

Levvā

Título do documento		
/ tipo	/ área responsável	/ início da vigência
Documentação de negócios	Produtos	Junho/2025

Sumário

Objetivo.....	2
Documentos referenciados	2
Abrangência	2
Considerações gerais	3
Considerações especiais (contextualização)	4
Regras de Negócio	5
1. Validação de sessão do Usuário.....	5
2. Consulta de Taxas.....	7
3. Consulta de Cidades.....	22
4. Tabelas Auxiliares	40
5. Considerações Finais.....	42



Objetivo

Este documento tem como objetivo detalhar as funcionalidades da API CROSS para consulta de taxas e cidades, permitindo a aplicação de múltiplos filtros combináveis, como nome, classe, país e tipo de transporte. Essa abordagem visa facilitar a criação de propostas comerciais e análises logísticas, assegurando a segurança, precisão e consistência dos dados.

Documentos referenciados

1. Documentação_Técnica_POC_QMS_Craft_Levva.docx

Abrangência

Este documento funcional aplica-se a todos os profissionais envolvidos na utilização da API CROSS para consulta de taxas e cidades, com foco na aplicação de múltiplos filtros combináveis. Abaixo estão os principais perfis de usuários elegíveis:

- **Equipe de Desenvolvimento:** responsável pela implementação e manutenção da API, garantindo que os filtros funcionem corretamente e atendam às necessidades dos usuários.
- **Analistas de Negócios:** utilizam a API para obter dados precisos e atualizados.

Este documento também serve como referência para futuras atualizações e melhorias na API, assegurando que todos os envolvidos estejam alinhados quanto às funcionalidades e objetivos do sistema.



Considerações gerais

Esta funcionalidade foi desenvolvida para facilitar e agilizar o trabalho dos usuários ao consultar taxas e cidades. Ela permite aplicar múltiplos filtros combináveis, como nome, classe, país e tipo de transporte, tornando o processo mais rápido e preciso.

Ao combinar filtros inteligentes, validação de sessão e padronização de dados, conseguimos:

- **Reduzir o tempo necessário para compor propostas comerciais**, pois as informações são acessadas de forma mais eficiente.
- **Minimizar erros manuais**, já que os dados são consistentes e confiáveis.
- **Melhorar a eficiência operacional em processos logísticos complexos**, pois as informações estão organizadas e atualizadas.

Além disso, o tratamento flexível de parâmetros permite que o sistema se adapte a diferentes contextos de negócio, como análise de tarifários, planejamento logístico e gestão de ofertas comerciais, sem comprometer a segurança e a integridade dos dados.

Essa abordagem também melhora a navegabilidade das interfaces e oferece uma experiência de uso mais fluida, confiável e responsiva, tornando o sistema mais adaptável às necessidades dos usuários.



Considerações especiais (contextualização)

- Definição de parâmetro

Um parâmetro é uma informação que você fornece para que o sistema possa realizar uma tarefa específica. Por exemplo, se você quer encontrar todas as cidades que começam com a letra "S", você pode passar "S" como parâmetro. O sistema então usará esse parâmetro para filtrar os resultados e mostrar apenas as cidades que atendem a esse critério.

- Definição de Tabela

Uma **tabela** é como uma planilha organizada dentro de um banco de dados. Ela é composta por linhas e colunas:

- **Linhas** (ou registros): cada linha representa uma entrada única de dados.
- **Colunas** (ou campos): cada coluna representa um tipo específico de informação, como nome, idade ou endereço.

Por exemplo, em uma tabela de clientes, cada linha pode representar um cliente diferente, e as colunas podem incluir nome, telefone e e-mail. Isso permite armazenar e organizar grandes volumes de informações de forma estruturada e eficiente.

Regras de Negócio

1. Validação de sessão do Usuário

- A sessão do usuário deve ser validada em todos os endpoints, exceto aqueles explicitamente marcados como exceções;
- Sessões inválidas devem retornar uma lista vazia ou um erro apropriado;
- Todos os endpoints que recebem o parâmetro **userSession** devem validar a sessão do usuário na tabela **CSAG311**. Sessões inválidas devem retornar erro ou resultado vazio;
- Verificação de Resultado
 - Se a consulta retornar algum registro, a sessão é considerada válida;
- Continuação do Processamento
 - Em caso de sessão válida, o endpoint prossegue com a lógica de negócio específica.
Se a consulta não retornar registros, a sessão é considerada inválida.
- Tratamento de Sessão Inválida
 - Em caso de sessão inválida, o endpoint retorna uma lista vazia ou um erro apropriado, onde nenhuma operação adicional é executada.





- Estrutura da Tabela CSAG311 (Sessões de Usuário)

Campo	Tipo	Descrição	Chave	Relacionamentos
UsuarioSessao	VARCHAR(50)	Identificador único da sessão	PK	-
ID_USUARIO	VARCHAR(20)	Código do usuário associado à sessão	FK	CSAG310.ID_USUARIO
DT_INICIO	DATETIME	Data e hora de início da sessão	-	-
DT_ULTIMO_ACESSO	DATETIME	Data e hora do último acesso	-	-
IP_ORIGEM	VARCHAR(15)	Endereço IP de origem da sessão	-	-
ATIVO	CHAR(1)	Indicador se a sessão está ativa ('S'/'N')	-	-



2. Consulta de Taxas

- O usuário deve ser capaz de consultar taxas por classe, nome, proposta ou combinação de filtros;
- Apenas taxas ativas devem ser retornadas;
- As taxas devem ser ordenadas alfabeticamente pelo nome;
- Campos retornados devem incluir código, nome e classes associadas;
- O usuário pode listar todas as taxas disponíveis ou buscar taxas por classe e nome;
- É possível obter taxas associadas a propostas ou filtrar por classe e nome simultaneamente;
- As taxas devem incluir informações como código, nome e classes associadas.

2.1 Endpoints de Taxas

Para cada uma das ações mencionadas anteriormente, deverá ser implementado um endpoint correspondente, responsável por executar a funcionalidade desejada:

2.1.1 GET api/Tax/selected-classes

Este endpoint é responsável por validar a sessão do usuário seguindo o processo padrão descrito anteriormente. Em caso de sessão inválida, o sistema deve retornar uma lista vazia.



- Validação: O parâmetro **taxId** deve corresponder a um código de taxa válido existente na tabela **HCGS3001**.

6. Caso o **taxId** informado não exista na tabela, o endpoint deverá retornar uma lista vazia.

- Filtragem: O sistema procura somente as classes que estão ligadas à taxa informada.

Essa ligação é feita com base no campo **CLASSE_TAXA**, que fica na tabela **HCGS3001**.

