Incorrect implementation of the OpenAPI spec

The OpenAPI specification file, `gerenciamento-de-vendas.YAML`, found at the Limber documentation website is inapropriate.

Broken strings

Values that have been specified as strings should be enclosed within quotes:

```
Invalid YAML from provided source: while scanning a plain scalar
  in "<byte string>", line 1338, column 60
found unexpected ':'
  in "<byte string>", line 1338, column 62
found
  in "<byte string>", line 1375, column 60
found unexpected ':'
  in "<byte string>", line 1375, column 62
```

Line 1338 is `api.vendas.venda.items.{idVenda}.get.responses.receitas[].horario_str`, line 1375 is `api.vendas.venda.items.{idVenda}.get.responses.receitasAgrupadas[].horario_str`.

Incorrect types

The following is the output when trying to parse the file with an OpenAPI spec validator:

```
Failed to parse OpenAPI document

20 validation errors for OpenAPI
paths./api/vendas.get.responses.200.other.content.application/json.schema.other.pr
operties.data_hora_inc.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='Date', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/vendas.get.responses.200.other.content.application/json.schema.other.pr
operties.data_hora_inc.other.type.list[str-enum[DataType]]
  Input should be a valid list [type=list_type, input_value='Date',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/vendas.get.responses.200.other.content.application/json.schema.other.pr
operties.pedido_de_cancelamento.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='Boolean', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/vendas.get.responses.200.other.content.application/json.schema.other.pr
operties.pedido_de_cancelamento.other.type.list[str-enum[DataType]]
  Input should be a valid list [type=list_type, input_value='Boolean',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
```

```
paths./api/vendas.get.responses.200.other.content.application/json.schema.other.pr
operties.importante.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='Boolean', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/vendas.get.responses.200.other.content.application/json.schema.other.pr
operties.importante.other.type.list[str-enum[DataType]]
  Input should be a valid list [type=list_type, input_value='Boolean',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.parameters.2.other.schema.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='dateTime', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.parameters.2.other.schema.other.type.list[str-
enum[DataType]]
  Input should be a valid list [type=list_type, input_value='dateTime',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.parameters.3.other.schema.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='dateTime', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.parameters.3.other.schema.other.type.list[str-
enum[DataType]]
  Input should be a valid list [type=list_type, input_value='dateTime',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.parameters.4.other.schema.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='dateTime', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.parameters.4.other.schema.other.type.list[str-
enum[DataType]]
  Input should be a valid list [type=list_type, input_value='dateTime',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.parameters.5.other.schema.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='dateTime', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.parameters.5.other.schema.other.type.list[str-
enum[DataType]]
  Input should be a valid list [type=list_type, input_value='dateTime',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.parameters.8.other.schema.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='dateTime', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.parameters.8.other.schema.other.type.list[str-
enum[DataType]]
  Input should be a valid list [type=list_type, input_value='dateTime',
input_type=str]
```

```
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.parameters.9.other.schema.other.type.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='dateTime', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.parameters.9.other.schema.other.type.list[str-
enum[DataType]]
  Input should be a valid list [type=list_type, input_value='dateTime',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
paths./api/ec/clientes.get.responses.200.other.content.application/json.schema.oth
er.properties.termos.other.items.other.properties.data_hora_aceite_termo.other.typ
e.str-enum[DataType]
  Input should be 'string', 'number', 'integer', 'boolean', 'array', 'object' or
'null' [type=enum, input_value='timestamp', input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/enum
paths./api/ec/clientes.get.responses.200.other.content.application/json.schema.oth
er.properties.termos.other.items.other.properties.data_hora_aceite_termo.other.typ
e.list[str-enum[DataType]]
  Input should be a valid list [type=list_type, input_value='timestamp',
input_type=str]
    For further information visit https://errors.pydantic.dev/2.11/v/list_type
```

`data_hora_inc` property

As described by the OpenAPI 3.0 documentation, date data types should be specified under the `format` key.

The `Date` type used is not in accordance with the standart (i.e. it does not exist).

Since `data_hora_inc`'s example follows the RFC 3339, section 5.6 notation, a `format: date-time` should be specified.

Here is what `data_hora_inc` should look like, in accordance with the OpenAPI 3.0 specs and the aforementioned recommendations:

```
data_hora_inc:
    type: string
    format: date-time
    description: Data e hora do registro da venda UTC
```

`pedido_de_cancelamento` property

The boolean type is lower case for OpenAPI 3.0. `pedido_de_cancelamento` (line 287) has its type specified with an upper case initial (`Boolean`).

This is what it should look like:

```
pedido_de_cancelamento:
    type: boolean
    description: Se a venda tem algum item com pedido de cancelamento
```

Additionally, we have found that your description contains the `pedido_de_cancelamento` twice in the response table - once with the correct `boolean`, and another with the incorrect spelling, both having different descriptions.

The same issue is present in the definition for `importante` (line 299).

`dateTime` type

`dateTime` is not a valid type for the OpenAPI specification.

As previously stated, a `string` type with `format: date-time` should be used for datetimes. That is, assuming the formatting is consistent with RFC 3339, section 5.6.

If it is not to the standard datetime notation, we'd like to request a `pattern` keyword to be added to the schema of properties with the `date-time` format. The use of Python's date format codes for declaring such pattern would be greatly appreciated.

`dateTime` "types" were used in the request properties for the `/clientes` endpoint.

`/vendas` endpoint `requestBody`

The date request properties for `/vendas` are correctly set to `string`. They do not, however, contain a `format` nor a `pattern` in their schema.

Please specify a `format` and `pattern` in accordance with the former request.

Here is an example of a date property from `/vendas`'s `requersBody` where `format` and `pattern` schema properties aren't documented (Line 151):

```
dataVendaInit:
    type: string
    description: Data início para pesquisa
```

`clientes.data_hora_aceite_termo`

`clientes.data_hora_aceite_termo` has `type: timestamp`, which also does not exist in the OpenAPI 3.0 spec. The example shows it as a `date-time` with the same formatting as `data_hora_inc`, meaning the same remarks for the latter apply here.

Other formatting issues

CEP

Line 1466 has `"CEP"` in quotes. It shouldn't be in quotes.

`statusArray` and `formas_pag_desc`

The definition of these properties are missing the `items` schema key:

```
    statusArray:
        type: array
        description: Status da venda
```

The example for `formas_pag_desc` (line 367) is also inconsistent with the property's schema.

Lack of parameter templating

Incorrect path templating for (path parameters do not match with path) happen with the following:

```
GET /api/vendas/venda/{idVenda}
GET /api/vendas/venda/cancelamentos/{idVenda}
GET /api/vendas/venda/pagamentos/{idVenda}
GET /api/vendas/venda/itens/{idVenda}
```

They should each have a schema for the `idVenda` parameter:

```
parameters:
    -   name: idVenda
        in: path
        schema:
            type: number
        required: true
        description: ID da venda (codigo)
```