

SKYTEF

soluções em captura de transações

Descrição da Solução NAVS

(Navegador Skytef)

SKYTEF

Versão 2.3.2.1 – 23/06/2023

Para terminais PAX modelos S920, D195 e Q92

Sumário

1.0	Introdução	6
2.0	Descrição Inicial	7
3.0	Lista de códigos de status HTTP	7
3.1	1xx Informativa	7
3.1.1	100 Continuar	7
3.1.2	101 Mudando protocolos	8
3.1.3	102 Processamento (WebDAV) (RFC 2518)	8
3.1.4	122 Pedido-URI muito longo	8
3.2	2xx Sucesso	8
3.2.1	200 OK	8
3.2.2	201 Criado	8
3.2.3	202 Aceito	8
3.2.4	203 não-autorizado (desde HTTP/1.1)	8
3.2.5	204 Nenhum conteúdo	9
3.2.6	205 Reset	9
3.2.7	206 Conteúdo parcial	9
3.2.8	207-Status Multi (WebDAV) (RFC 4918)	9
3.3	3xx Redirecionamento	9
3.3.1	300 Múltipla escolha	9
3.3.2	301 Movido	9
3.3.3	302 Encontrado	9
3.3.4	304 Não modificado	10
3.3.5	305 Use Proxy (desde HTTP/1.1)	10
3.3.6	306 Proxy Switch	10
3.3.7	307 Redirecionamento temporário (desde HTTP/1.1)	10
3.4	4xx Erro de cliente	10
3.4.1	400 Requisição inválida	10
3.4.2	401 Não autorizado	11
3.4.3	402 Pagamento necessário	11
3.4.4	403 Proibido	11
3.4.5	404 Não encontrado	11
3.4.6	405 Método não permitido	11

Especificação de Interface

3.4.7	406 Não Aceitável	11
3.4.8	407 Autenticação de proxy necessária	11
3.4.9	408 Tempo de requisição esgotou (Timeout)	11
3.4.10	409 Conflito	12
3.4.11	410 Gone	12
3.4.12	411 comprimento necessário	12
3.4.13	412 Pré-condição falhou	12
3.4.14	413 Entidade de solicitação muito grande	12
3.4.15	414 Pedido-URI Too Long	12
3.4.16	415 Tipo de mídia não suportado	12
3.4.17	416 Solicitada de Faixa Não Satisfatória	12
3.4.18	417 Falha na expectativa	12
3.4.19	418 Eu sou um bule de chá	13
3.4.20	422 Entidade improcessável (WebDAV) (RFC 4918)	13
3.4.21	423 Fechado (WebDAV) (RFC 4918)	13
3.4.22	424 Falha de Dependência (WebDAV) (RFC 4918)	13
3.4.23	425 coleção não ordenada (RFC 3648)	13
3.4.24	426 Upgrade Obrigatório (RFC 2817)	13
3.4.25	450 bloqueados pelo Controle de Pais do Windows	13
3.4.26	499 cliente fechou Pedido (utilizado em ERPs/VPSA)	13
3.5	5xx outros erros	14
3.5.1	500 Erro interno do servidor (Internal Server Error)	14
3.5.2	501 Não implementado (Not implemented)	14
3.5.3	502 Bad Gateway	14
3.5.4	503 Serviço indisponível (Service Unavailable)	14
3.5.5	504 Gateway Time-Out	15
3.5.6	505 HTTP Version not supported	15
3.6	6xx outros erros	15
4.0	Inicialização	16
5.0	Configuração	16
5.1	Configuração do tipo de conexão	16
5.2	Configuração de CHiPs	16
5.3	Configuração de WiFi	18
6.0	Carga da aplicação no terminal e arquivos de atualização (extensão AUP)	18
7.0	Carga de imagens e outros arquivos para uso do NAVS	25
8.0	Mifare: gravação e leitura de dados	27
9.0	Detalhamento dos Comandos	28
9.1	LOCAL	29
9.2	DATA	29
9.3	LUA	29
9.4	CANCEL_KEY	30
9.5	POST	30
9.6	RESET ou REBOOT	31

Especificação de Interface

9.7	PRINTER	31
9.7.1	Configuração hierárquica do comando PRINTER	32
9.8	TONE	32
9.9	DELAY	33
9.10	PRNLOGO	33
9.11	CONLOGO	34
9.12	DEVICE	34
9.13	FTP	35
9.14	UNZIP	36
9.15	CONSOLE	36
9.15.1	Configuração hierárquica do comando CONSOLE	38
9.16	INIT	38
9.17	RECTANGLE	38
9.18	LAUNCH_ITEM	39
9.19	CAPTURE	39
9.20	WRITE_AT (deprecado)	40
9.21	RELICENSE	40
9.22	Web Services - Comandos WS_GET, WS_POST e WS	41
9.23	DELFILE	42
9.24	CHGCONFNT	42
9.25	FILE	42
9.26	SLEEP	44
9.27	GPRS_SWITCH_SIM	44
9.28	SELECT	44
9.29	PRNFEED	46
9.30	GENERATE_QR_CODE	46
9.31	CODE128	47
9.32	CODE_BAR_3_9	47
9.33	CODEBAR	48
9.34	CODE_BAR_2_5	48
9.35	CTLSWRITE	48
9.36	ANIMATE	49
9.37	EMV_BC	49
9.38	NAVS_UPDATE	62

