



## Manual de Integração

### API / Webservice REST

Versão 0.3

Este manual descreve a API de integração com o sistema Meus Pedidos. Em caso de dúvidas, entre em contato pelo email [tiago@meuspedidos.com.br](mailto:tiago@meuspedidos.com.br).

## Conteúdo

<b>1. Visão Geral</b>	<b>4</b>
<b>2. Como funciona?</b>	<b>5</b>
<i>Buscando Registros</i>	<i>5</i>
<i>Criando Novos Registros</i>	<i>5</i>
<i>Atualizando Registros</i>	<i>6</i>
<i>Excluindo Registros</i>	<i>6</i>
<i>Validação de Dados</i>	<i>6</i>
<b>3. Sincronizando Dados na Prática</b>	<b>8</b>
<i>Buscando Informações: exemplo prático</i>	<i>8</i>
<i>Enviando Informações: exemplo prático</i>	<i>9</i>
<b>4. Autenticação e Segurança</b>	<b>10</b>
<b>5. Entidades do MeusPedidos</b>	<b>11</b>
<b>5.1. Clientes</b>	<b>12</b>
<i>Email</i>	<i>13</i>
<i>Telefone</i>	<i>13</i>
<i>Contato</i>	<i>13</i>
<b>5.2. Produtos</b>	<b>16</b>
<b>5.3. Tabelas de Preço</b>	<b>18</b>
<b>5.4. Produtos &lt;-&gt; Tabelas de Preço</b>	<b>20</b>
<b>5.5. Transportadoras</b>	<b>21</b>
<b>5.6. Usuários (Vendedores)</b>	<b>22</b>
<b>5.7. Condições de Pagamento</b>	<b>23</b>

<b>5.8. Formas de Pagamento</b>	<b>24</b>
<b>5.9. Pedidos</b>	<b>25</b>
<b>Itens do Pedido</b>	<b>26</b>
<b>Campos Extras do Pedido</b>	<b>27</b>
<b>5.10. Campos Extras</b>	<b>29</b>
<b>6. Histórico de Revisões</b>	<b>30</b>

# 1. Visão Geral

A integração com o MeusPedidos é feita através de uma API em protocolo HTTP (modelo REST), e os dados são comunicados em formato JSON. Apesar das siglas complicadas, o processo é muito simples. Você irá se comunicar com nossos servidores através de requisições HTTP, de forma muito parecida com um navegador de internet.

O protocolo HTTP é utilizado pelos navegadores para se comunicar com servidores web, através dos comandos GET e POST. Você irá utilizar estes mesmos comandos para fazer leitura, cadastro e atualização de registros em nossa API.

É possível desenvolver a integração em qualquer linguagem de programação. Os únicos componentes que você irá precisar são: uma biblioteca para realizar conexões HTTP, e uma biblioteca para leitura/escrita de dados em formato JSON. A maioria das linguagens possuem bibliotecas prontas para este tipo de operação.

## 2. Como funciona?

Cada **entidade** (Pedidos, Clientes, Produtos, etc) existente no MeusPedidos é acessível através de uma URL. Para fazer a leitura, cadastro e alteração destas entidades, você deverá executar alguns comandos HTTP nesta URL (os comandos serão detalhados mais adiante).

Por exemplo, vamos analisar a entidade **Clientes**. A URL que identifica os clientes é a seguinte: <http://meuspedidos.com.br/api/v1/clientes/>. Chamamos isto de **URL raiz** da entidade Clientes. Ela representa todos os clientes cadastrados no sistema.

Cada registro individual também possui sua URL. Esta URL é sempre composta da URL raiz da entidade, **seguida pelo ID do registro**. Então, um cliente cujo ID no banco de dados seja 123, será representado pela URL <http://meuspedidos.com.br/api/v1/clientes/123/>.

### Buscando Registros

Para buscar os registros de uma entidade, deve-se utilizar o comando HTTP GET.

Se você fizer um GET na URL raiz da entidade, será retornada uma lista com todos os registros cadastrados. Se você fizer GET na URL específica de 1 registro, será retornado apenas aquele registro individual.

Por exemplo, para buscar todos os clientes cadastrados, basta fazer um GET no endereço <http://meuspedidos.com.br/api/v1/clientes/>. Para buscar o cliente 1001, basta fazer GET no endereço <http://meuspedidos.com.br/api/v1/clientes/1001/>.

