

# Suroga - workbooks

É crucial que o *backoffice* da Suroga tenha acesso aos *workbooks* detalhados nas listas do MS365/SharePoint, pois esta plataforma serve como o **hub operacional e de revisão de dados** antes que a informação crítica seja inserida ou alterada no sistema principal (*Django administration interface*).

O acesso a estes *workbooks* (Dashboard e Ride Registry) é essencial para garantir a eficiência operacional, o cumprimento financeiro e a conformidade regulamentar.

Em resumo, a importância reside no facto de que o *backoffice* da Suroga não apenas visualiza estes dados, mas também os **utiliza ativamente para o trabalho administrativo e de validação** (seja gestão de fornecedores, inventário, ou resolução de problemas). Os *workbooks* transformam a complexa estrutura de dados da API Django em interfaces comprehensíveis e acionáveis para as operações diárias.

## 1. Dashboard (Summary View): Visão Operacional e Financeira em Tempo Real

A Dashboard fornece uma **visão geral operacional em tempo real** (*Real-time operational overview*) [Query Table], permitindo que a equipa de *backoffice* execute verificações rápidas de saúde (*operational health check*) [Query Table] e tome decisões informadas sobre recursos, liquidez e cumprimento fiscal.

Métrica	Importância para o Backoffice	Supporte nos Fontes
<b>Active Rides</b> $(=COUNTIF('rides_registry'!A2:A, "active"))$	Permite o acompanhamento imediato dos serviços de streaming em andamento. A gestão do ciclo de vida das Rides (corridas) deve ser eficiente e eficaz.	As Rides (corridas) no sistema Suroga possuem diversos status, como Blocked, Canceled, Deleted e Closed, que são vitais para o fluxo de trabalho.
<b>Unpaid Invoices</b> $(=COUNTIF('invoices'!L:L, "issued"))$	Monitoriza a liquidez e o cumprimento dos termos de pagamento. A Suroga API é central na gestão do fluxo financeiro, emitindo faturas à Suroo App e recebendo faturas de Fornecedores e Greenconnections.	O sistema de <b>Invoicing &amp; Billing</b> ( <i>Faturação e Cobrança</i> ) rastreia o status das faturas (e.g., "Issued" - Emitida, "Overdue" - Vencida). A fiabilidade do pagamento é uma preocupação fundamental para os Fornecedores.
<b>Avg. Response Time</b> $(=AVERAGE('messages'!G:G))$	É um KPI (Indicador Chave de Desempenho) crucial para avaliar a qualidade e a rapidez do suporte. A Suroga facilita a comunicação contínua através do seu sistema de mensagens.	O modelo <b>MessageBoard</b> é usado para gerir as comunicações.
<b>Supplier Utilization</b> $(='stock')$	Garante que os Fornecedores (Agências) disponíveis e o stock de serviços sejam	O sistema precisa gerir ativamente o stock de serviços para publicidade à Suroo APP e rastrear a utilização do stock

Métrica	Importância para o Backoffice	Supporte nos Fontes
<code>scheduler' !B5)</code>	alocados de forma eficiente. A Suroga API gera Fornecedores e utiliza filtragem avançada para fazer a correspondência com a procura de serviços.	(Conforme a Seção RideMkt modelo de mercado de serviços).
<b>VAT Liability</b> <code>(=SUM('invoicing'!I:I))</code>	Essencial para o <b>cumprimento fiscal</b> . A Suroga lida com o cálculo do IVA (VAT), incluindo mecanismos de <i>reverse-charge</i> para transações B2B intra-UE.	Para o cumprimento total das regras Portuguesas de SAF-T (necessário para a Suroga em Portugal), o sistema depende de um <b>software certificado por terceiros</b> integrado com o Stripe, e não apenas do Stripe. Monitorizar a responsabilidade do IVA é, portanto, vital para a conformidade legal.

## 2. Ride Registry: Rastreamento do Ciclo de Vida e Conformidade

O *Ride Registry* (Registo de Corridas) é fundamental porque **centraliza o rastreamento do ciclo de vida da Ride**, criando uma **trilha de auditoria compatível com o GDPR** (Regulamento Geral sobre a Proteção de Dados) [Query Table].