O campo **CLASSE_TAXA** guarda os códigos das classes em uma única sequência de texto (por exemplo: "OFD").

O sistema separa essa sequência em partes e, para cada código, busca as informações completas na tabela **CCGS221**.

- Ordenação: As classes são exibidas na mesma ordem em que estão cadastradas na tabela **CCGS221**.

O sistema não aplica nenhuma ordenação extra; ele apenas mantém a ordem original da tabela.

- Estrutura da Tabela HCGS3001 (Taxas)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_TAXA	VARCHAR(10)	Código único da taxa	PK	HCGS3006.ID_TAXA
NM_TAXA	VARCHAR(100)	Nome/descrição da taxa	-	-
CLASSE_TAXA	VARCHAR(10)	Classes associadas (concatenadas)	-	Referência lógica a CCGS221.CLASSE
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-

ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se a taxa está ativa ('S'/'N')	-	-

- Estrutura da Tabela CCGS221 (Classes de Taxa)

Campo	Tipo	Descrição	Chave	Relacionamentos
CLASSE	CHAR(1)	Código da classe ('O', 'F', 'D')	PK	Referenciado logicamente por HCGS3001.CLASSE_T AXA
DESCRICAO	VARCHAR(50)	Descrição da classe ('Origem', 'Frete', 'Destino')	-	-
ORDEM	INT	Ordem de exibição/ processamento	-	-
ATIVO	CHAR(1)	Indicador se a classe está ativa ('S'/'N')	-	-

2.1.2 GET api/Tax/classes

Este endpoint tem como objetivo fornecer uma lista fixa com as classes básicas de taxa do sistema ('O' (Origem), 'F' (Frete) e 'D' (Destino)), sem necessidade de consulta ao banco de dados, já que essas categorias são pré-definidas e codificadas diretamente na aplicação.

- Sem Validação: Este endpoint não exige validação de sessão porque retorna uma lista fixa de dados gerais. Como as informações



não são sensíveis nem personalizadas, o acesso pode ser feito por qualquer usuário, mesmo sem estar autenticado.

- Retorno: Cada item retornado pelo endpoint traz três campos:
 - Id: Representa o código da classe, sendo um dos seguintes valores fixos:
'O' (Origem), 'F' (Frete) ou 'D' (Destino);
 - Value: É o nome completo da classe, apresentado de forma mais legível para o usuário ('Origem', 'Frete', 'Destino');
 - Classe: Repete o valor do campo Id.

Este campo é redundante, mas está presente por compatibilidade com outras partes do sistema;
- Interfaces: Este endpoint é usado principalmente para preencher menus de seleção (como combos ou filtros) nas telas do sistema. Ele fornece uma lista padronizada de categorias de taxas, garantindo que os usuários sempre vejam a mesmas opções em todas as interfaces.

Manter essa lista consistente é essencial para que o sistema classifique corretamente as taxas e funcione de forma confiável.

2.1.3 GET api/Tax/list

Antes de responder à solicitação, o endpoint verifica se a sessão do usuário está válida, seguindo o processo padrão do sistema.

Se a sessão não for válida ou estiver expirada, o sistema não retorna nenhum dado e responde com uma **lista vazia**. Isso garante que **apenas usuários autenticados** tenham acesso às informações disponíveis nesse endpoint.



- **Listagem:** Este endpoint traz todas as taxas cadastradas na tabela chamada **HCGS3001**. Ele não aplica nenhum **filtro extra** além de mostrar somente as taxas que estão **ativas**.
Para isso, só aparecem na lista as taxas que têm o campo **ATIVO** marcado como '**S**'. Ou seja, taxas desativadas ou inativas **não serão mostradas**.
- **Informações Retornados:** Este endpoint retorna somente duas informações de cada taxa: Código da **taxa (ID_TAXA)** e **Nome da taxa (NM_TAXA)**.
Outros dados, como as classes associadas à taxa, não são incluídos na resposta deste endpoint.
- **Ordenação:** As taxas retornadas por este endpoint são organizadas em ordem alfabética, com base no nome da **taxa (NM_TAXA)**.
Essa ordenação ajuda o usuário a encontrar mais facilmente uma taxa específica, especialmente quando há muitas opções na lista.
- **Interfaces:** Este endpoint é usado principalmente para alimentar listas suspensas (combos de seleção) em telas do sistema.
Ele serve como a fonte principal para exibir todas as taxas disponíveis, permitindo que o usuário escolha uma opção de forma rápida e prática.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.



2.1.4 GET api/Tax/all-with-classes

Antes de retornar qualquer dado, o endpoint verifica se o usuário está autenticado corretamente, seguindo as regras padrão do sistema.

Se a sessão for inválida (por exemplo, expirada ou incorreta), nenhuma informação é exibida, o sistema simplesmente retorna uma lista vazia.

- **Consolidação de Dados:** Para cada taxa registrada na tabela **HCGS3001**, o sistema procura as classes associadas a ela. Essas classes estão armazenadas em um único campo chamado **CLASSE_TAXA**, como uma string com vários códigos juntos (por exemplo: "OFD" representa Origem, Frete e Destino). No retorno da API, essa informação é enviada **exatamente** como está na tabela, ou seja, como uma **única** string de códigos, sem detalhar o que cada código significa.
- **Filtragem de Taxas Ativas:** Somente as taxas que estão ativas (ou seja, com o campo ATIVO igual a 'S') são exibidas nos resultados. As taxas que estão inativas são ignoradas e não aparecem na lista retornada pela API.
- **Campos Retornados:** Cada taxa retornada pela API contém as seguintes informações:
 - **Id:** é o código identificador da taxa (vem do campo **ID_TAXA**).
 - **Value:** é o nome da taxa (vem do campo **NM_TAXA**).



- Classe: são os códigos das classes associadas à taxa, todos juntos em uma única sequência de texto (do campo **CLASSE_TAXA**).
- Ordenação: As taxas são organizadas em ordem alfabética, com base no nome da taxa (campo **NM_TAXA**).
Isso facilita para o usuário encontrar rapidamente a taxa desejada em listas maiores.
- Análises: Este endpoint é usado principalmente em análises e relatórios, ajudando a entender como as taxas estão distribuídas por classe.
Com ele, é possível identificar de forma rápida quais taxas se aplicam a cada etapa do processo logístico, como origem, frete e destino.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.

2.1.5 GET api/Tax/search/by-class

Antes de retornar qualquer informação, o endpoint verifica se a sessão do usuário é válida, seguindo as regras padrão do sistema. Se a sessão for inválida (por exemplo, expirada ou inexistente),



nenhuma informação será exibida — o sistema retorna uma lista vazia.

