

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 compreensíveis 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	Suporte nos Fontes
Active Rides (=COUNTIF('ride_registry':0:0, 'active'))	Permite o acompanhamento imediato dos serviços de <i>streaming</i> em andamento. A gestão do ciclo de vida das <i>Rides</i> (corridas) deve ser eficiente e eficaz.	As <i>Rides</i> (corridas) no sistema Suroga possuem diversos <i>status</i> , como Blocked , Canceled , Deleted e Closed , que são vitais para o fluxo de trabalho.
Unpaid Invoices (=COUNTIF('invoicing':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 Invoicing & Billing (<i>Faturação e Cobrança</i>) rastreia o <i>status</i> das faturas (e.g., "Issued" - Emitida, "Overdue" - Vencida). A fiabilidade do pagamento é uma preocupação fundamental para os Fornecedores.
Avg. Response Time (=AVERAGE('messaging':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 MessageBoard é usado para gerir as comunicações.
Supplier Utilization (='stock	Garante que os Fornecedores (Agências) disponíveis e o <i>stock</i> de serviços sejam	O sistema precisa gerir ativamente o <i>stock</i> de serviços para publicidade à Suroo APP e rastrear a utilização do <i>stock</i>

Métrica	Importância para o Backoffice	Suporte nos Fontes
<code>scheduler '!B5)</code>	alocados de forma eficiente. A Suroga API gere Fornecedores e utiliza filtragem avançada para fazer a correspondência com a procura de serviços.	(<code>ContrSectionRideMkt</code> modelo de mercado de serviços).
VAT Liability (<code>=SUM('invoicing'!I:I)</code>)	Essencial para o cumprimento fiscal . 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 software certificado por terceiros 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
Ride ID (<code>= "RIDE - "&TEXT(ROW(), "00000")</code>)	Assegura que cada serviço tenha um identificador único, crucial para o sistema. O seu rastreamento ajuda a prevenir dupla-marcação (<i>double-booking</i>) [Query Table].	Mapeado para <code>Ride.pk</code> (Chave Primária do modelo Ride) [Query Table, 229].
Requisits (JSON)	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.
Status (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 <i>status</i> no modelo Ride (e.g., <code>Ride.Blocked</code> , <code>Ride.Canceled</code> , <code>Ride.Closed</code>).
Time-to-Confirm	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 Booking.

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

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	Suporte nos Fontes
Service ID / Name (<i>servicetype.id</i> / <i>servicetype.Name</i>)	O <i>servicetype</i> (tipo de serviço) é a base para categorizar as ofertas de serviços na plataforma. O <i>backoffice</i> gere o <i>stock</i> de serviços e as suas configurações (equipamento, regras, duração) a partir daqui.	O modelo <i>servicetype</i> define nome, duração, equipamento e regras de serviço (em JSON).
Min Duration (<i>servicetype.Duration</i>)	Define a duração mínima do serviço que a Suroga API publicita à Suroo App. Isto está ligado ao modelo <i>RideDuration</i> (Duração da Corrida), assegurando que apenas durações predefinidas sejam utilizadas.	A duração (<i>Duration</i>) é um campo de chave estrangeira (<i>ForeignKey</i>) no modelo <i>servicetype</i> ligando ao modelo <i>RideDuration</i> .
Required Equipment (<i>servicetype.Equipment</i>)	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 <i>servicetype</i> inclui um campo <i>Equipment</i> (equipamento) que é uma <i>ForeignKey</i> para o modelo <i>equipment</i> .
NUTS Codes	Permite que o <i>backoffice</i>	O modelo <i>contractsection</i>

Parâmetro	Importância para o Backoffice	Suporte nos Fontes
(contractsection.nuts0-3)	defina com precisão a cobertura territorial 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 cobertura territorial via NUTS .
Base Price (€) / Peak Surcharge (contractsection.variante)	Estas colunas controlam a estrutura de preços . 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 variante (formato JSON) do modelo contractsection . 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 price 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	Suporte nos Fontes
Payment ID (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 StripTransactionID e SurooBankTransactionID .	Mapeia para a chave primária (pk) do modelo payment log , que registra as transações de pagamento.
Ride Link (payment log.Ride)	Associa inequivocamente o pagamento a uma Booking (Reserva) específica. Isto é essencial para conciliar os pagamentos (e.g., da Suroo App à Suroga API, ou da Suroga API ao Fornecedor) com o serviço prestado.	O payment log possui uma chave estrangeira Ride (que se refere a um objeto Booking).