### Criando Novos Registros

Para criar novos registros, deve-se executar um comando HTTP POST na URL raiz da entidade. No corpo da requisição, você deve enviar as informações do registro em formato JSON. Também deverá ser informado o header “Content-Type” = “application/json” na requisição.

Em caso de sucesso, será retornado o StatusCode 201 (CREATED), e o corpo da resposta virá em branco. Serão enviados 2 *headers* HTTP identificando o novo registro criado. São eles:

- MeusPedidosID: contém o ID do registro no banco de dados do MeusPedidos.
- MeusPedidosURL: contém a URL de acesso ao registro. É sempre formada pela URL raiz, seguida do ID do registro.

Continuando o exemplo dos clientes, caso você queira cadastrar um novo cliente, basta fazer POST na URL <http://meuspedidos.com.br/api/v1/clientes/>. Digamos que o servidor salve este registro em nosso Banco de Dados, e o ID gerado seja 456. Então, a API irá retornar os seguintes *headers*:

- MeusPedidosID: 456
- MeusPedidosURL: <http://meuspedidos.com.br/api/v1/clientes/456/>

## Atualizando Registros

Para atualizar um registro já existente, basta executar um comando PUT na URL do registro, enviando no corpo da requisição as informações em formato JSON. (Este comando PUT não é utilizado pelos navegadores web, mas é um comando válido do protocolo HTTP). Também deverá ser informado o header “Content-Type” = “application/json” na requisição.

Em caso de sucesso, será retornado o StatusCode 200 (OK), e o corpo da resposta virá em branco. Nenhum *header* será enviado.

## Excluindo Registros

Para preservar a integridade das informações, o MeusPedidos trabalha com o conceito de exclusão lógica de dados. Todos os registros possuem uma propriedade **excluido** para este fim. Portanto, para realizar exclusão de dados, basta executar um comando PUT, informando o valor *true* na propriedade **excluido** do registro.

## Validação de Dados

Ao executar comandos POST e PUT, a API irá validar os dados enviados, de acordo com a descrição da Entidade. Em caso de erro de validação, será retornado o StatusCode 412 (Precondition Failed), e no corpo da resposta será enviado um objeto JSON descrevendo os erros encontrados. O exemplo abaixo mostra uma situação onde o campo **nome** da Entidade era obrigatório, mas não foi informado.

```
{
  "mensagem": "Dados inválidos",
  "erros": [
    {
      "mensagem": "Este campo é obrigatório.",
      "campo": "nome"
    }
  ]
}
```

### 3. Sincronizando Dados na Prática

A API do Meus Pedidos trabalha com um mecanismo de sincronização incremental, baseado na data de última alteração de cada registro. Desta forma, você poderá buscar apenas os registros que foram cadastrados ou sofreram alteração no Meus Pedidos (novos pedidos, novos clientes) evitando o tráfego desnecessário de dados. Sem este mecanismo, a API retornaria todos os registros cadastrados em toda requisição, o que tornaria processamento mais demorado.

Cada registro da API possui um campo chamado **ultima\_alteracao**. Como o próprio nome indica, este campo guarda a hora exata em que o registro foi alterado dentro do Meus Pedidos. Ao fazer um GET, você poderá enviar o parâmetro **alterado\_apos** na URL. Ao fazer isto, serão retornados apenas os registros cuja **ultima\_alteracao** seja superior ao **alterado\_apos** que você enviou.

Exemplo:

GET [http://meuspedidos.com.br/api/v1/clientes?alterado\\_apos=2014-02-28%2012:32:55](http://meuspedidos.com.br/api/v1/clientes?alterado_apos=2014-02-28%2012:32:55)  
Isto retornará os clientes que sofreram alteração depois das 12:32:55 do dia 28 de Fevereiro de 2014 (detalhe: os caracteres “%20” entre a Data e a Hora representam o espaço em branco no formato URL Encoding. Referência: [http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp)).

#### Buscando Informações: exemplo prático

A forma ideal de sincronizar os dados é manter uma tabela de mapeamento para cada Entidade (clientes, produtos, pedidos, etc). Estas tabelas irão relacionar os IDs do seu ERP aos IDs do Meus Pedidos. Também é importante que a tabela armazene o campo **ultima\_alteracao** dos registros do Meus Pedidos. Assim, antes de fazer um GET na API, você poderá buscar o maior valor deste campo (ex: select max(ultima\_alteracao) from mapeamento\_pedidos), e enviar este valor no parâmetro **alterado\_apos**. A API irá retornar apenas os pedidos que foram alterados depois da sua última requisição.