Especificação de Interface

9.39	EMV_ID	62
9.40	PAGS	63
9.41	GET TYPE	66
9.41.1	CARD	66
9.41.2	MODEL	67
9.41.3	SERIALNO	67
9.41.4	STSCOMM - Deprecado - usar COMM_STATUS	67
9.41.5	COMM_STATUS	68
9.41.6	HIDDEN	68
9.41.7	VAR	68
9.41.8	CANCEL_KEY	71
9.41.9	FIELD	71
9.41.10	VALUE	73
9.41.11	IDCTLS ou IDMIFARE	73
9.41.12	INSERTCARD	73
9.41.13	KEY	74
9.41.14	WIFI_LIST	74
9.41.15	ANYKEY - Deprecado - Usar KEY	74
9.41.16	SPECIALKEY - Deprecado - Usar KEY	75
9.41.17	VERSION	75
9.41.18	DESNBS	75
9.41.19	PASS	76
9.41.20	EMV	76
9.41.21	TOUCH	77
9.41.22	LAUNCH	77
9.41.23	HOUR	77
9.41.24	DATETIME	78
9.41.25	CPF	78
9.41.26	CNPJ	78
9.41.27	BATTERY_STATUS	79
9.41.28	IMEI	79
9.41.29	ICCID	79
9.41.30	IMSI	80
9.41.31	SSLKEYS	80
9.41.32	WIFIMAC	81
9.41.33	MD5	81
9.41.34	OSVERSION	81
9.41.35	LAC	81
9.41.36	CID	81
9.41.37	SIGNAL_LEVEL	82
9.41.38	DATE ou DATA	82
9.41.39	NONINT	83
9.41.40	TIMEOUT	83
9.41.41	SWEVERSION	84
9.41.42	SWEPSCVERSION	84
9.41.43	BCVERSION	84
9.41.44	LIBLOLVERSION	84
9.41.45	LIBPAYVERSION	84
9.41.46	SIM_SLOTS	85
9.41.47	MIFARE_READ	85
9.42	SET TYPE	85
9.42.1	VAR	85

Especificação de Interface

9.42.2	CANCEL_KEY	86
9.42.3	PRINTER	86
9.42.4	CONSOLE	86
9.42.5	TIME	86
10.0	Programação por eventos	87
11.0	Histórico das versões	88

1.0 Introdução

NAVS é a plataforma de desenvolvimento da Skytef que proporciona às redes de POS flexibilidade nas aplicações. Possibilita rápido desenvolvimento de novas funcionalidades e serviços, atendendo aos mais diversos mercados como redes adquirentes, *private label*, bancos, financeiras, correspondente bancário, fidelidade, entre outros.

NAVS é um navegador que permite que o POS acesse o servidor, utilizando o mesmo tipo de conexão das transações. O acesso ao servidor possibilita realizar atualizações das aplicações e implementar novos serviços, o que pode ser feito em toda rede ou em um único terminal, de acordo com a necessidade do gestor.

Vantagens

- Reduz tempo nos processos de certificação de aplicações
- Transforma a rede de POS numa rede de serviços
- Possibilita ativação/desativação de serviços nos terminais POS remotamente
- Facilita o desenvolvimento, pois é uma plataforma amigável e intuitiva

2.0 Descrição Inicial

NAVS é uma aplicação que será carregada no POS e interpretará comandos recebidos do Servidor.

Na carga inicial, deverá existir um arquivo XML contendo as informações da primeira comunicação do POS para o Servidor, como uma inicialização.

Na resposta, o Servidor deverá enviar os comandos XML para terminar a inicialização e iniciar a primeira transação.

O POS capturará os dados desta transação e enviará ao Servidor que deverá responder com os comandos para terminar a transação e iniciar a próxima.

IMPORTANTE:

O NAVS utiliza o protocolo HTTP ou HTTPS.

O campo mandatório que deve ser enviado pelo Host no protocolo HTTP é o "Content_Lenght".

Abaixo os principais erros do protocolo HTTP, conforme a Wikipedia:

(https://pt.wikipedia.org/wiki/Lista_de_c%C3%B3digos_de_estado_HTTP).