- **Filtragem por classes:** Se o parâmetro **classId** (código da classe) for informado, o sistema irá filtrar as taxas que contêm essa classe.
Essa verificação é feita buscando o valor dentro do campo **CLASSE_TAXA**, mesmo que ele esteja no início, meio ou fim da string concatenada (por isso usamos o comando LIKE '%classId%').
Se o **classId** não for informado, o sistema simplesmente retorna todas as taxas que estão ativas, sem aplicar esse filtro.
- **Filtragem de Taxas Ativas:** Somente as taxas que estão ativas (com o campo **ATIVO** igual a 'S') são exibidas no resultado. **Taxas inativas** (com outro valor no campo ATIVO) não aparecem na listagem, elas são ignoradas completamente pelo sistema.
- **Campos Retornados:** Cada item retornado pelo endpoint traz três informações principais:
 - **Id:** é o código identificador da taxa, correspondente ao campo **ID_TAXA**.
 - **Value:** é o nome da taxa, vindo do campo **NM_TAXA**.
 - **Classe:** mostra as classes associadas à taxa, apresentadas como uma única string com os códigos das classes concatenados (campo **CLASSE_TAXA**), como por exemplo: "OFD".
- **Ordenação:** Os resultados retornados por este endpoint são organizados em ordem alfabética, com base no nome da taxa (campo **NM_TAXA**).
Essa ordenação facilita a visualização e a localização das taxas, especialmente quando a lista é extensa.
- **Interfaces de Filtro:** Este endpoint serve principalmente para ajudar os usuários a encontrarem taxas específicas, filtrando-as por classe.
Ele é usado em telas e sistemas onde o usuário precisa escolher

taxas relacionadas a uma etapa específica do processo logístico, tornando a busca mais rápida e fácil.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.

2.1.6 GET api/Tax/search/by-name

Antes de mostrar qualquer informação, o endpoint verifica se a sessão do usuário está válida, seguindo o processo padrão.

Se a sessão não estiver válida, ele não retorna dados e entrega uma lista vazia.

- **Busca por prefixo:** O sistema procura taxas cujo nome (**NM_TAXA**) comece com as letras que você digitar no parâmetro chamado "**initial**".
Essa busca funciona como se estivesse procurando palavras que começam exatamente com esse prefixo.
Além disso, a busca não se importa se as letras são maiúsculas ou minúsculas, elas são tratadas da mesma forma.
- **Uso de Query Armazenada:** Este endpoint usa uma consulta SQL que já está salva na tabela chamada **CSAG367_FVT**.
Essa consulta tem o nome '**WS_HCGS3001_BUSCAR_TAXAS**'.
Usar consultas salvas assim ajuda a organizar melhor o código e facilita reutilizar consultas que são mais complexas, sem precisar escrever tudo de novo.
- **Filtragem de Taxas Ativas:** O sistema mostra somente as taxas que estão ativas, ou seja, aquelas com o campo **ATIVO = 'S'**.



As taxas que não estão ativas não aparecem na lista, elas são totalmente ignoradas.

- Campos Retornados: Cada taxa retornada contém:
 - Id: código que identifica a taxa (campo **ID_TAXA**);
 - Value: nome da taxa (campo **NM_TAXA**);
 - Value2: este campo não é usado neste endpoint e sempre retorna **null** (vazio).
- Ordenação: Os resultados são organizados em ordem alfabética, usando o nome da taxa (**NM_TAXA**), para facilitar a busca.
- Busca Incremental: Este endpoint ajuda a fazer buscas que vão ficando mais precisas enquanto o usuário digita. Ou seja, conforme o usuário escreve, a lista de resultados vai se ajustando automaticamente para mostrar as opções que mais combinam com o que foi digitado até o momento.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.

- Estrutura da Tabela CSAG367_FVT (Queries SQL Armazenadas)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_QUERY	VARCHAR(50)	Identificador único da query	PK	-
DESCRICAO	VARCHAR(200)	Descrição da finalidade da query	-	-
SQL_TEXT	TEXT	Texto completo da query SQL	-	-
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO



ATIVO	CHAR(1)	Indicador se a query está ativa ('S'/'N')	-	-
-------	---------	---	---	---

2.1.7 GET api/Tax/search/by-class-filter

Este endpoint é diferente dos outros porque não precisa verificar se o usuário está logado. Por isso, ele pode ser usado por partes do sistema que funcionam mesmo sem uma sessão ativa.

- Filtragem por Classe: Se o parâmetro **classFilter** for informado, o sistema irá buscar apenas as taxas que contenham essa classe no campo **CLASSE_TAXA**.
Essa busca é feita usando uma verificação parcial, ou seja, procura qualquer ocorrência da classe dentro do valor (como um "contém").
- Por exemplo: se **classFilter = 'F'**, o sistema encontrará taxas com **CLASSE_TAXA = 'OFD'**, porque a letra F está presente.
Se o parâmetro não for informado, o sistema retorna todas as taxas ativas, sem aplicar esse filtro.
- Uso Condicional de Query Armazenada: O comportamento do endpoint muda dependendo do uso do parâmetro **classFilter**:
 - Se **classFilter** não for informado ou estiver vazio: O sistema executa uma consulta SQL simples, feita diretamente no código;
 - Se **classFilter** tiver algum valor: O sistema usa uma **query armazenada** no banco de dados (pré-definida), que é preparada para aplicar o filtro de classe corretamente.
- Filtragem de Taxas Ativas: O sistema só exibe as taxas que estão ativas, ou seja, aquelas com o campo **ATIVO** igual a 'S'.



Qualquer taxa marcada como inativa é automaticamente ignorada e não aparece na listagem de resultados. Isso garante que o usuário visualize apenas informações atualizadas e relevantes.

- **Campos Retornados:** Cada item retornado pelo endpoint contém as seguintes informações:
 - **Id:** é o código da taxa, representado pelo campo **ID_TAXA**;
 - **Value:** é o nome da taxa, vindo do campo **NM_TAXA**;
 - **Value2:** não é usado neste endpoint, por isso sempre retorna null;

Essa estrutura facilita o uso dos dados em componentes visuais, como listas suspensas ou campos de busca.

- **Ordenação:** Os resultados retornados por este endpoint são organizados em ordem alfabética, com base no nome da taxa (**campo NM_TAXA**). Essa ordenação facilita para o usuário encontrar a taxa desejada com mais rapidez, especialmente quando há uma grande quantidade de registros.
- **Uso em Processos Automatizados:** Este endpoint foi criado para ser usado principalmente por processos automáticos, como tarefas em segundo plano (background) ou jobs agendados. Além disso, ele também pode ser acessado por partes do sistema que não possuem uma sessão de usuário ativa, já que não exige autenticação.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.