Digamos que você executou um GET em <http://meuspedidos.com.br/api/v1/pedidos/> e foi retornado 1 único pedido, cujo ID no Meus Pedidos é 9876543 e **ultima\_alteracao** é “2014-03-01 07:21:44”. Digamos que você salve este pedido no seu banco de dados, gerando o ID 4312 no seu ERP. Em seguida, insira um registro na tabela de mapeamento de pedidos, informando que o seu ID 4312 equivale ao ID 9876543 do Meus Pedidos. Salve também neste registro o campo **ultima\_alteracao** do pedido.

Na próxima vez que você buscar Pedidos, selecione o maior valor da coluna **ultima\_alteracao** na tabela de mapeamento de pedidos, e envie este valor no parâmetro **alterado\_apos** da API. Neste caso, ficaria: GET [http://meuspedidos.com.br/api/v1/pedidos/?alterado\\_apos=2014-03-01%2007:21:44](http://meuspedidos.com.br/api/v1/pedidos/?alterado_apos=2014-03-01%2007:21:44).

Caso aquele pedido ID 9876543 tenha sido alterado no Meus Pedidos, ele será enviado novamente pela API. Ao verificar a sua tabela de mapeamento, você vai descobrir que ele

já está cadastrado no seu ERP. Assim, poderá atualizar o registro já existente, evitando a duplicação de Pedidos.

**Detalhe importante:** nestas operações GET, o servidor retorna no máximo 500 registros por requisição, para evitar longos processamentos no caso de bases grandes. Portanto, sempre que você executar um GET e forem retornados exatamente 500 registros, você deverá salvá-los no banco de dados, em seguida executar uma nova operação GET (não esquecendo de atualizar o parâmetro **alterado\_apos**). Os registros retornados pelo servidor são ordenados pelo campo **ultima\_alteracao**, portanto não existe risco de perda de dados, caso este limite de 500 registros seja atingido. Basta executar a operação novamente.

### Enviando Informações: exemplo prático

Digamos que seu usuário cadastrou um novo Produto, cujo ID é 1050 no seu ERP. Para cadastrar este produto no Meus Pedidos, você faz um POST em <http://meuspedidos.com.br/api/v1/produtos/> com os dados. Digamos que o servidor retorne com sucesso, e informe o *header* MeusPedidosID=12345678. Neste momento, você irá inserir um registro na sua tabela de mapeamento de produtos, informando que o seu produto ID 1050 equivale ao produto ID 12345678 no Meus Pedidos. O campo **ultima\_alteracao** deverá ser nulo, para não interferir no processo de busca de dados.

No futuro, quando este produto ID 1050 sofrer alteração no seu sistema, você saberá que ele equivale ao ID 12345678 no Meus Pedidos. Então, basta fazer um PUT no endereço <http://meuspedidos.com.br/api/v1/produtos/12345678/> com as novas informações. Esta é a forma correta de manter os 2 sistemas sincronizados.

## 4. Autenticação e Segurança

A segurança dos seus dados é uma das nossas maiores prioridades. Por isso nossa API possui um mecanismo de autenticação que impede que outros usuários tenham acesso e façam qualquer tipo de alteração nas informações dos seus clientes.

Este mecanismo de autenticação é baseado em *tokens*. Os *tokens* são parecidos com uma senha de acesso, mas com a vantagem de serem únicos dentro do sistema. Eles são gerados automaticamente e nunca são repetidos. Desta forma, garantimos que todos os clientes e todas as empresas parceiras possuem um *token* exclusivo que as identifica.

Ao fazer qualquer operação na API, você deverá enviar 2 *tokens* nos *headers* do comando HTTP:

- ApplicationToken: Identifica a aplicação externa (por exemplo, seu ERP) que está se conectando.
- CompanyToken: Identifica o cliente da aplicação externa (os clientes da sua empresa) que está se conectando.

Ao receber estes *tokens* de acesso, será sua responsabilidade guardar estas chaves em segurança, para garantir que ninguém fora da sua empresa tenha acesso aos dados.

## 5. Entidades do MeusPedidos

Nesta seção iremos detalhar todas as entidades disponíveis na API. Estão disponíveis na API quase todas as entidades presentes no sistema Web, com exceção de Comissões e Faturamento de Pedidos.