3.0 Lista de códigos de status HTTP

A seguir está uma lista de códigos de resposta em HTTP (HyperText Transfer Protocol). Isso inclui os códigos padrões de internet da IETF, outras especificações e alguns códigos adicionais usados. O primeiro dígito do código de status indica uma das cinco classes de resposta, o mínimo necessário para um cliente HTTP é que ele reconheça essas cinco classes. Microsoft IIS pode usar sub-códigos decimais adicionais específicos para fornecer mais informações, mas estes não estão listados aqui. As frases utilizadas são os exemplos padrão, mas qualquer alternativa humana legível pode ser fornecida. Salvo disposição em contrário, o código de status é parte do padrão HTTP/1.1.

3.1 *1xx Informativa*

Solicitação recebida, dando continuidade ao processo. Essa classe de códigos de status indica uma resposta provisória, constituída apenas do Status-Line e cabeçalhos opcionais, e é encerrado por uma linha vazia. Desde a versão HTTP/1.0 não são definidos todos os códigos de status 1xx, os servidores não devem enviar uma resposta 1xx para um cliente HTTP/1.0, exceto sob condições experimentais.

3.1.1 100 Continuar

Isso significa que o servidor recebeu os cabeçalhos da solicitação, e que o cliente deve proceder para enviar o corpo do pedido (no caso de haver um pedido um corpo deve ser enviado, por exemplo, um POST pedido). Se o corpo é grande o pedido, enviando-os para um servidor, quando o pedido já foi rejeitada com base em cabeçalhos inadequado é ineficiente. Para ter um cheque do servidor se o pedido pode ser aceite com base no pedido de cabeçalhos sozinho, o cliente deve enviar Esperar: 100-continue como um cabeçalho no seu pedido

Especificação de Interface

inicial e verifique se a 100 Continuar código de status é recebido em resposta antes de permanente (ou receber 417 Falha na expectativa e não continuar).

3.1.2 101 Mudando protocolos

Isso significa que o solicitante pediu ao servidor para mudar os protocolos e o servidor está reconhecendo que irá fazê-lo.

3.1.3 102 Processamento (WebDAV) ([RFC 2518](#))

Como uma solicitação WebDAV pode conter muitos sub-pedidos que envolvam operações de arquivo, pode demorar muito tempo para concluir o pedido. Este código indica que o servidor recebeu e está processando o pedido, mas nenhuma resposta ainda não está disponível. Isso impede que o cliente o tempo limite e supondo que o pedido foi perdido.

3.1.4 122 Pedido-URI muito longo

Este é um padrão IE7 somente código não significa que o URI é mais do que um máximo de 2083 caracteres. (Ver código 414).

3.2 2xx Sucesso

Esta classe de códigos de status indica a ação solicitada pelo cliente foi recebida, compreendida, aceita e processada com êxito.

3.2.1 200 OK

Padrão de resposta para solicitações HTTP sucesso. A resposta real dependerá do método de solicitação usado. Em uma solicitação GET, a resposta conterá uma entidade que corresponde ao recurso solicitado. Em uma solicitação POST a resposta conterá a descrição de uma entidade, ou contendo o resultado da ação.

3.2.2 201 Criado

O pedido foi cumprido e resultou em um novo recurso que está sendo criado.

3.2.3 202 Aceito

O pedido foi aceito para processamento, mas o tratamento não foi concluído. O pedido poderá ou não vir a ser posta em prática, pois pode ser anulado quando o processamento ocorre realmente.

3.2.4 203 não-autorizado (desde HTTP/1.1)

O servidor processou a solicitação com sucesso, mas está retornando informações que podem ser de outra fonte.

3.2.5 204 Nenhum conteúdo

O servidor processou a solicitação com sucesso, mas não é necessário nenhuma resposta.

3.2.6 205 Reset

O servidor processou a solicitação com sucesso, mas não está retornando nenhum conteúdo. Ao contrário da 204, esta resposta exige que o solicitante redefinir a exibição de documento.

3.2.7 206 Conteúdo parcial

O servidor está entregando apenas parte do recurso devido a um cabeçalho intervalo enviados pelo cliente. O cabeçalho do intervalo é usado por ferramentas como wget para permitir retomada de downloads interrompidos, ou dividir um download em vários fluxos simultâneos.

3.2.8 207-Status Multi (WebDAV) ([RFC 4918](#))

O corpo da mensagem que se segue é um XML da mensagem e pode conter um número de códigos de resposta individual, dependendo de quantas sub-pedidos foram feitos.

3.3 3xx Redirecionamento

O cliente deve tomar medidas adicionais para completar o pedido. Essa classe de código de status indica que a ação ainda precisa ser levado pelo agente do usuário, a fim de atender à solicitação. A ação necessária pode ser realizada pelo agente, sem interação com o usuário, se e somente se o método utilizado no segundo pedido é GET ou HEAD. Um agente do usuário não deve redirecionar automaticamente uma solicitação de mais de cinco vezes, uma vez que tais redirecionamentos geralmente indicam um loop infinito .

3.3.1 300 Múltipla escolha

Indica várias opções para o recurso que o cliente pode acompanhar. É, por exemplo, poderia ser usado para apresentar opções de formato diferente para o vídeo, arquivos de lista com diferentes extensões, ou desambiguação sentido da palavra.