Estrutura da Tabela CSAG367_FVT (Queries SQL Armazenadas)

A estrutura desta tabela já foi detalhada anteriormente.



2.1.8 GET api/Tax/for-proposal

- **Validação de Parâmetros:** O parâmetro **proposalId** deve conter o código de uma proposta válida que exista no sistema. Se esse código **não** for encontrado ou se a proposta não tiver nenhuma taxa associada, o endpoint vai retornar uma lista vazia.
- **Filtragem por Proposta e Classe:** O sistema procura as taxas que estão vinculadas à proposta informada, consultando a tabela **HCGS3006**. Se o parâmetro **classFilter** for preenchido, o sistema faz um filtro adicional para exibir apenas as taxas que contêm essa classe. Esse filtro é feito usando a expressão **LIKE '%classFilter%'** no campo **CLASSE_TAXA** da tabela **HCGS3001**, o que permite localizar a classe mesmo que ela esteja no meio da string.
- **Join de Tabelas:** O sistema cruza informações das tabelas **HCGS3001** (onde estão todas as taxas cadastradas) e **HCGS3006** (que relaciona as taxas às propostas). Essa junção é feita usando o campo **ID_TAXA**, que existe nas duas tabelas. Dessa forma, é possível recuperar ao mesmo tempo os dados gerais da taxa e a descrição específica dela dentro da proposta.
- **Campos Retornados:** Cada item retornado pelo endpoint contém:
 - **Id:** Código identificador da taxa (vem da tabela HCGS3001.ID_TAXA);
 - **Value:** Nome da taxa (vem da tabela HCGS3001.NM_TAXA);



- **Value2:** Descrição personalizada da taxa dentro da proposta (vem da tabela HCGS3006.DESCRICAO). Essas informações permitem exibir tanto os dados gerais da taxa quanto a forma como ela foi descrita para uma proposta específica.
- **Ordenação:** Os resultados são organizados em ordem alfabética, usando o nome da taxa (campo **NM_TAXA**). Isso facilita encontrar rapidamente a taxa desejada na lista.
- **Uso em Interfaces de Proposta:** Este endpoint é usado principalmente para mostrar e editar as taxas que fazem parte de uma proposta comercial. Com ele, é possível ver claramente quais taxas estão sendo aplicadas em cada fase do processo logístico.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.

- Estrutura da Tabela HCGS3006 (Taxas da Proposta)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_PROPOSTA	VARCHAR(10)	Código da proposta	PK, FK	HCGS3005.ID_PROPOSTA
ID_TAXA	VARCHAR(10)	Código da taxa	PK, FK	HCGS3001.ID_TAXA
DESCRICAO	VARCHAR(200)	Descrição específica da taxa na proposta	-	-
VALOR	DECIMAL(15,2)	Valor da taxa na proposta	-	-
MOEDA	CHAR(3)	Código da moeda do valor	FK	CSAG331.SIGLA
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
		Usuário que incluiu o registro		

ID_USUARIO	VARCHAR(20)		FK	CSAG310.ID_USUARIO
------------	-------------	--	----	--------------------



2.1.9 GET api/Tax/for-proposal

Este endpoint verifica se a sessão do usuário é válida seguindo o processo padrão.

Se a sessão não for válida, ele retorna uma lista vazia.

- Filtragem Combinada: O sistema pode aplicar dois filtros opcionais, que podem ser usados juntos:
- Filtro por nome: seleciona taxas cujo nome (**NM_TAXA**) começa com o texto passado no parâmetro initial. Isso é feito usando a busca **LIKE 'initial%'**.
- Filtro por classe: seleciona taxas que incluem a classe indicada pelo parâmetro **classId**. A busca usa **LIKE '%classId%'** no campo **CLASSE_TAXA**.
Se os dois filtros forem usados ao mesmo tempo, o sistema busca taxas que atendam a ambos os critérios (usando "E" lógico).
Se nenhum filtro for passado, o sistema retorna **todas** as taxas que estão ativas.
- Filtragem de Taxas Ativas: Somente as taxas que estão marcadas como ativas (com o campo **ATIVO = 'S'**) são incluídas no resultado da consulta.
Taxas inativas são ignoradas e não aparecem na listagem de forma alguma.
- Campos Retornados: Cada item retornado por este endpoint contém:
 - Id: o código identificador da taxa, vindo do campo **ID_TAXA**.
 - Value: o nome da taxa, obtido do campo **NM_TAXA**.



- Classe: uma string que reúne, em sequência, os códigos das classes associadas à taxa, conforme armazenado no campo **CLASSE_TAXA**.
- Ordenação: Os resultados retornados por este endpoint são organizados em ordem alfabética, com base no nome da taxa (campo **NM_TAXA**). Isso facilita para o usuário localizar uma taxa específica em listas longas, tornando a navegação mais rápida e intuitiva.
- Uso em Interfaces de Busca Avançada: Este endpoint foi criado para ser usado em interfaces com busca avançada. Ele permite que o usuário aplique mais de um filtro ao mesmo tempo, como nome e classe da taxa, para refinar os resultados. Isso facilita encontrar exatamente a taxa desejada de forma mais rápida e precisa.

Estrutura da Tabela HCGS3001 (Taxas)

A estrutura desta tabela já foi detalhada anteriormente.

3. Consulta de Cidades

- O usuário deve ser capaz de buscar cidades por prefixo de nome, país ou contexto de comércio;
- A descrição das cidades deve seguir o padrão "Cidade (UF) - País";



- Para endpoints relacionados a propostas, deve ser possível filtrar cidades de origem e destino com base em tarifários válidos;
- O usuário pode buscar cidades por prefixo de nome, país ou modalidade de transporte;
- É possível consultar cidades de destino válidas para uma origem específica ou pontos de via entre origem e destino;
- As descrições das cidades devem seguir o formato padrão: Cidade (UF) - País.

3.1 Endpoints de Cidades

Para cada uma das ações mencionadas anteriormente, deverá ser implementado um endpoint correspondente, responsável por executar a funcionalidade desejada:

3.1.1 GET `api/City/country`

Este endpoint verifica se a sessão do usuário ainda é válida, seguindo o processo padrão.

Se a sessão não for válida, ele retorna uma mensagem de erro correspondente.

- Validação de Parâmetros: O parâmetro `cityId` precisa conter o código de uma cidade que exista na tabela **CSAG325**. Se esse código não corresponder a nenhuma cidade cadastrada, o endpoint retorna o erro 404 Not Found, indicando que a cidade não foi encontrada.
- Busca Direta: O sistema acessa diretamente o campo **PAIS** na tabela **CSAG325**, usando a cidade informada como referência.