O *backoffice* precisa de ter este acesso para:

- Validação e Gestão de Dados:** O *backoffice* da Suroga utiliza as listas do SharePoint para **compilar e rever alterações** em informações, garantindo a integridade dos dados e o cumprimento das regras de negócio antes de serem inseridas no sistema principal Django.

Coluna	Finalidade Operacional	Mapeamento no Modelo de Dados
<b>Ride ID</b> <code>(="RIDE - "&amp;TEXT(ROW(), "00000"))</code>	Assegura que cada serviço tenha um identificador único, crucial para o sistema. O seu rastreamento ajuda a <b>prevenir dupla-marcação</b> ( <i>double-booking</i> ) [Query Table].	Mapeado para <code>Ride.pk</code> (Chave Primária do modelo Ride) [Query Table, 229].
<b>Requisits (JSON)</b>	Armazena os requisitos técnicos detalhados para o serviço. O <i>backoffice</i> utiliza este <i>field</i> para determinar a elegibilidade do Fornecedor e o <i>stock</i> necessário.	Mapeado para <code>Ride.Requisits</code> , um campo de 1000 caracteres no modelo Ride.
<b>Status</b> (Dropdown: Active/Blocked/Canceled/Closed)	Permite que o <i>backoffice</i> verifique e confirme o estado de cada transação, assegurando que o fluxo operacional seja seguido (e.g., impedindo a eliminação de <i>Rides</i> que não foram canceladas).	Mapeado para os campos booleanos de <code>status</code> no modelo <code>Ride</code> (e.g., <code>Ride.Blocked</code> , <code>Ride.Canceled</code> , <code>Ride.Closed</code> ).
<b>Time-to-Confirm</b>	Mede o tempo desde o registo até ao início da <i>Ride</i> . Este é um marco vital, pois o início da <i>Ride</i> ( <code>RideStartDate</code> ) é usado para medir o desempenho e, implicitamente, para acionar processos de faturação ou registo	Mapeado para <code>Booking.RideStartDate</code> , um campo de data e hora no modelo <code>Booking</code> .

Coluna	Finalidade Operacional	Mapeamento no Modelo de Dados
<b>GPS Coordinates</b> (=CONCATENATE( [@lat], ", ", [@long]))	de tempo. Permite rastrear a localização exata do serviço. Os serviços Suroga são iniciados marcando a presença do utilizador numa localização geográfica específica.	Mapeado para <code>Ride.GPS_latitude/longitude [Query Table]</code> .

### 3. Ride Configuration (Service Parameter Master)

Este *workbook* é o **mestre de parâmetros de serviço** [Query Table], sendo essencial para **aplicar parâmetros de serviço padronizados em todos os Fornecedores** (*suppliers*) e prevenir perdas de receita através de preços dinâmicos [Query Table]. O *backoffice* usa esta interface para configurar as regras que, no *backend* do Django, mapeiam os serviços (**servicetype**) com os termos contratuais e geográficos (**contractsection**).

Parâmetro	Importância para o Backoffice	Supporte nos Fontes
<b>Service ID / Name</b> ( <b>servicetype.id</b> / <b>servicetype.Name</b> )	O <b>servicetype</b> (tipo de serviço) é a base para categorizar as ofertas de serviços na plataforma. O <b>backoffice</b> gere o <b>stock</b> de serviços e as suas configurações (equipamento, regras, duração) a partir daqui.	O modelo <b>servicetype</b> define nome, duração, equipamento e regras de serviço (em JSON).
<b>Min Duration</b> ( <b>servicetype.Duration</b> )	Define a duração mínima do serviço que a Suroga API publicita à Suroo App. Isto está ligado ao modelo <b>RideDuration</b> (Duração da Corrida), assegurando que apenas durações predefinidas sejam utilizadas.	A duração ( <b>Duration</b> ) é um campo de chave estrangeira ( <b>ForeignKey</b> ) no modelo <b>servicetype</b> ligando ao modelo <b>RideDuration</b> .
<b>Required Equipment</b> ( <b>servicetype.Equipment</b> )	Determina qual o equipamento necessário para cada tipo de serviço. Isto é vital para o <i>Demand-Driven Matching</i> (Correspondência Orientada pela Procura), onde o <i>backoffice</i> deve garantir que o equipamento dos Fornecedores é compatível.	O <b>servicetype</b> inclui um campo <b>Equipment</b> (equipamento) que é uma <b>ForeignKey</b> para o modelo <b>equipment</b> .
<b>NUTS Codes</b>	Permite que o <i>backoffice</i> O modelo <b>contractsection</b>	