3.3.2 301 Movido

Esta e todas as solicitações futuras devem ser direcionada para o URI.

3.3.3 302 Encontrado

Este é um exemplo de boas práticas industriais contradizendo a norma. Especificação HTTP/1.0 ([RFC 1945](#)) exigiu o cliente para executar um redirecionamento temporário (o que descreve frase original era "Movido

Especificação de Interface

Temporariamente”), mas os browsers populares executadas 302 com a funcionalidade de um 303 Consulte Outros. Por isso, acrescentou HTTP/1.1 códigos de status 303 e 307 a distinguir entre os dois comportamentos. No entanto, a maioria das aplicações Web e os quadros ainda usam o código de status 302 como se fosse o 303.

3.3.4 304 Não modificado

Indica que o recurso não foi modificado desde o último pedido. Normalmente, o cliente fornece um cabeçalho HTTP como o Se-Modificado-Desde cabeçalho para proporcionar um tempo contra o qual para comparar. Usando este poupa largura de banda e de reprocessamento no servidor e cliente, uma vez que apenas os dados do cabeçalho devem ser enviados e recebidos em comparação com a totalidade da página que está sendo reprocessados pelo servidor, em seguida, enviado novamente utilizando mais largura de banda do servidor e cliente.

3.3.5 305 Use Proxy (desde HTTP/1.1)

Muitos clientes HTTP (como o Mozilla e Internet Explorer) podem não tratar corretamente as respostas com este código de status, principalmente por razões de segurança.

3.3.6 306 Proxy Switch

Mudança de proxy. Deixou de ser usado.

3.3.7 307 Redirecionamento temporário (desde HTTP/1.1)

Nesta ocasião, o pedido deve ser repetido com outro URI, mas futuras solicitações ainda pode usar a URI original. Em contraste com a 303, o método de pedido não deve ser mudado quando a reedição do pedido original. Por exemplo, uma solicitação POST deve ser repetido com outro pedido POST.

3.4 4xx Erro de cliente

A classe 4xx de código de status é destinado para os casos em que o cliente parece ter cometido um erro. Exceto quando estiver respondendo a uma solicitação HEAD, o servidor deve incluir uma entidade que contém uma explicação sobre a situação de erro, e se é uma condição temporária ou permanente. Esses códigos de status são aplicáveis a qualquer método de solicitação. Os agentes do usuário devem exibir qualquer entidade incluída para o usuário. Estes são tipicamente os códigos de erro mais comuns encontrados durante online.

3.4.1 400 Requisição inválida

O pedido não pode ser entregue devido à sintaxe incorreta.

3.4.2 401 Não autorizado

Semelhante ao 403 Proibido, mais especificamente para o uso quando a autenticação é possível, mas não conseguiu ou ainda não foram fornecidos. A resposta deve incluir um cabeçalho do campo www-authenticate contendo um desafio aplicável ao recurso solicitado. Veja Basic autenticação de acesso e autenticação Digest acesso.

3.4.3 402 Pagamento necessário

Reservado para uso futuro. A intenção original era que esse código pudesse ser usado como parte de alguma forma de dinheiro digital ou de micro pagamento regime, mas isso não aconteceu, e esse código não é usado normalmente.

3.4.4 403 Proibido

O pedido é reconhecido pelo servidor mas este recusa-se a executá-lo. Ao contrário resposta “401 Não Autorizado”, autenticação não fará diferença e o pedido não deve ser requisitado novamente.

3.4.5 404 Não encontrado

O recurso requisitado não foi encontrado, mas pode ser disponibilizado novamente no futuro. As solicitações subsequentes pelo cliente são permitidas.

3.4.6 405 Método não permitido

Foi feita uma solicitação de um recurso usando um método de pedido não é compatível com esse recurso, por exemplo, usando GET em um formulário, que exige que os dados a serem apresentados via POST, PUT ou usar em um recurso somente de leitura.

3.4.7 406 Não Aceitável

O recurso solicitado é apenas capaz de gerar conteúdo não aceitáveis de acordo com os cabeçalhos Accept enviados na solicitação.

3.4.8 407 Autenticação de proxy necessária

3.4.9 408 Tempo de requisição esgotou (Timeout)

O servidor sofreu timeout ao aguardar a solicitação. De acordo com as especificações HTTP W3: “O cliente não apresentou um pedido dentro do tempo que o servidor estava preparado para esperar. O cliente PODE repetir o pedido sem modificações a qualquer momento mais tarde.”

3.4.10 409 Conflito

Indica que a solicitação não pôde ser processada por causa do conflito no pedido, como um conflito de edição.