Essa consulta é simples: não utiliza joins nem aplica nenhuma lógica extra.

- **Retorno de Código de País:** O endpoint retorna somente o código do país, como por exemplo **'BR'** para **Brasil** ou **'US'** para **Estados Unidos**.
Nenhuma outra informação sobre o país é incluída na resposta.
- **Uso em Validações:** Este endpoint serve, principalmente, para confirmar dados geográficos de uma cidade.
Ele também é utilizado para aplicar regras fiscais ou aduaneiras, dependendo do país ao qual a cidade pertence.

- **Estrutura da Tabela CSAG325 (Cidades)**

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_CIDADE	VARCHAR(10)	Código único da cidade	PK	HCGS3000.ORIGEM, HCGS3000.DESTINO, HCGS3000.VIA
DESCRICAO	VARCHAR(100)	Nome da cidade	-	-
UF	VARCHAR(5)	Código do estado/ província	-	TsubdivisionCode.SUBDIVISION_CODE
PAIS	CHAR(2)	Código do país	FK	CSAG329.SIGLA
CSAG336_ID	INT	Tipo de cidade	FK	CSAG336.ID
DT_INCLUSAO	DATETIME		-	-
		Data de inclusão do registro		
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se a cidade está ativa ('S'/'N')	-	-

3.1.2 GET api/City/search/by-trade



Este endpoint verifica se a sessão do usuário está válida, seguindo o processo padrão.

Se a sessão não for válida, ele não retorna dados — apenas uma lista vazia.

- **Filtragem por Nome e Países:** O sistema procura por cidades cujo nome (campo **DESCRICAÇÃO**) começa com o texto informado no parâmetro **cityNamePrefix**.

Se o parâmetro **countryIds** também for enviado, o sistema aplica um filtro adicional para considerar apenas as cidades que pertencem aos países listados.

Essa filtragem por país é feita usando a cláusula IN, comparando os códigos dos países com os valores fornecidos na lista.

- **Formatação de Descrição:** Para cada cidade encontrada, o sistema monta uma descrição no seguinte formato: **Cidade (UF) - País**

Essa descrição é criada combinando as seguintes informações:

- O nome da cidade (CSAG325.DESCRICAÇÃO);
- A sigla do estado (CSAG325.UF), colocada entre parênteses;
- O nome do país (CSAG329.DESCRICAÇÃO), adicionado após um hífen.

Exemplo:

Se a cidade for "**São Paulo**", a UF for "**SP**" e o país for "**Brasil**", o resultado será: **São Paulo(SP) – Brasil**

- **Join de Tabelas:** O sistema faz uma junção (join) entre duas tabelas: **CSAG325**, que contém as cidades, e **CSAG329**, que contém os países.

Essa junção é feita relacionando o campo **PAÍS** da tabela de cidades (CSAG325) com o campo **SIGLA** da tabela de países (CSAG329).

- **Filtragem de Cidades Ativas:** Somente as cidades com status ativo (**ATIVO = 'S'**) são incluídas nos resultados. Cidades marcadas como inativas são totalmente excluídas da listagem e não aparecem na resposta.
- **Campos Retornados:** Campos retornados pelo endpoint:



- Id – Código da cidade, vindo do campo CSAG325.ID_CIDADE;
 - Value – Descrição formatada no padrão: "Cidade (UF) - País;
 - Pais – Código do país, conforme o campo CSAG325.PAIS;
 - Uf – Sigla do estado ou província, do campo CSAG325.UF;
 - lata – Este campo existe, mas **não é preenchido** neste endpoint;
 - Port – Também presente, mas **não é preenchido** neste endpoint.
- Ordenação: Os resultados são exibidos em ordem alfabética, com base no nome da cidade, usando o campo **CSAG325.DESCRICAO**.
 - Uso em Interfaces de Comércio: Este endpoint é usado principalmente para selecionar cidades em situações relacionadas ao comércio internacional. Ele permite que as interfaces de usuário filtrem as cidades com base no país selecionado, facilitando a escolha correta no preenchimento de formulários ou cadastros.

Estrutura da Tabela CSAG325 (Cidades)

A estrutura desta tabela já foi detalhada anteriormente.

- Estrutura da Tabela CSAG329 (Países)

Campo	Tipo	Descrição	Chave	Relacionamentos
SIGLA	CHAR(2)	Código do país (ex: 'BR', 'US')	PK	CSAG325.PAIS
DESCRICAO	VARCHAR(100)	Nome do país	-	-
SIGLA_ISO	CHAR(3)		-	-
		Código ISO do país (ex: 'BRA', 'USA')		

DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se o país está ativo ('S'/'N')	-	-

3.1.3 GET api/City/search/via

O endpoint verifica se a sessão do usuário está válida, seguindo o processo padrão.

Se a sessão não for válida, ele responde com uma lista vazia, sem dados.

- Item Especial "DIRECT SERVICE": Se o parâmetro contiver a palavra '**DIRE**' (independentemente de maiúsculas ou minúsculas), um item especial será adicionado no começo da lista.
Esse item terá **Id** = "0" e **Value** = "**DIRECT SERVICE**".
Ele representa uma rota direta, sem paradas intermediárias.
- Filtragem por Nome e Países: O funcionamento deste endpoint é igual ao do **endpoint search/by-trade**.
Ele procura cidades cujo nome começa com o texto informado no parâmetro **cityNamePrefix**.
Se o parâmetro **countryIds** for informado, o sistema também filtra as cidades para incluir somente aquelas que pertencem aos países indicados.
- Formatação de Descrição: Para cada cidade encontrada, o sistema cria uma descrição no formato: **Cidade(UF) - País**. Essa formatação é a mesma utilizada no endpoint **search/by-trade**, garantindo consistência na apresentação dos dados.
- Ordenação com Item Especial: Quando o item '**DIRECT SERVICE**' está presente, ele sempre aparece no topo da lista.
As demais cidades são exibidas em ordem alfabética, com base no nome.
- Uso em Interfaces de Rota: Este endpoint é usado principalmente para configurar os pontos intermediários de uma rota.

Quando a opção '**DIRECT SERVICE**' é selecionada, ela indica que a rota será direta, sem nenhum ponto intermediário.

Estrutura da Tabela CSAG325 (Cidades)

A estrutura desta tabela já foi detalhada anteriormente.

Estrutura da Tabela CSAG329 (Países)

A estrutura desta tabela já foi detalhada anteriormente.

