api.php` (**com rotas de impressão**)

6.4 Passo 3: Executar Migration

Execute a migration para criar a tabela no banco de dados:

```
# Acesse o diretório do projeto  
cd /caminho/do/projeto/venturize-hotelaria  
  
# Execute a migration  
php artisan migrate
```

Nota: Se a migration já foi executada anteriormente, o Laravel detectará e não executará novamente. Para forçar a execução, use `php artisan migrate:refresh` (cuidado: isso apagará dados existentes).

6.5 Passo 4: Verificar Relacionamentos no Model Pedido

Abra o arquivo `app/Models/Bar/Pedido.php` e verifique se contém os seguintes métodos e relacionamentos:

- Relacionamento `impressoes()`
- Relacionamento `ultimaImpressao()`
- Método `foiImpresso()`
- Método `temImpressaoPendente()`
- Método `totalImpressoes()`
- Scopes `impressos()` e `naoImpressos()`

6.6 Passo 5: Verificar Rotas da API

Abra o arquivo `routes/api.php` e verifique se as rotas de impressão estão configuradas:

```
Route::prefix('print')->name('api.print.')->group(function () {  
    Route::get('/pedido/{pedidoId}/status', [PrintController::class, 'verif  
    Route::get('/pedido/{pedidoId}', [PrintController::class, 'getPedidoFor  
    Route::get('/pedidos-pendentes', [PrintController::class, 'getPedidosPe  
    Route::post('/pedido/{pedidoId}/impresso', [PrintController::class, 'ma  
    Route::post('/pedido/{pedidoId}/tentativa', [PrintController::class, 'r  
    Route::post('/pedido/{pedidoId}/erro', [PrintController::class, 'marcar  
    Route::get('/pedido/{pedidoId}/historico', [PrintController::class, 'ge
```

```
Route::get('/estatisticas', [PrintController::class, 'getEstatisticasImagem']);
```

6.7 Passo 6: Limpar Cache

Limpe todos os caches do Laravel para garantir que as mudanças sejam reconhecidas:

```
php artisan cache:clear  
php artisan config:clear  
php artisan route:clear  
php artisan view:clear  
  
# Reconstruir cache de rotas (recomendado para produção)  
php artisan route:cache
```

6.8 Passo 7: Verificar Permissões

Certifique-se de que as pastas têm permissões corretas:

```
chmod -R 775 storage bootstrap/cache  
chown -R www-data:www-data storage bootstrap/cache
```

Nota: Ajuste o usuário/grupo (`www-data`) conforme sua configuração de servidor.

6.9 Passo 8: Testar a Instalação

Execute os seguintes testes para verificar se a instalação foi bem-sucedida:

Teste 1: Verificar Tabela no Banco de Dados

```
# Acesse o MySQL/PostgreSQL  
mysql -u usuario -p nome_banco  
  
# Execute:  
DESCRIBE impressoes_pedidos;  
# ou  
SELECT * FROM impressoes_pedidos LIMIT 1;
```

Teste 2: Testar Endpoint de Estatísticas

```
# Via curl
curl -X GET "http://seu-dominio.com/api/print/estatisticas"

# Deve retornar JSON com estatísticas
```

Teste 3: Testar Listagem de Pedidos Pendentes

```
curl -X GET "http://seu-dominio.com/api/print/pedidos-pendentes"
```

7. Configuração do Agente de Impressão

7.1 Agente de Impressão

Implementando em Node.js (executável rodando a parte e em background):

8. Tratamento de Erros

8.1 Códigos de Erro da API

Código	Descrição
CHECK_PRINT_STATUS_ERROR	Erro ao verificar status de impressão
PRINT_DATA_ERROR	Erro ao gerar dados para impressão
PENDING_ORDERS_ERROR	Erro ao obter pedidos pendentes
MARK_PRINTED_ERROR	Erro ao marcar pedido como impresso
REGISTER_ATTEMPT_ERROR	Erro ao registrar tentativa de impressão
REGISTER_ERROR_ERROR	Erro ao registrar falha de impressão
GET_HISTORY_ERROR	Erro ao obter histórico de impressões
GET_STATS_ERROR	Erro ao obter estatísticas

Código	Descrição
IMPRESSAO_NOT_FOUND	Registro de impressão não encontrado
ALREADY_PRINTED	Pedido já foi impresso anteriormente

8.2 Formato de Resposta de Erro

Todos os endpoints retornam erros no seguinte formato:

```
{  
    "success": false,  
    "message": "Descrição do erro",  
    "error_code": "CODIGO_DO_ERRO"  
}
```

9. Segurança e Boas Práticas

9.1 Monitoramento

- Monitore o endpoint de estatísticas regularmente
- Configure alertas para alta taxa de erros
- Revise logs de impressão periodicamente
- Monitore performance das consultas ao banco de dados

10. Troubleshooting

10.1 Problemas Comuns

Problema: Migration não executa

Solução:

- Verifique se a migration já foi executada: `php artisan migrate:status`
- Verifique permissões do banco de dados
- Verifique se a tabela já existe: `SHOW TABLES LIKE 'impressoes_pedidos'`

Problema: Rotas não funcionam

Solução:

- Limpe cache de rotas: `php artisan route:clear`
- Reconstrua cache: `php artisan route:cache`
- Verifique se o arquivo `routes/api.php` está sendo carregado

Problema: Erro 500 ao acessar endpoints

Solução:

- Verifique logs: `tail -f storage/logs/laravel.log`
- Verifique se o model `ImpressaoPedido` existe
- Verifique relacionamentos no model `Pedido`
- Verifique permissões de escrita no storage

Problema: Agente não encontra pedidos pendentes

Solução:

- Verifique se há pedidos abertos com itens
- Verifique se há registros de impressão pendentes na tabela
- Teste o endpoint manualmente: `curl http://seu-dominio.com/api/print/pedidos-pendentes`

11. Manutenção e Atualizações

11.1 Limpeza de Dados Antigos

Para manter o banco de dados otimizado, considere criar um comando Artisan para limpar registros antigos:

```
# Exemplo de comando para limpar impressões antigas (mais de 90 dias)
php artisan make:command LimparImpressoesAntigas

# No comando, execute:
ImpressaoPedido::where('created_at', '<', now()->subDays(90))
    ->where('status_impressao', 'sucesso')
    ->delete();
```

11.2 Backup Regular

Configure backups regulares da tabela `impressoes_pedidos`:

```
# Script de backup diário
mysqldump -u usuario -p nome_banco impressoes_pedidos > backup_impressoes_`
```

12. Suporte e Contato

Para suporte técnico ou dúvidas sobre a implementação, entre em contato:

Ferraz Tecnologia

Documentação gerada em: Novembro 2025

Versão do Sistema: 1.0