As entidades são formadas por diversas propriedades, e cada propriedade possui um tipo específico. A API realiza a validação dos dados com base nestes tipos de dados, e recusa operações que não estejam de acordo. Os tipos aceitos são:

- String: sequência de caracteres, envoltos em aspas duplas. Caso a quantidade de caracteres seja limitada, também é informado a quantidade máxima aceita. Ex: String(30) significa que o campo aceita no máximo 30 caracteres.
- Boolean: *true* ou *false*.
- Integer: número inteiro.
- Double: número com casas decimais, utiliza o “.” (ponto) como separador decimal.
- Date: data no formato yyyy-mm-dd.
- DateTime: data e hora no formato yyyy-mm-dd hh:MM:ss.

## 5.1. Clientes

- URL Raíz: <http://meuspedidos.com.br/api/v1/clientes/>
- Métodos: GET, GET(id), POST, PUT(id)
- Campos obrigatórios: razao\_social, tipo
- Campos:

Campo	Tipo	Observações
id	Integer	Identificador único (somente leitura)
razao_social	String(100)	Razão social para pessoa jurídica. Nome do cliente para pessoa física.
nome_fantasia	String(100)	Nome fantasia (somente pessoa jurídica).
tipo	String(1)	“J” para pessoa jurídica, “F” para pessoa física.
cnpj	String(18)	CNPJ para pessoa jurídica, CPF para pessoa física. Apenas números, sem pontuação.
inscricao_estadual	String(30)	
suframa	String(20)	Código Suframa para clientes da Zona Franca de Manaus. Caso seja informado, todos os pedidos deste cliente terão IPI zerado (isento).
rua	String(100)	Endereço do cliente (nome da rua e número)
complemento	String(50)	
cep	String(9)	Pode ser informado com ou sem hífen.
bairro	String(30)	
cidade	String(50)	
estado	String(2)	Sigla do Estado.
observacao	String(500)	Utilize para guardar quaisquer informações que não tenham campos específicos.
emails		Lista de objetos Email (descritos adiante) com os emails do cliente.

Campo	Tipo	Observações
telefones		Lista de objetos Telefone (descritos adiante) com os telefones do cliente.
contatos		Lista de objetos Contato (descritos adiante) com os contatos do cliente.
excluido	Boolean	Indica se o cliente está excluído.
ultima_alteracao	DateTime	Data e hora da última modificação deste cliente no Meus Pedidos (somente leitura).

## Email

Segue a descrição do objeto Email. Apenas a propriedade "email" é obrigatória.

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
email	String(75)	Email
tipo	String(1)	<i>Atenção: este campo não é mais utilizado. Favor ignorar.</i>

## Telefone

Segue a descrição do objeto Telefone. Apenas a propriedade "numero" é obrigatória.

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
numero	String(30)	Número do telefone, com DDD e formatação inclusos.
tipo	String(1)	<i>Atenção: este campo não é mais utilizado. Favor ignorar.</i>

## Contato

Segue a descrição do objeto Contato. Apenas a propriedade "nome" é obrigatória.

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(50)	Nome do contato.
cargo	String(30)	
emails		Lista de objetos Email contendo os emails do contato.
telefones		Lista de objetos Telefone contendo os telefones do contato.
excluido	Boolean	Indica se o contato está excluído.

Exemplo de cliente em formato JSON:

```
{
  "id": 123,
  "razao_social": "Loja do Zé LTDA",
  "nome_fantasia": "Zé Store",
  "tipo": "J",
  "cnpj": "12345678000180",
  "inscricao_estadual": "ISENTO",
  "suframa": "",
  "rua": "Rua Abdon Batista, 121",
  "complemento": "sala 1402",
  "bairro": "Centro",
  "cep": "89201010",
  "cidade": "Joinville",
  "estado": "SC",
  "observacao": "Cliente com ótimo histórico de
pagamentos.",
  "emails": [{
    "email": "jose@zestore.com.br"
  }, {
    "email": "marcos@zestore.com.br"
  }],
  "telefones": [{
    "numero": "(11) 98765-4321"
  }, {
    "numero": "(47) 9876-5432"
  }],
  "contatos": [{
    "nome": "Lucas da Silva",
    "cargo": "Gerente de Compras",
    "emails": [{
      "email": "lucas@zestore.com.br"
    }],
    "telefones": [{
      "numero": "(21) 1111-1234"
    }]
  }],
  "excluido": false,
  "ultima_alteracao": "2014-01-25 02:12:43"
}
```

## 5.2. Produtos

Cada produto possui um preço de venda padrão, informado nesta entidade. Para produtos que possuem múltiplas tabelas de preço, é possível cadastrar tabelas adicionais através da entidade Tabela de Preço.