3.4.11 410 Gone

Indica que o recurso solicitado não está mais disponível e não estará disponível novamente. Isto deve ser usado quando um recurso foi intencionalmente removido e os recursos devem ser removidos. Ao receber um código de estado 410, o cliente não deverá solicitar o recurso novamente no futuro. Clientes como motores de busca devem remover o recurso de seus índices. A maioria dos casos de uso não necessitam de clientes e motores de busca para purgar o recurso, e um "404 Not Found" pode ser utilizado.

3.4.12 411 comprimento necessário

O pedido não especifica o comprimento do seu conteúdo, o que é exigido pelo recurso solicitado.

3.4.13 412 Pré-condição falhou

O servidor não cumpre uma das condições que o solicitante coloca na solicitação.

3.4.14 413 Entidade de solicitação muito grande

A solicitação é maior do que o servidor está disposto ou capaz de processar.

3.4.15 414 Pedido-URI Too Long

O URI fornecido foi muito longo para ser processado pelo servidor.

3.4.16 415 Tipo de mídia não suportado

A entidade tem um pedido tipo de mídia que o servidor ou o recurso não tem suporte. Por exemplo, o cliente carrega uma imagem como image / svg + xml, mas o servidor requer que imagens usar um formato diferente.

3.4.17 416 Solicitada de Faixa Não Satisfatória

O cliente solicitou uma parte do arquivo, mas o servidor não pode fornecer essa parte. Por exemplo, se o cliente pediu uma parte do arquivo que está para além do final do arquivo.

3.4.18 417 Falha na expectativa

O servidor não pode cumprir as exigências do campo de cabeçalho Espere-pedido.

3.4.19 418 Eu sou um bule de chá

Este código foi definido em 1998 como uma das tradicionais brincadeiras de 1º de abril da IETF, na [RFC 2324](#), Hyper Text Cafeteira Control Protocol, e não é esperado para ser implementado por servidores HTTP reais.

3.4.20 422 Entidade improcessável (WebDAV) ([RFC 4918](#))

O pedido foi bem formado, mas era incapaz de ser seguido devido a erros de semântica.

3.4.21 423 Fechado (WebDAV) ([RFC 4918](#))

O recurso que está sendo acessado está bloqueado.

3.4.22 424 Falha de Dependência (WebDAV) ([RFC 4918](#))

A solicitação falhou devido à falha de uma solicitação anterior (por exemplo, um PROPPATCH).

3.4.23 425 coleção não ordenada ([RFC 3648](#))

Definido em projectos de “WebDAV Avançada Coleções Protocolo”, mas não está presente no “Web Distributed Authoring and Versioning (WebDAV) Ordenados Coleções protocolo”.

3.4.24 426 Upgrade Obrigatório ([RFC 2817](#))

O cliente deve mudar para um outro protocolo, como TLS/1.0 . Resposta n º 444 Um Nginx extensão do servidor HTTP. O servidor retorna nenhuma informação para o cliente e fecha a conexão (útil como um impedimento para malware). Com 449 Repetir Uma extensão de Microsoft. O pedido deve ser repetida após a realização da ação apropriada.

3.4.25 450 bloqueados pelo Controle de Pais do Windows

Uma extensão de Microsoft. Este erro é dado quando Parental Controls do Windows estão ativadas e está bloqueando o acesso a determinada página da web.

3.4.26 499 cliente fechou Pedido (utilizado em ERPs/VPSA)

Um Nginx extensão do servidor HTTP. Este código é introduzido para registrar o caso quando a conexão é fechada pelo cliente ao servidor HTTP é o processamento de seu pedido, fazendo com que servidor não consiga enviar o cabeçalho HTTP de volta.

3.5 5xx outros erros

3.5.1 500 Erro interno do servidor (Internal Server Error)

Indica um erro do servidor ao processar a solicitação. Na grande maioria dos casos está relacionada as permissões dos arquivos ou pastas do software ou script que o usuário tenta acessar e não foram configuradas no momento da programação/construção do site ou da aplicação. Para corrigir, verifique o diretório em que o arquivo ou recurso que houve falha de acesso está localizado, e este arquivo (bem como todos os outros), obedecem às regras seguintes:

Pastas — chmod 755 (não utilizar 777) Arquivos — chmod 644 (não utilizar o 777, só utilizar outro se for expressamente solicitado na instalação)

OBS.: algumas aplicações e ou sistemas requerem permissões diferenciadas, pelo qual é importante verificar com os criadores do scripts/sistema, qual seria a permissão correta a usar. O exemplo descreve como é realizado em sistemas operacionais Unix-like. Fazer analogia como é realizado em sistemas como Windows (Windows 7, 8, XP entre outros).

Este erro também pode ocorrer se o arquivo .htaccess do seu site estiver modificando os parâmetros ou tentando fazer o PHP utilizar comandos como: php_flag ou php_value. Remova qualquer entrada com esses comandos do arquivo .htaccess. Se for fazer modificações nos parâmetros do PHP, utilize o arquivo php.ini para fazer isso.