Coluna	Importância para o Backoffice	Suporte nos Fontes
Amount Due (payment log .Value)	O campo <code>Value</code> no <code>payment log</code> 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> (<code>xlookup</code>) recupera o valor da fatura do "Ride Registry", garantindo que o valor registado no <code>payment log .Value</code> está correto [Query Table, 616].
Status (payment log .OperationType)	Permite ao <i>backoffice</i> monitorizar o estado de todos os fluxos de caixa. O <code>payment log</code> regista o <code>OperationType</code> (Tipo de Operação), que inclui o estado do pagamento.	O campo <code>OperationType</code> no <code>payment log</code> define o tipo de transação (e.g., "Issued," "Paid," "Disputed"). O modelo <code>country</code> armazena o campo <code>VAT</code> (IVA), com um valor padrão de 23.0% .
VAT % (Tax compliance)	Esta fórmula complexa (incluindo a taxa de IVA Portuguesa de 23% para PT e consultas <code>VLOOKUP</code> 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.	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.
Days Overdue (payment log .SettlementDate)	O registo da data de liquidação (<code>SettlementDate</code>) e a subsequente verificação de dias em atraso são essenciais para manter a fiabilidade dos pagamentos aos Fornecedores, que é uma das suas principais preocupações.	A coluna usa o <code>SettlementDate</code> do <code>payment log</code> 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	Suporte nos Fontes
Supplier (<code>profilerepresentative.Name</code>)	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.
NUTS Region (<code>contractsection.nuts3</code>)	Essencial para a definição precisa da área de serviço 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> .
Capacity (<code>SectonStock.capacity</code>)	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 gere ativamente o <i>stock</i> de serviços. O campo <code>SectonStock</code> na secção do contrato armazena estes detalhes.
Booked % / Conflict Alert	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 eficiência e a escalabilidade do agendamento.	O sistema Django automatiza o agendamento (<code>Booking viewsets</code>) e a gestão de <i>stock</i> (<code>ContrSectionRideMkt</code> 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 *taxa variável baseada em contrato*.

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

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

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 mascaramento 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	Suporte nos Fontes
Message ID (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.
Sentiment Score	O <i>backoffice</i> utiliza esta métrica (baseada em palavras-chave como "urgent") para etiquetar a prioridade (<i>Priority tagging</i>) [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.
Response Time	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 melhorar o suporte e o desempenho operacional .	O tempo de resposta está ligado à data em que a mensagem é lida (Read) versus a data de envio (Date).
PII Mask	A fórmula visa maskarar PII (como números de telefone) [Query Table], um requisito fundamental para a conformidade com o GDPR [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	Suporte nos Fontes
On-Time Rate	Mede a taxa de serviços concluídos a tempo, ajudando a identificar falhas no serviço [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 status</i>).
Revenue/Supplier	Calcula a receita gerada por Fornecedor (Agência) [Query Table]. Esta métrica é essencial para verificar a fiabilidade dos pagamentos e para gerir o fluxo de dinheiro, uma vez que a Suroga API paga aos Fornecedores por serviço concluído (taxa variável baseada no contrato).	O cálculo depende dos registos de pagamento (<code>payments[value]</code>) presentes no modelo <code>payment log</code> .
Stream	Calculado com base na média das classificações dos	A classificação do cliente

Métrica	Justificativa Operacional e Mapeamento de Dados	Suporte nos Fontes
Quality	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 <i>backoffice</i> 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	Suporte nos Fontes
Invoice # (Invoice.pk)	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 (Total Due) com o Value registado no modelo de log de pagamentos (payment log).	
Supplier ID (contract.Supplier)	Permite a validação de que a fatura está ligada ao Fornecedor correto. A Suroga contrata diretamente os Fornecedores para os serviços, e a gestão dos contratos e Fornecedores é uma função chave do <i>backoffice</i> .	O campo Supplier no modelo contract rastreia o Fornecedor vinculado.
Fee Type / VAT % (Reverse charge flag, MOSS rules)	Esta é a justificação central para conformidade. A Suroga API é responsável por aplicar o mecanismo de reverse-charge (inversão do sujeito passivo) para transações B2B intra-UE . A fórmula =IF([@[fee type]]="EU B2B"0%, . . .) 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 <i>software</i> certificado).
Total Due (payment log.Value)	O valor total devido é a base para o registo no payment log. O <i>backoffice</i> precisa de verificar se a fatura final, após a aplicação de impostos (Subtotal * (1 + VAT %)), corresponde ao que será registado como transação monetária.	O modelo payment log regista o Value (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	Suporte nos Fontes
Net Amount (Fee calc)	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 <i>backoffice</i> verificar se as comissões 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.
Reserve Hold (Risk buffer)	Esta coluna é crítica para a gestão de risco. Implementa o retentor de reserva de 5% $([\text{net}] * 0.05)$ 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 retenção de 5% por 30 dias para cobertura de disputas (<i>dispute reserve</i>).	O <i>backoffice</i> monitoriza os pagamentos para garantir a mitigação de risco e fiabilidade.
Payable (€100 threshold)	Garante que o pagamento líquido atinge o limite mínimo de 100€ $(\text{MAX}(0 [\dots] - 100))$ [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 limite mínimo de 100€ .	
Method (Cost optimization)	Esta lógica de seleção de método reflete a otimização de custos [Query Table]. A Suroga usa o Stripe como infraestrutura primária para pagamentos (SEPA a 0,25€ por <i>payout</i>), mas o PayPal é uma opção suplementar . Ao usar PayPal para montantes inferiores a 100€, a Suroga pode estar a tentar reduzir custos ou gerir a complexidade de diferentes <i>payouts</i> .	

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.