É possível informar o estoque do produto no campo **saldo\_estoque**. Essa informação será mostrada para o vendedor na emissão do pedido. Caso não deseje trabalhar com estoque, informe o valor *null* neste campo.

É possível trabalhar com recurso de Grade de cores e tamanho, bastando informar nos campos `grade_cores` e `grade_tamANHos`. Caso não utilize este recurso, informe *null* nestes campos.

- URL Raíz: <http://meuspedidos.com.br/api/v1/produtos/>
- Métodos: GET, GET(id), POST, PUT(id)
- Campos Obrigatórios: nome, preco\_tabela
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
codigo	String(50)	Código de referência do produto.
nome	String(100)	
comissao	Double	Comissão do produto. Ex: 5.0 para 5%. Informe null caso não queira calcular comissões pelo Meus Pedidos.
preco_tabela	Double	Preço deste produto na tabela padrão.
ipi	Double	Caso não possua IPI, informe <i>null</i> .
tipo_ipi	String(1)	Tipo do IPI do produto: "P" para percentual, "V" para valor fixo em Reais.
st	Double	Percentual de Substituição Tributária do produto. Caso não possua ST, informe null.
grade_cores		Lista de cores (String) em que o produto está disponível. Caso não trabalhe com variação de cores, informe <i>null</i> .

Nome	Tipo	Observações
grade_tamANHos		Lista de tamanhos (String) em que o produto está disponível. Caso não trabalhe com variação de tamanhos, informe <i>null</i> .
moeda	String(1)	Moeda do produto: Real ("0"), Dolar ("1") ou Euro ("2")
unidade	String(10)	Unidade de medida do produto, por exemplo "Kg" ou "Cx". Opcional.
saldo_estoque	Double	Caso não controle estoque, informe <i>null</i> .
observacoes	String(500)	Utilize para guardar quaisquer informações que não tenham campos específicos.
excluido	Boolean	Indica se o produto está excluído.
ultima_alteracao	DateTime	Data e hora da última modificação deste produto no Meus Pedidos (somente leitura).

Exemplo de produto em formato JSON:

```
{
  "id": 1001,
  "codigo": "485AR",
  "nome": "Arroz Pacote",
  "comissao": 5.00,
  "preco_tabela": 2.90,
  "ipi": 7.00,
  "tipo_ipi": "P",
  "st": null,
  "grade_cores": ["Azul", "Verde"],
  "grade_tamANHos": null,
  "moeda": "0",
  "unidade": "Kg",
  "saldo_estoque": 19.00,
  "observacoes": "Arroz Parboilizado",
  "excluido": false,
  "ultima_alteracao": "2014-01-15 02:12:43"
}
```

### 5.3. Tabelas de Preço

Representa as demais tabelas de preço disponíveis, além da tabela padrão. Cada produto pode ter um preço diferente em cada tabela de preço. Existem 3 tipos de tabelas:

- Preço Livre: o preço dos produtos é informado livremente através da API.
  - Acréscimo: o preço dos produtos é calculado automaticamente, aplicando um acréscimo sobre o preço padrão (cadastrado na entidade Produto).
  - Desconto: o preço dos produtos é calculado automaticamente, aplicando um desconto sobre o preço padrão (cadastrado na entidade Produto).
- URL Raíz: [http://meuspedidos.com.br/api/v1/tabelas\\_preco/](http://meuspedidos.com.br/api/v1/tabelas_preco/)
  - Métodos: GET, GET(id), POST, PUT(id)
  - Campos Obrigatórios: nome, tipo
  - Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(100)	
tipo	String(2)	Tipo da tabela. Deve ser "P" (preço livre), "A" (acréscimo) ou "D" (desconto).
acrescimo	Float	Acréscimo a ser aplicado nas tabelas do tipo "A". Informe <i>null</i> , caso não seja do tipo "A".
desconto	Float	Desconto a ser aplicado nas tabelas do tipo "D". Informe <i>null</i> , caso não seja do tipo "D".
excluido	Boolean	Indica se a tabela de preço está excluída.
ultima_alteracao	DateTime	Data e hora da última modificação desta tabela de preço no Meus Pedidos (somente leitura).