3.5.2 501 Não implementado (Not implemented)

O servidor ainda não suporta a funcionalidade ativada

3.5.3 502 Bad Gateway

Em regra, o erro quando há uma configuração imprecisa entre os computadores de back-end, possivelmente incluindo o servidor Web no site visitado. Antes de analisar este problema, é necessário limpar o cache do navegador, completamente.

Se estiver navegando na Web e observar este problema em todos os websites visitados, então 1) o seu provedor de serviço de Internet tem uma falha/sobrecarga em um equipamento principal ou 2) tem algo de errado com a sua conexão interna à Internet, por exemplo, o firewall não está funcionando corretamente. Se for o primeiro caso, somente o seu provedor pode ajudar. Se for o segundo, você precisa corrigir o que quer que esteja prevenindo que você acesse a Internet.

Se tiver este problema somente em alguns websites visitados, provavelmente existe um problema nos sites. Por exemplo, uma das peças dos equipamentos estão falhando ou estão sobrecarregadas. Entre em contato com os responsáveis destes sites.

3.5.4 503 Serviço indisponível (Service Unavailable)

O servidor está em manutenção ou não consegue dar conta dos processamentos de recursos devido à sobrecarga do sistema. Isto deve ser uma condição temporária.

3.5.5 504 Gateway Time-Out

É caracterizado por erros particulares do site em questão. Pode ser que o site esteja em manutenção ou não exista.

3.5.6 505 HTTP Version not supported

A maioria dos browsers assumem que os servidores de rede suportam versões 1.x do protocolo HTTP. Na prática, as versões muito antigas como a 0.9 são pouco utilizadas atualmente, não apenas porque eles fornecem pouca segurança e desempenho mais baixo do que as versões mais recentes do protocolo. Então, se acontecer esse erro no seu navegador de rede, a única opção é fazer o upgrade do software do servidor de rede. Se a versão da solicitação 1.x falhar, pode ser porque o servidor de rede está suportando versões incorretas do protocolo 1.x, em vez de não suportá-las.

3.6 6xx outros erros

Especificação de Interface

4.0 Inicialização

Ao iniciar, o NAVS efetua uma série de inicializações a saber:

- Cheque da licença contra o sistema **Licença On-Line - LOL**
 - Para tanto as variáveis respectivas devem estar corretamente preenchidas ou via a configuração do NAVS ou via o comando **SET TYPE=VAR**
- Caso haja um arquivo **NAVS.ZIP** presente este será descomprimido, seu conteúdo extraído e o mesmo será excluído do terminal.
- Por último, executa um arquivo com os primeiros comandos da aplicação. Há uma diferença aqui em relação ao seu antecessor uma vez que alguns terminais usam sistema operacional Linux o qual é **case sensitive**. Dessa forma, o NAVS passa a procurar, na ordem abaixo, os seguintes arquivos para executar:
 - init.xml -> INIT.XML -> INIT.xml -> init.XML
 - O primeiro arquivo, na ordem acima, que for localizado será carregado para execução e a busca se encerrará.

5.0 Configuração

Para entrar na configuração do NAVS acionar simultaneamente as teclas **VERDE** e **1** segurando ambas pressionadas por uns instantes.

Além da configuração feita pelos menus NAVS, é possível sobrepô-la através de um arquivo config.xml carregado no terminal. Analogamente ao arquivo init.xml, o NAVS procura pelo arquivo na ordem abaixo:

- config.xml -> CONFIG.XML -> CONFIG.xml -> config.XML

IMPORTANTE: Nesse arquivo só deve conter comandos <SET TYPE=VAR ...>

5.1 Configuração do tipo de conexão

Para configurar a parte de comunicação do terminal, entrar na opção **1 Conexão** e escolher uma das 4 opções possíveis:

- **GPRS (CHIP)**
- **WiFi**
- **Ethernet**
- **Offline**

5.2 Configuração de CHIPS

Ao se usar um CHIP para comunicação GPRS, as configurações de APN, usuário e senha devem ser feitas. De preferência, as operadoras devem ser consultadas para obter os valores para o chip usado.

De qualquer forma, aqui estão os valores usados para chips públicos das 4 operadoras principais do Brasil:

- **VIVO:**

Especificação de Interface

- APN: **ZAP.VIVO.COM.BR**
- USUÁRIO: **VIVO**
- PASSWORD: **VIVO**
- **CLARO:**
 - APN: **CLARO.COM.BR**
 - USUÁRIO: **CLARO**
 - PASSWORD: **CLARO**
- **TIM:**
 - APN: **TIMBRASIL.BR**
 - USUÁRIO: **TIM**
 - PASSWORD: **TIM**
- **OI:**
 - APN: **GPRS.OI.COM.BR**
 - USUÁRIO: **OI**
 - PASSWORD: **OI**
- **LYRA (de qualquer operadora):**
 - APN: **GPRSNAC.COM.BR**
 - USUÁRIO:
 - PASSWORD:

Caso o terminal POS possua dois *slots* de Chips habilitados, será possível configurar até 2 chips. Nesse caso, a opção **2 Rede** permitirá escolher qual chip será usado. Nesse mesmo menu, haverá a escolha de qual chip configurar, do **SIM 1** ou do **SIM2**. Caso o terminal possua apenas 1 *slot* habilitado, a opção **2 Rede** irá direto para a configuração do único *slot* possível (o **SIM 1**).

