

Envio de Laudos Veros

Como é feito os envios dos laudos para o Veros.

Para que os **envios dos laudos** que foram **solicitados** pelo Veros sejam enviados pelo Provet, é necessário que tenhamos as seguintes etapas configuradas:

1. Criar uma pasta chamada "**Veros**" para receber os laudos no C:\Vertis.
C:\Vertis\Veros

2. Inserir no arquivo **vertisconnect.cfg** com o seguinte bloco:

[VEROS]

Tempo=3500

FilePathPDF=C:\Vertis\Veros

3. Instalar o serviço EnvioVerosSVC no servidor.

Devemos **instalar** o serviço EnvioVerosSVC no servidor do cliente, e ele se encontra no caminho:

||192.168.200.4|gpi|Producao|Atualização Clientes|Serviços|SVC Envio Veros

Devemos copiar o arquivo **EnvioVerosSVC.exe** para a pasta Vertis do servidor do cliente e criar um serviço desse arquivo na máquina local.

Depois iremos executar o **CMD** como **Administrador** e digitar o seguinte comando, dentro da pasta onde o .exe se encontra.

EnvioVerosSVC.exe -install

4. Incluir os endpoints na tabela **link_api** para as unidades do cliente.

Edit Data - SRV_PROVET (10.0.18.20:58432) - vertis - link_api

	cod_api [PK] serial	ind_tipo_api character varying(20)	link_api character varying(250)	path_pasta_ character varying(250)	cod_unid_ne integer	cod_unid_op integer	ind_situacao character(1)	usuario character varying(50)	senha character varying(50)	ind_dev_mod character varying(10)
1	123	ENVIO_VEROS	https://portal.veros.vet/api2/provet	''	211	1	A		''	N
2	124	ENVIO_VEROS	https://portal.veros.vet/api2/provet	''	211	2	A		''	N
3	125	ENVIO_VEROS	https://portal.veros.vet/api2/provet	''	211	3	A		''	N
4	126	ENVIO_VEROS	https://portal.veros.vet/api2/provet	''	211	4	A		''	N

link_api - Endpoint disponibilizado pelo Veros.

ind_situacao - "A"

5. Incluir as origens na tabela **integrador_origem**

Edit Data - SRV_PROVET (10.0.18.20:58432) - vertis - integrador_origem

	id_integrador [PK] integer	desc_item character varying(50)	desc_sub_item character varying(50)	situacao character(1)	ind_prioridad integer	tipo_envio character(1)	cod_unid_ne integer	cod_unid_op integer	tipo_origem character(2)	fkid_mod_crt integer	hora_inicial time without	hora_final time without
1	41	Integracao VEROS	Envio de Resultados - VEROS	S	2	I	211	1	VR		00:00:00	23:59:59
2	42	Integracao VEROS	Envio de Resultados - VEROS	S	2	I	211	2	VR		00:00:00	23:59:59
3	43	Integracao VEROS	Envio de Resultados - VEROS	S	2	I	211	3	VR		00:00:00	23:59:59
4	44	Integracao VEROS	Envio de Resultados - VEROS	S	2	I	211	4	VR		00:00:00	23:59:59

tipo_origem - "VR"

situacao - "S"

-- .: Select Publicação Laudo | Veros

```
Select org.cod_unid_oper, uop.nom_unid_oper, rev.nom_animal,
rev.nom_proprietario,
    case when rev.ind_origem = 'VRT' then 'Vertis' when rev.ind_origem =
'VRO' then 'Veros(API)' else 'Outros' end as origem_os,
    etr.cod_req_exame, etr.seq_exa_requisit, erq.cod_produto,
pro.nom_prod_compra, tar.id_integrador_tarefa,
    case when tar.ind_situacao = 'E' then 'Enviado' when tar.ind_situacao =
'A' then 'Fila' else 'Outros' end as situacao_envio_laudo,
    id_ord_serv_solic, seq_exa_req_solic, erq.dth_publicacao
from integrador_tarefa_envio_laudo_terceiros etr
    join integrador_tarefa tar on etr.id_int_tarefa =
tar.id_integrador_tarefa
    join integrador_origem org on org.id_integrador_origem =
tar.id_origem
    left join req_exame_vet rev on rev.cod_req_exame =
etr.cod_req_exame
    left join exame_requisitado erq on erq.seq_exa_requisit =
etr.seq_exa_requisit
    left join unid_operacional uop on uop.cod_unid_oper =
org.cod_unid_oper
    left join produto pro on pro.cod_produto = erq.cod_produto
where org.tipo_origem = 'VR'
order by etr.cod_req_exame desc , etr.seq_exa_requisit
```

Para que o fluxo da integração com o Veros ocorra, é preciso que o Veros solicite algum exame a unidade que for receber esse pedido via API, onde é passado via formato **JSON** os dados do tutor, animal e exames da o.s, dessa forma:

JSON

```
{
  "itens_os": [
    {
      "id_ctrl": "182067",
      "exame": "HEMO - HEMOGRAMA ",
      "dth_lancamento": "2025-09-09 00:00:00",
      "cod_produto": "21",
      "qtd_produto": 1,
      "seq_exa_req_solic": "125966"
    }
  ],
  "cod_unid_negoc": 211,
  "cod_unid_oper": 1,
  "ind_liquidante": "C",
  "tip_faturamento": "V",
  "cnpj_cpf_liquidante": "40691823000125",
  "obs_ordem_servico": "",
  "id_ord_serv_solic": "81877",
  "dth_inclusao": "2025-09-09 19:02:10",
  "ind_retira_amostra": "S",
  "os_med": {
    "cnpj_clinica": "40691823000125",
    "veterinario": {
      "codigo": 1234,
      "nome": "ISABELLA ZEQUE MACEDO",
      "crm_v": "60584",
      "crm_v_uf": "SP",
      "cpf_cnpj": "",
      "email": "",
      "telefone": ""
    },
    "tutor": {
      "cadastra": "N",
      "codigo": "0",
      "nome": "THAISA CARVALHO",
      "cpf_cnpj": "",
      "email": "",
      "telefone": "",
      "cep": "",
      "nro_logradouro": "",
      "compl_logradouro": ""
    },
    "animal": {
      "cadastra": "N",
      "codigo": 0,
      "nome": "MIKE CARVALHO",
      "cod_especie": "1",
      "cod_raca": "24",
      "dat_aniversario": "01/01/2015",
      "ind_sexo": "M",
      "ind_castrado": "S"
    }
  }
}
```

O Veros solicita via API passando os dados do tutor, animal e os exames contidos nesse pedido, e a Vertis API recebe esse JSON.

Campos **obrigatórios** de parceiro são:

Animal

Nome do animal - `nome`
Sexo do animal - `ind_sexo`
Data nascimento - `dat_aniversario`
Castrado ou não - `ind_castrado`
Código da Espécie - `cod_especie`
Código da Raça - `cod_raca`

Tutor

Nome do tutor - `nome`

Clínica

CNPJ - `cnpj_clinica`

Veterinário

Nome do veterinário - `nome`
CRMV - `crm`
UF CRMV - `uf`

Campos **obrigatórios** gerais são:

Ordem de Serviço

Código do produto - `cod_produto`
Quantidade - `qtd_produto`
Cód.Sequencial do exame - `seq_exa_req_solic`
Cód OS Veros - `id_ord_serv_solic`

Unidade Operacional

`cod_unid_negoc`
`cod_unid_oper`

Ao receber o pedido com os campos [devidamente preenchidos](#), a Vertis API faz a gravação desses dados nas tabelas necessárias (exame_requisitado, req_exame_vet, sistema_log) e também a O.S é criada no Vertis com os dados recebidos.

Todas as O.S que são abertas pela integração com o Veros entram como “[Solicitado](#)” na situação, até o momento da entrega das amostras para análises, quando a amostra [chega](#) até a unidade, é [confirmado o recebimento](#) pela tela do Mapa de Trabalho, em seguida é encaminhada para análise.

Após a amostra ser analisada, entra na fase de [digitação do laudo](#), e após o exame ser laudado e finalizado o [Gerenciador de PDF](#) do Vertis entra em ação, quando o laudo é [convertido para PDF](#), sua situação é alterada para “[Emitido](#)”, aonde o [Publicador de Laudos](#) do Vertis prepara sua publicação para o Veros, pela coluna [ind_origem](#) da tabela [integrador_tarefa](#), quando o ind_situacao estiver = “A” de Aguardando, o publicador entende que irá para a [fila](#) de publicação, e após ser executada sua publicação, essa coluna é alterada para “E” de [Emitido](#), e no Vertis se encontra como Publicado.

id_integrador [PK] serial	ind_prioridad integer	dth_cadastro timestamp w	ind_situacao character varying(1)	dth_execuca timestamp w	dth_cancelan timestamp w	cod_usu_can integer	cod_parceiro integer	id_origem integer
12499	2	2025-09-10	A					41
12500	2	2025-09-10	A					41
12501	2	2025-09-10	A					41
12502	2	2025-09-10	A					41
12503	2	2025-09-10	A					41
12504	2	2025-09-10	A					41
12505	2	2025-09-10	E	2025-09-12				41
12506	2	2025-09-10	E	2025-09-12				41
12507	2	2025-09-10	E	2025-09-12				41
12508	2	2025-09-10	E	2025-09-12				41
12509	2	2025-09-10	E	2025-09-12				41
12510	2	2025-09-10	E	2025-09-12				41

Na tabela [integrador_tarefa](#), filtrando pela [PK](#) do [id_origem](#) que foi cadastrada na nossa [5ª](#) etapa deste manual, vemos os laudos que foram [publicados](#) para o Veros, conseqüentemente suas situações.

Fluxograma da Integração Vertis X Veros.