Parâmetro	Importância para o Backoffice	Supporte nos Fontes
(contractsection.nuts0-3)	defina com precisão a <b>cobertura territorial</b> de cada secção contratual. Os Fornecedores usam estas regiões NUTS (NUTS0 até NUTS3) para definir as suas áreas de serviço.	gerencia cláusulas contratuais e a <b>cobertura territorial via NUTS</b> .
<b>Base Price (€) / Peak Surcharge</b> (contractsection.variante)	Estas colunas controlam a <b>estrutura de preços</b> . O preço de base e a lógica de sobretaxa ( <i>Peak Surcharge</i> ) são cruciais para o cálculo de preços dinâmicos e para prevenir a fuga de receita ( <i>revenue leakage</i> ) [Query Table].	Os preços dinâmicos e variações contratuais são armazenados no campo <b>variante</b> (formato JSON) do modelo <b>contractsection</b> . Esta informação é usada em funções de cálculo de preços que consideram a data e hora do serviço para determinar o preço final (como visto na função <b>price</b> do sistema).

O *backoffice* precisa desta interface de alto nível (SharePoint) porque as modificações na informação do Fornecedor e dos contratos são realizadas principalmente através da interface de administração Django, com as listas SharePoint a serem usadas para **compilar e rever as alterações** antes da sua submissão final, garantindo a integridade dos dados e o cumprimento das regras de negócio.

## 4. Payment Control (Financial Flow Management)

Este *workbook* é vital para a **gestão do fluxo financeiro** e para **assegurar o cumprimento total do IVA** (VAT compliance) em todas as jurisdições da UE [Query Table]. A Suroga API atua como o **hub central de pagamentos**.

Coluna	Importância para o Backoffice	Supporte nos Fontes
<b>Payment ID</b> (payment log.pk)	Garante a rastreabilidade de cada transação no sistema de registo de pagamentos. O sistema de pagamento rastreia transações com IDs específicos, como <b>StripTransactionID</b> e <b>SurooBankTransactionID</b> .	Mapeia para a chave primária (pk) do modelo <b>payment log</b> , que registra as transações de pagamento.
<b>Ride Link</b> (payment log.Ride)	Associa inequivocavelmente o pagamento a uma <b>Booking</b> (Reserva) específica. Isto é essencial para conciliar os pagamentos (e.g., da Suroo App à Fornecedor) com o serviço prestado.	O <b>payment log</b> possui uma chave estrangeira <b>Ride</b> (que se refere a um objeto <b>Booking</b> ).

Coluna	Importância para o Backoffice	Supporte nos Fontes
<b>Amount Due</b> (paymentlog . Value)	O campo <b>Value</b> no <b>payment log</b> regista o montante da transação. Este valor reflete o custo cobrado (à Suroo App) ou o valor a pagar (ao Fornecedor, Greenconnections ou Greenstreaming).	A fórmula no <i>workbook</i> ( <i>xlookup</i> ) recupera o valor da fatura do "Ride Registry", garantindo que o valor registado no <b>payment log . Value</b> está correto [Query Table, 616].
<b>Status</b> (paymentlog . OperationType)	Permite ao <i>backoffice</i> monitorizar o estado de todos os fluxos de caixa. O <b>payment log</b> regista o <b>OperationType</b> (Tipo de Operação), que inclui o estado do pagamento.	O campo <b>OperationType</b> no <b>payment log</b> define o tipo de transação (e.g., "Issued," "Paid," "Disputed").
<b>VAT % (Tax compliance)</b>	Esta fórmula complexa (incluindo a taxa de IVA Portuguesa de 23% para PT e consultas VLOOKUP para outros países) é crucial [Query Table]. O <i>backoffice</i> monitoriza isto porque a Suroga tem de gerir o IVA e o mecanismo de <i>reverse-charge</i> em transações B2B intra-UE.	O modelo <b>country</b> armazena o campo VAT (IVA), com um valor padrão de <b>23.0%</b> . Embora o Stripe automatize o cálculo do IVA, a conformidade total (como a geração SAF-T em Portugal) requer supervisão e integração com <i>software</i> certificado, tornando a monitorização do IVA pelo <i>backoffice</i> um passo crítico.
<b>Days Overdue</b> (paymentlog . SettlementDate)	O registo da data de liquidação ( <b>SettlementDate</b> ) e a subsequente verificação de dias em atraso são essenciais para manter a <b>fiabilidade dos pagamentos</b> aos Fornecedores, que é uma das suas principais preocupações.	A coluna usa o <b>SettlementDate</b> do <b>payment log</b> para calcular o atraso, ajudando a gestão do risco de pagamento.