Para configurar, após definir o tipo de conexão como **GPRS**, entrar na opção **2 Rede** e definir a APN, usuário e senha do chip nas três opções mostradas no menu:

- **APN**
- **User**
- **Password**

Especificação de Interface

IMPORTANTE: Para se obter as letras, escolha a tecla que possui a letra desejada e a pressione. Em seguida, use a tecla à direita da tecla **0** ou a tecla **#** ou escrito **MENU** para escolher a letra. Caso se deseje uma letra minúscula, basta, após chegar na letra desejada, teclar a tecla à esquerda da tecla **0** ou a tecla com o desenho de uma impressora.

5.3 Configuração de WiFi

Uma vez escolhido WiFi, entrar na opção **2 Rede** e caso queira escolher a sua rede WiFi por uma lista, entre com a opção **4 Lista de APs**. Para paginar essa lista siga as dicas escritas na tela. Para escolher a rede desejada, tecle o número associada a ela.

Uma vez escolhida a rede, escolher a opção **2 KEY** para entrar com a senha da rede.

Ainda é possível determinar se é para o terminal receber automaticamente o IP do roteador da rede através do servidor interno *DHCP*. Para tal, entrar na opção **3 DHCP** e escolher a opção **1 Ativar/Desativar**. Caso a opção seja para usar IP fixo no terminal, escolha desativar indique os demais parâmetros para o terminal usar:

- IP
- MASK
- GATEWAY
- DNS 1
- DNS 2

Solicite essas informações para o pessoal que administra sua rede.

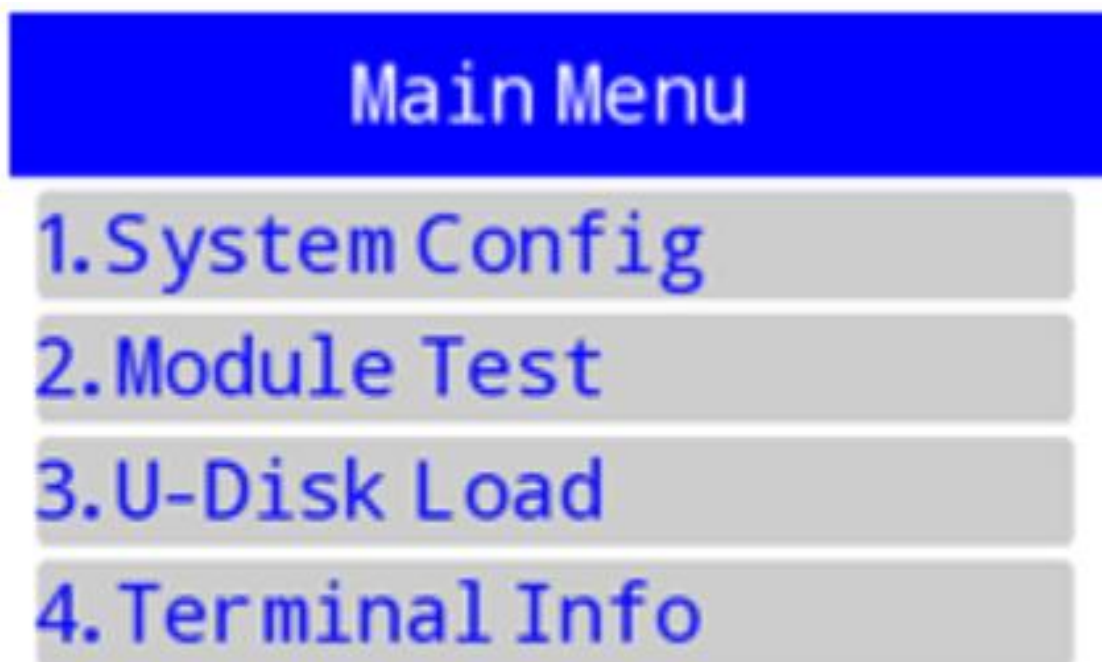
IMPORTANTE: para a lista de redes aparecer, o Wifi tem que estar ativo. Caso não aparece WiFi na linha de status no topo, sair da configuração com a opção 4 na tela vermelha e reentrar na configuração do NAVS.