Exemplo de tabela de preço em formato JSON:

```
{
  "id": 15638,
  "nome": "Acréscimo de 15%",
  "tipo": "A",
  " acrescimo": 15,
  "desconto": null
  "excluido": false,
  "ultima_alteracao": "2014-03-12 13:01:33",
}
```

## 5.4. Produtos <-> Tabelas de Preço

Esta entidade faz a associação entre os Produtos e as Tabelas de Preço. Representa o preço de um produto em determinada tabela de preço. Cada produto pode ter um preço diferente para cada tabela de preço cadastrada.

- URL Raíz: [http://meuspedidos.com.br/api/v1/produtos\\_tabela\\_preco/](http://meuspedidos.com.br/api/v1/produtos_tabela_preco/)
- Métodos: GET, GET(id), POST, PUT(id)
- Campos Obrigatórios: produto\_id, tabela\_id, preco
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
preco	Float	Preço do produto nesta tabela.
tabela_id	Integer	ID da Tabela de Preço. Este deve ser o ID cadastrado no Meus Pedidos, e não o ID do seu ERP.
produto_id	Integer	ID do Produto. Este deve ser o ID cadastrado no Meus Pedidos, e não o ID do seu ERP.
excluido	Boolean	Indica se o preço está excluído.
ultima_alteracao	DateTime	Data e hora da última modificação deste preço no Meus Pedidos (somente leitura).

Exemplo de produto <-> tabela preço em formato JSON:

```
{
  "id": 7824042,
  "tabela_id": 15637,
  "produto_id": 7181201,
  "preco": 25,
  "excluido": false,
  "ultima_alteracao": "2014-03-25 15:28:50",
}
```

## 5.5. Transportadoras

- URL Raíz: <http://meuspedidos.com.br/api/v1/transportadoras/>
- Métodos: GET, GET(id), POST, PUT(id)
- Campos Obrigatórios: nome
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(100)	
cidade	String(50)	
estado	String(2)	Sigla do Estado.
informacoes_adicionais	String(500)	Utilize para guardar quaisquer informações que não tenham campos específicos.
telefones	String	Lista de objetos com os telefones da transportadora.
excluido	Boolean	Indica se a transportadora está excluída.
ultima_alteracao	DateTime	Data e hora da última modificação desta transportadora no Meus Pedidos (somente leitura).

Exemplo de transportadora em formato JSON:

```
{
  "id": 987,
  "nome": "Transportadora Rapidex",
  "cidade": "Curitiba",
  "estado": "PR",
  "informacoes_adicionais": "Entrega apenas na região
Sudeste.",
  "telefones": [{
    "numero": "(47) 9876-5432"
  }, {
    "numero": "(11) 98765-4321"
  }],
  "excluido": false,
  "ultima_alteracao": "2014-01-15 02:12:43"
}
```

## 5.6. Usuários (Vendedores)

Representa os usuários que tem acesso ao Meus Pedidos. Somente podem ser cadastrados através da interface Web do Meus Pedidos. A recomendação é que seja criado 1 usuário para cada vendedor cadastrado no ERP do cliente. O mapeamento entre usuários do Meus Pedidos e vendedores do ERP pode ser feito através do email do vendedor.

- URL Raíz: <http://meuspedidos.com.br/api/v1/usuarios/>
- Métodos: GET, GET(id) - **somente leitura**
- Campos

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(100)	
email	String(75)	Deverá ser único.
telefone	String(15)	
administrador	Boolean	Indica se o usuário é Administrador do sistema.
excluido	Boolean	Indica se o usuário está excluído.
ultima_alteracao	DateTime	Data e hora da última modificação deste usuário no Meus Pedidos (somente leitura).

Exemplo de usuário em formato JSON:

```
{
  "id": 10,
  "nome": "Paulo da Silva",
  "email": "paulo@silva.com.br",
  "telefone": "(48) 9876-5432",
  "administrador": false,
  "excluido": false,
  "ultima_alteracao": "2014-01-15 02:12:43"
}
```

## 5.7. Condições de Pagamento

Aqui você poderá integrar suas condições de pagamento, por exemplo “30/60”, “30/60/90”, etc. Esta entidade é opcional. Caso você não integre esta informação, será utilizado um campo de texto livre na Condição de Pagamento dos pedidos. O vendedor poderá preencher qualquer valor.

- URL Raíz: [http://meuspedidos.com.br/api/v1/condicoes\\_pagamento/](http://meuspedidos.com.br/api/v1/condicoes_pagamento/)
- Métodos: GET, GET(id), POST, PUT(id)
- Campos Obrigatórios: nome
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(100)	
excluido	Boolean	Indica se a condição está excluída.
ultima_alteracao	DateTime	Data e hora da última modificação desta condição de pagamento no Meus Pedidos (somente leitura).