Em suma, o *backoffice* da Suroga utiliza o **Payment Control** como uma folha de cálculo mestre para auditar e gerir a complexa rede de pagamentos e obrigações fiscais (IVA), que são centrais para o modelo de negócio da Suroga API, que paga aos Fornecedores e cobra à Suroo App.

## 5. Stock Scheduler (Supplier availability matching)

Este *workbook* é crucial para o **Demand-Driven Matching** (Correspondência Orientada pela Procura) e para a gestão eficiente do *stock* de serviços. Permite ao *backoffice* visualizar e gerir a capacidade dos Fornecedores de fornecerem serviços em tempo real.

Coluna	Justificativa Operacional e Mapeamento de Dados	Supporte nos Fontes
<b>Supplier</b> (profilerepresentative.Name)	O <i>backoffice</i> precisa de identificar o Fornecedor (Agência) e o seu representante. O acesso à lista de nomes de representantes ( <code>profilerepresentative.Name</code> ) permite associar o <i>stock</i> diretamente à pessoa responsável pela operacionalização do serviço.	O modelo <code>profilerepresentative</code> armazena o nome e o vínculo à agência Fornecedor.
<b>NUTS Region</b> (contractsection.nuts3)	Essencial para a <b>definição precisa da área de serviço</b> do Fornecedor. A Suroga utiliza códigos NUTS (do NUTS0 ao NUTS3) para classificar o território. Esta coluna garante que a capacidade mapeada está associada à cobertura geográfica contratual, tal como definido no modelo <code>contractsection</code> .	Os Fornecedores definem as suas áreas de serviço usando as Regiões NUTS. O modelo <code>contractsection</code> inclui campos <code>nuts0</code> a <code>nuts3</code> .
<b>Capacity</b> (SectonStock.capacity)	Representa a capacidade de serviço que o Fornecedor tem contratualmente disponível. Esta informação é vital, pois o modelo <code>contractsection</code> contém um campo <code>SectonStock</code> em formato JSON, usado para gerir o inventário de serviços.	A Suroga geraativamente o <i>stock</i> de serviços. O campo <code>SectonStock</code> na secção do contrato armazena estes detalhes.
<b>Booked % / Conflict Alert</b>	Permite ao <i>backoffice</i> monitorizar a utilização em tempo real e identificar sobrecarga de recursos (alertas H/M/L) [Query Table]. Isto é crítico para garantir a <b>eficiência e a escalabilidade</b> do agendamento.	O sistema Django automatiza o agendamento (Booking viewsets) e a gestão de <i>stock</i> (ContrSectionRideMkt model), mas o <i>backoffice</i> necessita da visualização agregada para intervir e ajustar o <i>stock</i> na fonte (SharePoint).

Acesso a este *workbook* permite que o *backoffice* mantenha a **integridade dos dados de inventário** e assegure que o sistema de *matching* da Suroga API identifique Fornecedores elegíveis com base na procura e nos requisitos geográficos.