3.1.4 GET api/City/description/{cityCode}

Este endpoint é uma exceção e pode ser acessado sem verificar a sessão do usuário.

Isso é permitido porque ele fornece apenas informações de referência, que não são sensíveis.

- Validação de Parâmetros: O parâmetro cityCode precisa ser um código válido de uma cidade que esteja cadastrada na tabela **CSAG325**.
Se o código informado não corresponder a nenhuma cidade existente, o endpoint retorna o erro 404 Not Found.
- Formatação de Descrição: O sistema cria uma descrição no formato: **Cidade(UF) - País**.
- Para isso, ele junta as seguintes informações:
 - O nome da cidade, que está no campo CSAG325.DESCRICAO
 - A sigla do estado (UF), que fica entre parênteses, do campo CSAG325.UF
 - O nome do país, que vem após um hífen, do campo CSAG329.DESCRICAO



- **Join de Tabelas:** O sistema combina informações de duas tabelas: **CSAG325**, que contém dados das cidades, e **CSAG329**, que contém dados dos países.
Essa combinação é feita ligando o campo **PAIS** da tabela de cidades (CSAG325) ao campo **SIGLA** da tabela de países (CSAG329).
- **Uso em Padronização:** Este endpoint é usado principalmente para mostrar informações de cidades de forma padronizada. Ele ajuda a manter a apresentação das cidades consistente em várias partes do sistema.

Estrutura da Tabela CSAG325 (Cidades)

A estrutura desta tabela já foi detalhada anteriormente.

Estrutura da Tabela CSAG329 (Países)

A estrutura desta tabela já foi detalhada anteriormente.

3.1.5 GET api/City/proposal/via-points

- **Validação de Parâmetros:** Os seguintes parâmetros são obrigatórios para usar este endpoint:
 - pol (porto ou local de origem);
 - pod (porto ou local de destino);
 - mode (modal de transporte);
 - clientId (identificador do cliente);

Se qualquer um desses parâmetros não for informado, o endpoint retorna um erro indicando que a requisição está incompleta ou incorreta.



- **Tipo de Cliente:** O sistema consulta a tabela **CSAG340** para descobrir o tipo do cliente (por exemplo, 'CL' ou 'CD') com base no valor do **clientId**.
Esse tipo de cliente é usado para filtrar quais tarifários são aplicáveis à consulta.
- **Tabela por Modal:** O sistema escolhe a tabela correta com base no modal de transporte informado:
Se o modal for 'AIR' (transporte aéreo), a consulta é feita na tabela **HCGS3000_AIRFR8**.
Para todos os outros modais, a consulta é feita na tabela **HCGS3000**.
- **Filtragem por Origem, Destino e Validade:** O sistema busca tarifários que correspondam aos seguintes critérios: **origem, destino e tipo de cliente informados**.
Somente tarifários válidos na **data atual** (ou seja, dentro do período de validade) são considerados.
Para o modal **aéreo**, a busca usa o campo **ID_CIDADE** para identificar a origem e o destino.
Para os demais modais, o campo utilizado é **VIA**.
- **Formatação de Descrição:** Para cada cidade encontrada, o sistema gera uma descrição no formato: **Cidade(UF) – País**. Essa padronização é a mesma utilizada nos endpoints anteriores, garantindo consistência na apresentação das informações.
- **Campos Retornados:**
 - **Id:** Representa o código da cidade, obtido do campo CSAG325.ID_CIDADE.
 - **Value:** Contém a descrição formatada da cidade no padrão "Cidade(UF) - País"

Exemplo:

 - Nome da cidade: São Paulo (CSAG325.DESCRICAO)
 - Sigla do estado: SP (CSAG325.UF)
 - Nome do país: Brasil (CSAG329.DESCRICAO)
- **Ordenação:** Os resultados são organizados em ordem alfabética, com base no nome da cidade, que está no campo CSAG325.DESCRICAO.



Essa ordenação facilita a navegação e a seleção em interfaces de usuário, como menus suspensos (dropdowns) ou campos de busca.

- Seleção de Rotas: Este endpoint é usado principalmente para selecionar pontos de rota ao criar uma nova proposta. Ele permite consultar as rotas disponíveis entre uma origem e um destino específico, ajudando a identificar os caminhos possíveis para o transporte.

• Estrutura da Tabela HCGS3000 (Tarifários)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_TARIFA	VARCHAR(10)	Código único do tarifário	PK	-
ORIGEM	VARCHAR(10)	Cidade de origem	FK	CSAG325.ID_CIDADE
DESTINO	VARCHAR(10)	Cidade de destino	FK	CSAG325.ID_CIDADE
VIA	VARCHAR(10)	Cidade de via/rota	FK	CSAG325.ID_CIDADE
MODAL	VARCHAR(5)	Modo de transporte	FK	CSAG337.SIGLA
TIPO_CLIENTE	CHAR(2)	Tipo de cliente ('CL'/'CD')	-	-
DT_VALIDADE_INICIO	DATE	Data de início da validade	-	-
DT_VALIDADE_FIM	DATE	Data de fim da validade	-	-
VALOR	DECIMAL(15,2)	Valor do tarifário	-	-

MOEDA	CHAR(3)	Código da moeda	F K	CSAG331.SIGLA
DT_INCLUSÃO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	F K	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se o tarifário está ativo ('S'/'N')	-	-

- Estrutura da Tabela HCGS3000_AIRFR8 (Tarifários Aéreos)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_TARIFA	VARCHAR(10)	Código único do tarifário	P K	-
ORIGEM	VARCHAR(10)	Cidade de origem	F K	CSAG325.ID_CIDADE
DESTINO	VARCHAR(10)	Cidade de destino	F K	CSAG325.ID_CIDADE
ID_CIDADE	VARCHAR(10)	Cidade de via/escala	F K	CSAG325.ID_CIDADE
TIPO_CLIENTE	CHAR(2)	Tipo de cliente ('CL'/'CD')	-	-
DT_VALIDADE_INICIO	DATE	Data de início da validade	-	-
DT_VALIDADE_FIM	DATE	Data de fim da validade	-	-

VALOR	DECIMAL(15,2)	Valor do tarifário	-	-
MOEDA	CHAR(3)	Código da moeda	FK	CSAG331.SIGLA
DT_INCLUSÃO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se o tarifário está ativo ('S'/'N')	-	-

- Estrutura da Tabela CSAG340 (Clientes)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_CLIENTE	VARCHAR(10)	Código único do cliente	PK	-
NOME	VARCHAR(200)	Nome/razão social do cliente	-	-
TIPO	CHAR(2)	Tipo de cliente ('CL'/'CD')	-	-
CNPJ	VARCHAR(14)	CNPJ do cliente	-	-
ENDereco	VARCHAR(200)	Endereço do cliente	-	-
CIDADE	VARCHAR(10)	Cidade do cliente	FK	CSAG325.ID_CIDADE
DT_INCLUSÃO	DATETIME	Data de inclusão do registro	-	-

ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	F K	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se o cliente está ativo ('S'/'N')	-	-

3.1.6 GET api/City/proposal/origin-cities

Este endpoint verifica se a sessão do usuário está válida, seguindo o processo padrão do sistema. Se a sessão não for válida, ele não retorna nenhum dado, apenas uma lista vazia.