Exemplo de condição de pagamento em formato JSON:

```
{
  "id": 50,
  "nome": "30/60/90/120/150",
  "excluido": false,
  "ultima_alteracao": "2014-01-15 02:12:43"
}
```

## 5.8. Formas de Pagamento

Aqui você poderá integrar suas formas de pagamento, por exemplo “Boleto”, “Cheque”, etc. Esta entidade é opcional. Caso você não integre esta informação, o campo Forma de Pagamento não irá aparecer nos pedidos.

- URL Raíz: [http://meuspedidos.com.br/api/v1/formas\\_pagamento/](http://meuspedidos.com.br/api/v1/formas_pagamento/)
- Métodos: GET, GET(id), POST, PUT(id)
- Campos Obrigatórios: nome
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(100)	
excluido	Boolean	Indica se a forma está excluída.
ultima_alteracao	DateTime	Data e hora da última modificação desta forma de pagamento no Meus Pedidos (somente leitura).

Exemplo de forma de pagamento em formato JSON:

```
{
  "id": 50,
  "nome": "30/60/90/120/150",
  "excluido": false,
  "ultima_alteracao": "2014-01-15 02:12:43"
}
```

## 5.9. Pedidos

A entidade Pedidos engloba tanto pedidos realizados quanto orçamentos pendentes. O que diferencia um orçamento de um pedido efetivado é a propriedade **status**.

- URL Raíz: <http://meuspedidos.com.br/api/v1/pedidos/>
- Métodos: GET, GET(id)
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
cliente_id	Integer	Identificador único do cliente.
transportadora_id	Integer	Identificador único da transportadora (opcional)
criador_id	Integer	Identificador único do vendedor que fez o pedido.
nome_contato	String(50)	Nome do contato do cliente.
status	String(1)	Status atual do pedido. 0=Cancelado, 1=Orçamento, 2=Pedido.
numero	Integer	Número auto-incremental do pedido.
total	Double	Valor total do pedido.
condicao_pagamento	String(100)	Condição de pagamento em formato texto livre. É usado apenas <b>caso você não integre</b> a entidade Condições da Pagamento.
condicao_pagamento_id	Integer	Identificador único da condição de pagamento. É usado apenas <b>caso você integre</b> a entidade Condições de Pagamento (opcional)
forma_pagamento_id	Integer	Identificador único da forma de pagamento. É usado apenas <b>caso você integre</b> a entidade Formas de Pagamento. (opcional)
data_emissao	Date	Data de emissão do pedido. Pode ser <i>null</i> .
observacoes	String(500)	Informações adicionais que o vendedor registrou no pedido.

Nome	Tipo	Observações
itens	Lista	Contém uma lista de objetos Item do Pedido, representando os itens vendidos no pedido (veja abaixo).
extras	Lista	Contém uma lista de objetos Campo Extra do Pedido (veja abaixo).
ultima_alteracao	DateTime	Data e hora da última modificação deste pedido no Meus Pedidos (somente leitura).

## Itens do Pedido

Todo pedido possui uma propriedade **itens**, que representa os itens vendidos. Segue a descrição desta sub-entidade:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
produto_id	Integer	ID do produto no Meus Pedidos.
tabela_preco_id	Integer	ID da tabela de preço no Meus Pedidos. Caso tenha sido utilizado o preço padrão (campo "preco_tabela" da entidade Produto), a tabela_preco_id será <i>null</i> .
quantidade	Double	Quantidade vendida.
preco_liquido	Double	Preço de venda do produto.
cotacao_moeda	Double	Cotação da moeda, em caso de venda em moeda estrangeira.
quantidade_grades		Lista com as grades de cores e tamanhos usadas, e suas respectivas quantidades. Caso o produto possua somente Cor, o Tamanho será <i>null</i> , e vice-versa. Veja exemplo abaixo. A soma das quantidades nesta lista será igual ao campo <b>quantidade</b> do item.
descontos		Lista com os descontos aplicados sobre o item, em formato Double.

Nome	Tipo	Observações
observacoes	String(500)	Informações adicionais que o vendedor registrou no item.

### Campos Extras do Pedido

É possível configurar campos dinâmicos para os pedidos, para coletar informações que o MeusPedidos não fornece no sistema padrão. É possível configurar estes campos através da API ou pelo sistema Web, por favor verifique na próxima seção.