## 6. Price Hub (Supplier negotiation tracker)

Este *workbook* de Negociação é vital para **documentar exceções** e manter a **rastreabilidade financeira**, funcionando como uma barreira de proteção (*guardrail*) contra preços inconsistentes [Query Table]. É essencial porque o preço final pago ao Fornecedor deve ser validado, mesmo que a Suroga API pague aos Fornecedores a uma *tabela variável baseada em contrato*.

Coluna	Justificativa Operacional e Mapeamento de Dados	Supporte nos Fontes
<b>Ride ID</b> (ContrSectionRideMkt.Ride)	Garante que cada negociação está ligada inequivocamente a um serviço específico [Query	O <code>Ride ID</code> é fundamental para auditar o ciclo de vida da corrida e as transações financeiras.

Coluna	Justificativa Operacional e Mapeamento de Dados	Suporte nos Fontes
<b>Contract Price</b> (ContrSectionRideMkt.TotalPrice)	<p>Table]. O modelo <b>ContrSectionRideMkt</b> representa o mercado de serviços, associando o preço final calculado a uma Ride.</p> <p>Este preço é o valor que o sistema calcula automaticamente com base nos contratos ativos (<b>contractsection</b> e suas variantes) [Query Table, 29, 606]. O <i>backoffice</i> usa-o como referência base para a negociação.</p>	<p>O preço total (<b>TotalPrice</b>) e os preços por secção de contrato (<b>ContractPriceA</b> a F) são armazenados no modelo <b>ContrSectionRideMkt</b>.</p>
<b>Supplier Quote (Manual entry) / Variance</b>	<p>Permite registar a proposta manual do Fornecedor e calcular a <b>diferença</b> entre o preço contratual automático e o preço negociado [Query Table]. Esta variação é um <b>alerta de limite</b> que requer atenção administrativa e documentação para auditoria [Query Table].</p>	<p>O <i>backoffice</i> tem de lidar com a gestão contratual, que define as taxas, acionadores de faturação e penalidades.</p>
<b>Approval Status (Dropdown: Pending/Approved/Rejected)</b>	<p>Controla o <b>fluxo de trabalho</b> (<i>Workflow control</i>) [Query Table]. Uma cotação manual que exceda a variação (Variance) deve ser aprovada ou rejeitada para garantir que a Suroga API efetua o pagamento correto ao Fornecedor.</p>	<p>A validação de dados e o cumprimento das regras de negócio são funções primárias do <i>backoffice</i> ao usar as listas SharePoint.</p>

O *Price Hub* permite, portanto, que o *backoffice* execute a **supervisão financeira** e a **documentação de exceções**, transformando um processo automático de preços num processo auditável quando há desvios (negociação), o que é crucial para as relações B2B com os Fornecedores e para o cumprimento fiscal.

## 7. Messaging Monitor (Communication Oversight)

O *Messaging Monitor* permite à equipa de *backoffice* **supervisionar a comunicação em tempo real**, o que é essencial para medir os KPIs de serviço ao cliente (com um alvo de <15 minutos de resposta) [Query Table] e **reduzir riscos legais através da mascaragem automática de PII** (Informação de Identificação Pessoal) [Query Table].

A Suroga facilita a comunicação contínua através do seu sistema de mensagens, que é gerido pelo modelo **MessageBoard**.

Coluna	Justificativa Operacional e Mapeamento de Dados	Supporte nos Fontes
<b>Message ID</b> (MessageBoard.pk)	Essencial para rastrear e auditar cada comunicação [Query Table]. O modelo MessageBoard armazena o corpo da mensagem (Text), o estado de leitura (Read), e se a mensagem foi marcada como abusiva (Abuse) ou apagada (deleted).	O MessageBoard é o modelo de base para a comunicação dentro do <i>rides</i> app.
<b>Sentiment Score</b>	O <i>backoffice</i> utiliza esta métrica (baseada em palavras-chave como "urgent") para <b>etiquetar a prioridade</b> (Priority tagging) [Query Table]. Mensagens prioritárias podem exigir intervenção imediata para manter a satisfação do cliente ou resolver problemas.	O sistema de mensagens permite aos utilizadores receberem atualizações e enviarem mensagens, sendo que o <i>backoffice</i> deve monitorizar estas interações.
<b>Response Time</b>	Calcula o tempo de resposta [Query Table], um KPI crítico de SLA (Acordo de Nível de Serviço) [Query Table]. A monitorização deste tempo é crucial para <b>melhorar o suporte e o desempenho operacional</b> .	O tempo de resposta está ligado à data em que a mensagem é lida (Read) versus a data de envio (Date).
<b>PII Mask</b>	A fórmula visa <b>mascarar PII</b> (como números de telefone) [Query Table], um requisito fundamental para a <b>conformidade com o GDPR</b> [Query Table]. Embora os Fornecedores retenham total autonomia sobre a gestão dos seus <i>workers</i> , a Suroga tem obrigações de privacidade e auditoria (e.g., registo de quem criou/aderiu a chamadas, <i>timestamps</i> e IPs).	É fundamental evitar armazenar informações privadas ou sensíveis como metadados. A fórmula reflete o esforço de cumprir as regras de proteção de dados.