- Busca Ampliada: O sistema procura cidades que atendam a pelo menos um dos seguintes critérios, com base no valor informado em **cityNamePrefix**:
 - O nome da cidade (campo CSAG325.DESCRICAO) começa com o texto informado (**cityNamePrefix**);
 - No modo aéreo, o **código IATA** (campo HSAG325_AIR.IATA) também pode começar com esse texto (**cityNamePrefix**);
 - Ainda no **modo aéreo**, o **nome do aeroporto** (campo HSAG325_AIR.NOME) pode começar com esse mesmo prefixo (**cityNamePrefix**).
- Filtragem por País: Se o parâmetro countryId for enviado, o sistema mostra apenas as cidades que pertencem a esse país. Ele faz isso verificando se o valor do campo PAIS da tabela **CSAG325** é igual ao valor de countryId.
- Verificação de Tarifários: Para cada cidade, o sistema verifica se existe alguma tarifa válida onde essa cidade é usada como ponto de origem.



Essa verificação depende do modo de transporte (aéreo ou outro) e do tipo de cliente.

- Se o transporte for aéreo, a verificação é feita na tabela **HCGS3000_AIRFR8**;
 - Para os outros tipos de transporte, a verificação é feita na tabela **HCGS3000**.
- Formatação de Descrição: Para cada cidade, o sistema monta uma descrição no formato: **Cidade(UF) - País**
Essa forma de exibir as informações segue o mesmo padrão usado nos outros endpoints, garantindo consistência em todo o sistema.
 - Campos Retornados:
 - Id: Código da cidade, retirado do campo **CSAG325.ID_CIDADE**;
 - Value: Nome formatado da cidade no padrão "**Cidade(UF) - País**";
 - Pais: Código do país da cidade, do campo **CSAG325.PAIS**;
 - Uf: Sigla do estado ou província, vinda do campo **CSAG325.UF**;
 - Iata: Código IATA do aeroporto, usado apenas para o modo aéreo (campo **HSAG325_AIR.IATA**);
 - Port: Este campo existe, mas **não é preenchido** neste endpoint.
 - Ordenação: Os resultados são organizados em ordem alfabética, com base no nome da cidade (campo **CSAG325.DESCRICAO**). Isso facilita a leitura e a localização das cidades na lista.
 - Criação de Propostas: Este endpoint é utilizado principalmente na etapa de criação de propostas, para selecionar a cidade de origem.
Além disso, ele informa se já existem tarifários cadastrados para cada cidade, por meio de um indicador.
Isso facilita a escolha, destacando as cidades que já possuem tarifas disponíveis no sistema.
 - Criação de Propostas: Este endpoint é utilizado principalmente na etapa de criação de propostas, para selecionar a cidade de origem.



Além disso, ele informa se já existem tarifários cadastrados para cada cidade, por meio de um indicador. Isso facilita a escolha, destacando as cidades que já possuem tarifas disponíveis no sistema.

Estrutura da Tabela CSAG325 (Cidades)

A estrutura desta tabela já foi detalhada anteriormente.

- Estrutura da Tabela HSAG325_AIR (Aeroportos)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_CIDADE	VARCHAR(10)	Código da cidade	PK, FK	CSAG325.ID_CIDADE
IATA	CHAR(3)	Código IATA do aeroporto	-	-
NOME	VARCHAR(100)	Nome do aeroporto	-	-
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se o aeroporto está ativo ('S'/'N')	-	-

3.1.7 GET api/City/proposal/destination-cities

Antes de processar a solicitação, o endpoint verifica se a sessão do usuário é válida, seguindo as regras padrão do sistema. Se a sessão estiver inválida ou expirada, a resposta será uma lista vazia, ou seja, nenhum dado será retornado.



- **Parâmetro de origem:** Este endpoint aceita um parâmetro chamado **pol**, que indica de onde a carga vai sair (**cidade de origem**).
Com base nesse valor, o sistema mostra somente os destinos que podem ser usados com essa origem.
- **Busca Ampliada:** O funcionamento deste endpoint é semelhante ao **origin-cities**. Ele busca cidades com base no que foi digitado em **cityNamePrefix**, seguindo estes critérios:
 - O **nome da cidade** começa com o valor de **cityNamePrefix**;
 - No caso de transporte **aéreo**:
 - O **código IATA** começa com **cityNamePrefix**;
 - O **nome do aeroporto** começa com **cityNamePrefix**.

Em resumo, o sistema tenta encontrar correspondências que comecem com o texto informado, seja no nome da cidade, no código IATA ou no nome do aeroporto (quando o modo for aéreo).

- **Filtragem por País:** Se o parâmetro **countryId** (ID do país) for informado, o sistema aplicará um filtro adicional para considerar apenas os registros que correspondem a esse país.
Esse filtro é realizado utilizando o operador de igualdade (=) na coluna **PAIS** da tabela **CSAG325**, ou seja, o sistema busca registros onde o valor da coluna PAIS seja igual ao valor fornecido em **countryId**.
- **Verificação de Tarifários para Par Origem-Destino:** Para cada cidade, o sistema verifica se existe um tarifário válido que atenda a todos os seguintes critérios:
 - Cidade de origem: corresponde ao valor especificado no parâmetro **pol**;
 - Cidade de destino: é a cidade atualmente sendo avaliada;
 - Modo de transporte e tipo de cliente: coincidem com os valores fornecidos.

Essa verificação é mais restritiva do que a realizada pelo **endpoint origin-cities**, pois considera o par completo origem-destino, além do modo de transporte e tipo de cliente.