Nome	Tipo	Observações
campo_extra_id	Integer	Identificador único do campo extra.
nome	String(50)	Nome do campo extra.
valor	String(500)	Valor do campo extra.

Exemplo de pedido em formato JSON:

```
{
  "id": 8821,
  "cliente_id": 123,
  "transportadora_id": 10,
  "vendedor_id": 457,
  "nome_contato": "João da Silva",
  "status": "2",
  "numero": 223,
  "total": 1234.56
  "data_emissao": 2013-05-25,
  "condicao_pagamento": "30/60/90",
  "condicao_pagamento_id": 65,
  "forma_pagamento_id": null,
  "observacoes": "Favor faturar urgente.",
  "itens": [
    {
      "id": 10291,
      "produto_id": 21921,
      "tabela_preco_id": 85719,
      "quantidade": 10.00,
      "quantidade_grades": [{
        "cor": "Azul",
        "tamanho": null,
        "quantidade": 10.00
      }],
      "preco_liquido": 89.90,
      "subtotal": 899.00,
      "cotacao_moeda": null,
      "descontos": [ 10.50, 5.00],
      "observacoes": "Cor azul"
    }
  ],
  "extras": [{
    "id": 928,
    "campo_extra_id": 192,
    "valor": "2013-05-12"
  }],
  "ultima_alteracao": "2014-01-15 02:12:43"
}
```

## 5.10. Campos Extras

O Meus Pedidos é bastante flexível e trabalha com o conceito de campos extras no pedido. Sabemos que cada aplicação possui necessidades específicas, e diversas outras informações que precisam ser coletadas em campo. Ao invés de criar dezenas de campos diferentes no Meus Pedidos, nós optamos por permitir campos extras configuráveis.

Você poderá utilizar este recurso para coletar qualquer informação relevante, como por exemplo: Data de Entrega, Valor do Frete, etc. Existem 2 tipos de campo: Texto Livre e Data.

- URL Raíz: [http://meuspedidos.com.br/api/v1/campos\\_extras/](http://meuspedidos.com.br/api/v1/campos_extras/)
- Métodos: GET, GET(id), POST, PUT(id)
- Campos Obrigatórios: nome, tipo
- Campos:

Nome	Tipo	Observações
id	Integer	Identificador único (somente leitura)
nome	String(50)	
tipo	String(1)	Tipo do campo. Pode ser 0=Campo Texto Livre, 1=Campo Data.
excluido	Boolean	Indica se a condição está excluída.
ultima_alteracao	DateTime	Data e hora da última modificação deste campo extra no Meus Pedidos (somente leitura).

Exemplo de campo extra em formato JSON:

```
{
  "id": 99,
  "nome": "Data de Entrega",
  "tipo": "1",
  "excluido": false,
  "ultima_alteracao": "2013-10-01 18:04:25"
}
```

## 6. Histórico de Revisões

Versão	Data	Descrição
0.1	27/01/2014	Início da criação do Manual.
0.2	28/02/2014	Diversas alterações: <ul style="list-style-type: none"><li>- Nova seção <b>Sincronizando Dados na Prática</b></li><li>- Inclusão do campo <b>ultima_alteracao</b> em todas as Entidades</li><li>- Clientes: alteração no formato dos campos <b>emails</b>, <b>telefones</b> e <b>contatos</b>.</li><li>- Produtos: inclusão dos campos <b>tipo_ipi</b>, <b>st</b>, <b>grade_cores</b> e <b>grade_tamANHos</b>.</li><li>- Transportadoras: alteração no formato do campo <b>telefones</b>.</li><li>- Usuários: esta Entidade agora é somente leitura (não é possível cadastrar nem alterar pela API).</li><li>- Pedidos: alteração no formato dos campos <b>itens</b> e <b>extras</b>.</li><li>- Itens do Pedido: inclusão do campo <b>quantidade_grades</b>. Alteração no formato do campo <b>descontos</b>.</li><li>- Campos Extras: alteração do formato.</li></ul>
0.3	26/03/2014	<ul style="list-style-type: none"><li>- Nova entidade: <b>Tabelas de Preço</b>.</li><li>- Nova entidade: <b>Produtos &lt;-&gt; Tabelas de Preço</b>.</li><li>- Itens do Pedido: inclusão do campo <b>tabela_preco_id</b>, referenciando a tabela de preço utilizada na venda.</li><li>- Criada documentação dos objetos <b>Email</b>, <b>Telefone</b> e <b>Contato</b>.</li></ul>