## 8. Performance Analytics (KPI Dashboards)

O *Performance Analytics* é o dashboard de KPIs utilizado pelo *backoffice* para avaliar o desempenho do ecossistema Suroga, dos serviços individuais aos Fornecedores, cumprindo o objetivo de **identificar os Fornecedores com melhor desempenho para atribuição de bónus** [Query Table] e **identificar as causas-raiz de falhas no serviço** [Query Table].

Métrica	Justificativa Operacional e Mapeamento de Dados	Supporte nos Fontes
<b>On-Time Rate</b>	Mede a taxa de serviços concluídos a tempo, ajudando a <b>identificar falhas no serviço</b> [Query Table]. As <i>Rides</i> (corridas) têm datas de início ( <i>RideStartDate</i> ) e fim ( <i>RideEndDate</i> ) que marcam o ciclo de vida do serviço.	O sistema Django permite o rastreio do ciclo de vida do serviço, desde o registo até ao fecho ( <i>Closed</i> status).
<b>Revenue/Supplier</b>	Calcula a receita gerada por Fornecedor (Agência) [Query Table]. Esta métrica é essencial para <b>verificar a fiabilidade dos pagamentos</b> e para gerir o fluxo de dinheiro, uma vez que a Suroga API <b>paga</b> (payments[value]) presentes aos Fornecedores por serviço concluído (taxa variável baseada no contrato).	O cálculo depende dos registos de pagamento no modelo payment log.
<b>Stream</b>	Calculado com base na média das classificações dos	A classificação do cliente

Métrica	Justificativa Operacional e Mapeamento de Dados	Supporte nos Fontes
<b>Quality</b>	clientes ( <i>clientrank</i> ) [Query Table]. A Suroga valoriza o <i>feedback</i> e a classificação para controlo de qualidade.	(ClientRank) é armazenada no modelo Booking, permitindo a avaliação da qualidade do serviço (e.g., qualidade do <i>stream</i> ).

O *backoffice* da Suroga utiliza estas análises para realizar o **controlo de qualidade** e otimizar o desempenho, garantindo que os Fornecedores estão a cumprir os SLAs e os clientes estão satisfeitos. Esta gestão proativa é crucial dado que a **confiança e a fiabilidade de pagamento** são preocupações primárias para os Fornecedores.

## 9. Invoicing (Billing Engine): Conformidade Fiscal e Contratual

O *workbook* de Invoicing é vital para a **gestão B2B (Business-to-Business)** e para o **cumprimento integral do IVA (VAT Handling)** [Query Table], especialmente dada a complexidade das regras fiscais na União Europeia e em Portugal.