- **Formatação de Descrição:** Para cada cidade, o sistema cria uma descrição no formato: "Cidade (UF) - País". Por exemplo:
 - **Belém (PA) - Brasil**
 - **Rio de Janeiro (RJ) - Brasil**Essa formatação segue o mesmo padrão utilizado nos endpoints anteriores, garantindo consistência na apresentação das informações.
- **Campos Retornados:**
 - **Id:** Código único da cidade, armazenado no campo **CSAG325.ID_CIDADE;**
 - **Value:** Descrição formatada da cidade, no padrão: "Cidade (UF) - País";
 - **País:** Código do país, armazenado no campo **CSAG325.PAIS;**
 - **Uf:** Código da unidade federativa (estado ou província), armazenado no campo **CSAG325.UF;**
 - **Iata:** Código IATA do aeroporto, armazenado no campo **HSAG325_AIR.IATA**, aplicável apenas quando o modo de transporte é aéreo;
 - **Port:** Este campo não é utilizado neste endpoint.
- **Ordenação:** Os resultados são organizados em ordem alfabética crescente pelo **nome da cidade**, conforme registrado no campo **CSAG325.DESCRICAO**.
Isso significa que, ao apresentar os dados, as cidades serão listadas de A a Z, facilitando a localização e análise das informações.
- **Uso em Criação de Propostas:** Este endpoint é utilizado para selecionar a cidade de destino após a escolha da cidade de origem. Ele permite filtrar as cidades de destino disponíveis com base na cidade de origem selecionada, facilitando a busca por destinos viáveis para a viagem.

Estrutura da Tabela CSAG325 (Cidades)

A estrutura desta tabela já foi detalhada anteriormente.

Estrutura da Tabela HSAG325_AIR (Aeroportos)

A estrutura desta tabela já foi detalhada anteriormente.



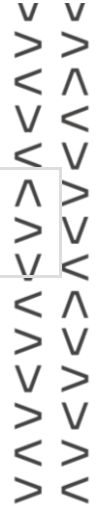
4. Tabelas Auxiliares

- Estrutura da Tabela CSAG310 (Usuários)

Campo	Tipo	Descrição	Chave	Relacionamentos
ID_USUARIO	VARCHAR(20)	Código único do usuário	PK	CSAG311.ID_USUARIO, várias FK em outras tabelas
NOME	VARCHAR(100)	Nome completo do usuário	-	-
EMAIL	VARCHAR(100)	Email do usuário	-	-
SENHA	VARCHAR(50)	Senha criptografada	-	-
PERFIL	VARCHAR(10)	Código do perfil de acesso	FK	CSAG309.ID_PERFIL
DT_ULTIMO_ACESSO	DATETIME	Data/hora do último acesso	-	-
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ATIVO	CHAR(1)	Indicador se o usuário está ativo ('S'/'N')	-	-

- Estrutura da Tabela CSAG337 (Modais de Transporte)

Campo	Tipo	Descrição	Chave	Relacionamentos
SIGLA	VARCHAR(5)	Código do modal (ex: 'AIR', 'SEA', 'ROAD')	PK	HCGS3000.MODAL
DESCRICAO	VARCHAR(50)	Descrição do modal	-	-
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO



ATIVO	CHAR(1)	Indicador se o modal está ativo ('S'/'N')	-	-
-------	---------	---	---	---

- Estrutura da Tabela CSAG337 (Modais de Transporte)

Campo	Tipo	Descrição	Chave	Relacionamentos
SIGLA	VARCHAR(5)	Código do modal (ex: 'AIR', 'SEA', 'ROAD')	PK	HCGS3000.MODAL
DESCRICAO	VARCHAR(50)	Descrição do modal	-	-
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se o modal está ativo ('S'/'N')	-	-
		Símbolo da moeda		
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se a moeda está ativa ('S'/'N')	-	-

- Estrutura da Tabela TsubdivisionCode (Subdivisões/Estados)

Campo	Tipo	Descrição	Chave	Relacionamentos
COUNTRY_CODE	CHAR(2)	Código do país	PK	CSAG329.SIGLA
SUBDIVISION_CODE	VARCHAR(5)	Código da subdivisão/ estado	PK	CSAG325.UF
SUBDIVISION_NAME	VARCHAR(100)	Nome da subdivisão/ estado	-	-
DT_INCLUSAO	DATETIME	Data de inclusão do registro	-	-
ID_USUARIO	VARCHAR(20)	Usuário que incluiu o registro	FK	CSAG310.ID_USUARIO
ATIVO	CHAR(1)	Indicador se a subdivisão está ativa ('S'/'N')	-	-

5. Considerações Finais

5.1 Validação de Sessão

A validação de sessão é essencial para garantir que apenas usuários autenticados possam acessar a **API CROSS**. Ela é realizada verificando se a sessão do usuário está ativa, consultando a tabela **CSAG311**, que armazena as sessões ativas. Se a sessão estiver expirada ou inválida, o acesso é negado, protegendo os dados e a integridade do sistema.

5.2 Formatação de Descrições de Cidades

Para manter consistência e facilitar a identificação de cidades, todas as descrições seguem o formato padrão: "**Cidade (UF) - País**". Por exemplo: "**São Paulo (SP) - Brasil**". Essa padronização é aplicada em todos os endpoints que



retornam informações de cidades, garantindo uniformidade em relatórios e interfaces de usuário.

5.3 Tarifários e Validade

Os tarifários, armazenados nas tabelas **HCGS3000** e **HCGS3000_AIRFR8**, possuem períodos de validade definidos. Ao buscar informações sobre tarifários, o sistema considera apenas os registros válidos na data atual, assegurando que apenas informações atualizadas sejam utilizadas nas operações.

5.4 Tipos de Cliente

Os clientes são classificados em tipos, como '**CL**' para cliente regular e '**CD**' para cliente direto. Essa categorização influencia quais tarifários são aplicáveis em determinados contextos, permitindo a implementação de políticas de preços diferenciadas por tipo de cliente.

5.5 Tratamento de Parâmetros Opcionais

Quando um parâmetro é marcado como opcional, o sistema precisa ser capaz de lidar com ele de forma flexível, ou seja, considerar tanto quando o parâmetro é fornecido quanto quando não é.

Para isso, nas consultas SQL, utilizamos uma técnica que verifica se o parâmetro foi informado. Se não foi (ou seja, é **NULL**), a condição correspondente é ignorada, permitindo que a consulta retorne todos os resultados possíveis para aquele campo. Se o parâmetro foi informado, a condição é aplicada normalmente, filtrando os resultados conforme o valor fornecido.

Por exemplo:

Se **@cidade** for **NULL** (não informado), a condição será verdadeira para todos os registros, ou seja, não haverá filtro por cidade.

Se **@cidade** tiver um valor, a condição será verdadeira apenas para os registros onde o campo cidade corresponda a esse valor.

Essa abordagem permite que a consulta seja flexível, adaptando-se ao contexto em que é utilizada, seja com filtros mais específicos ou mais amplos, conforme a necessidade.