Coluna	Justificativa para Acesso do Backoffice	Supporte nos Fontes
<b>Invoice #</b> ( <i>Invoice.pk</i> )	Garante a criação de um identificador único de fatura [Query Table]. Este ID é crucial para a rastreabilidade financeira e para mapear o valor total devido ( <b>Total Due</b> ) com o <b>Value</b> registado no modelo de log de pagamentos ( <b>payment log</b> ).	
<b>Supplier ID</b> ( <i>contract.Supplier</i> )	Permite a validação de que a fatura está ligada ao Fornecedor correto. A Suroga <b>contrata diretamente</b> os Fornecedores para os serviços, e a gestão dos contratos e Fornecedores é uma função chave do <i>backoffice</i> .	O campo <b>Supplier</b> no modelo <b>contract</b> rastreia o Fornecedor vinculado.
<b>Fee Type / VAT %</b> (Reverse charge flag, MOSS rules)	Esta é a justificação central para conformidade. A Suroga API é responsável por aplicar o <b>mecanismo de reverse-charge (inversão do sujeito passivo) para transações B2B intra-UE</b> . A fórmula <code>=IF( [@[fee type]]="EU B2B", 0%, ... )</code> reflete a aplicação destas regras fiscais, garantindo 0% de IVA quando apropriado, alinhado com as regras MOSS (Mini One-Stop Shop).	O Stripe automatiza o cálculo do IVA, mas a Suroga deve garantir a conformidade (especialmente para SAF-T em Portugal, que requer integração com software certificado).
<b>Total Due</b> ( <i>payment log.Value</i> )	O valor total devido é a base para o registo no <b>payment log</b> . O <i>backoffice</i> precisa de verificar se a fatura final, após a aplicação de impostos (Subtotal * (1 + VAT %)), corresponde ao que será registrado como transação monetária.	O modelo <b>payment log</b> regista o <b>Value</b> (valor) da transação.

## 10. Supplier Payments (Payout Management): Gestão de Payouts e Risco

Este *workbook* monitoriza o processo de pagamento efetivo aos Fornecedores, gerindo as deduções (comissões) e implementando mecanismos de mitigação de risco e otimização de custos [Query Table].

Coluna	Justificativa para Acesso do Backoffice	Supporte nos Fontes
<b>Net Amount (Fee calc)</b>	A Suroga API paga aos Fornecedores por serviço concluído, mas também paga comissões à Greenstreaming (por sessão) e Greenconnections (por recrutamento). A fórmula $((1 - [@[\text{greenstreaming \%}]]) - [@[\text{greenconnections \%}]]))$ permite ao backoffice verificar se as <b>comissões</b> pagas a estas entidades ( <i>Greenstreaming</i> e <i>Greenconnections</i> ) foram deduzidas corretamente do valor bruto antes do pagamento ao Fornecedor.	A Suroga API paga à Greenstreaming uma comissão por sessão.
<b>Reserve Hold (Risk buffer)</b>	Esta coluna é crítica para a gestão de risco. Implementa o <b>retentor de reserva de 5%</b> ( $[@[\text{net}]] * 0.05$ ) que é retido sobre os fundos disputados ( $\text{IF}([@[\text{disputed}]], \dots)$ ) [Query Table]. Este mecanismo é uma cláusula contratual: os <i>Suppliers</i> aceitam um <b>retenção de 5% por 30 dias para cobertura de disputas</b> ( <i>dispute reserve</i> ).	O <i>backoffice</i> monitoriza os pagamentos para garantir a mitigação de risco e fiabilidade.
<b>Payable (€100 threshold)</b>	Garante que o pagamento líquido atinge o <b>limite mínimo de 100€</b> ( $\text{MAX}(0, [\dots] - 100)$ ) [Query Table]. Este limite é um requisito operacional para otimizar os custos de transação e alinha-se com o facto de os Parceiros serem pagos via Stripe Connect com um <b>limite mínimo de 100€</b> .	
<b>Method (Cost optimization)</b>	Esta lógica de seleção de método reflete a <b>otimização de custos</b> [Query Table]. A Suroga usa o <b>Stripe como infraestrutura primária</b> para pagamentos (SEPA a 0,25€ por payout), mas o <b>PayPal é uma opção suplementar</b> . Ao usar PayPal para montantes inferiores a 100€, a Suroga pode estar a tentar <b>reduzir custos</b> ou gerir a complexidade de diferentes payouts.	

Em suma, estes *workbooks* proporcionam ao *backoffice* visibilidade total e controlo manual sobre os **cálculos de conformidade fiscal e a gestão de exceções financeiras** (reservas de disputa, limites mínimos), elementos que, embora automatizados pelo Stripe, exigem supervisão humana rigorosa para manter a legalidade operacional e a confiança dos Fornecedores.