6.0 Carga da aplicação no terminal e arquivos de atualização (extensão AUP)

O NAVS se apresenta para carga em terminais produtivos em duas versões dependendo da qual *SITEF* acessar:

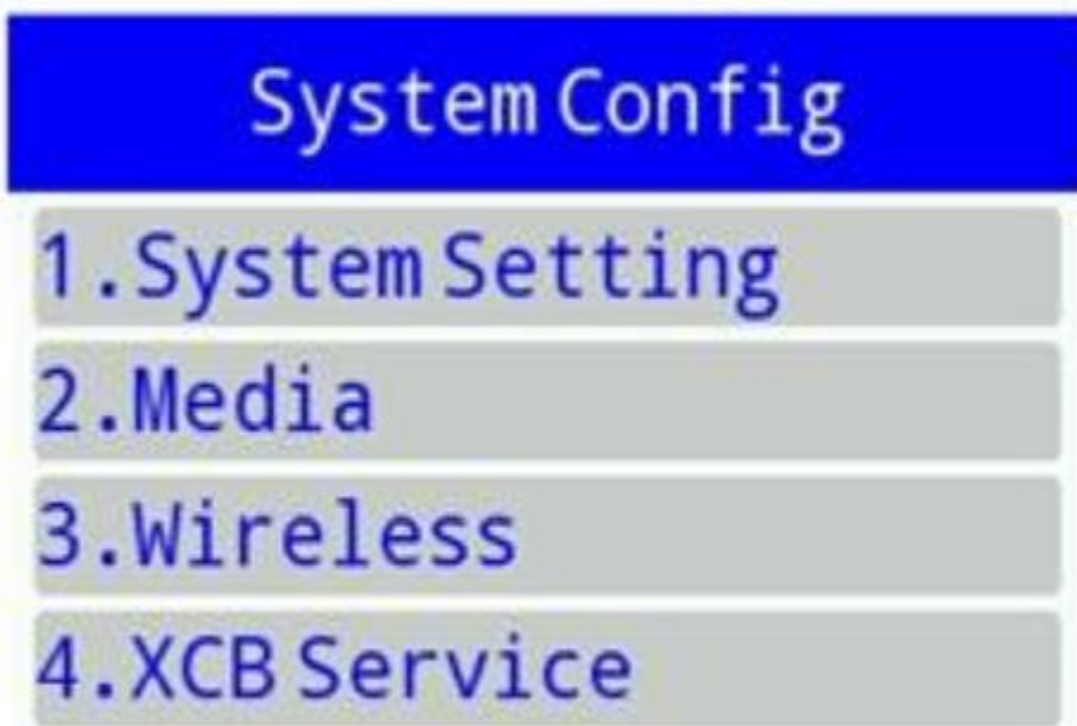
- Simulação: **NAVS2_SIM_2.x.y.z_sign.aip**
- Produção: **NAVS2_PRD_2.x.y.z_sign.aip**

Primeiramente certificar-se de que o terminal está preparado para a carga. Para tal, inicie o mesmo e aperte várias vezes a tecla **2** até a seguinte tela aparecer:



Entrar na opção **1. System Config**. A *password* inicial é: **123456**.

A tela **System Config** será mostrada. Pode ser que seja um pouco diferente da imagem abaixo:



Entre na opção **XCB Service** (atualmente é opção **5**).

XCB Service

1. Close Service

2. COM

3. USB

4. Network

Certifique-se de que a opção **USB** esteja marcada com asterisco. Caso não esteja, tecla sua opção.

Tecla a tecla vermelha até a mensagem **Quit TM?** aparecer. Confirme com a tecla verde. O Terminal irá reiniciar.

Para proceder com a carga, é necessário ter instalado no computador o aplicativo da PAX **TermAssist**.

Uma vez iniciado, a tela abaixo se apresentará:

Especificação de Interface



Conecte o terminal ao computador via um cabo USB-MICRO USB.

Clicar no ícone **Setting**:



A tela de configuração aparecerá:

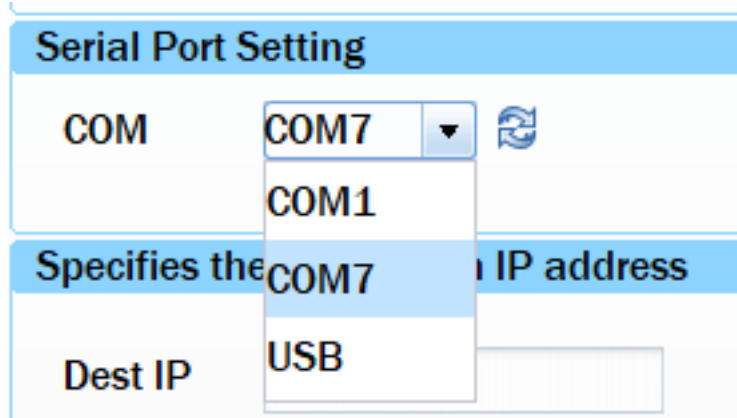
Especificação de Interface



Refresque a lista de portas seriais clicando no ícone abaixo:



No menu de portas seriais, selecione a porta do terminal (normalmente a diferente de COM1):



Clique no botão de conectar:

Especificação de Interface



A linha de status deve mudar para a verde (*Connected*):

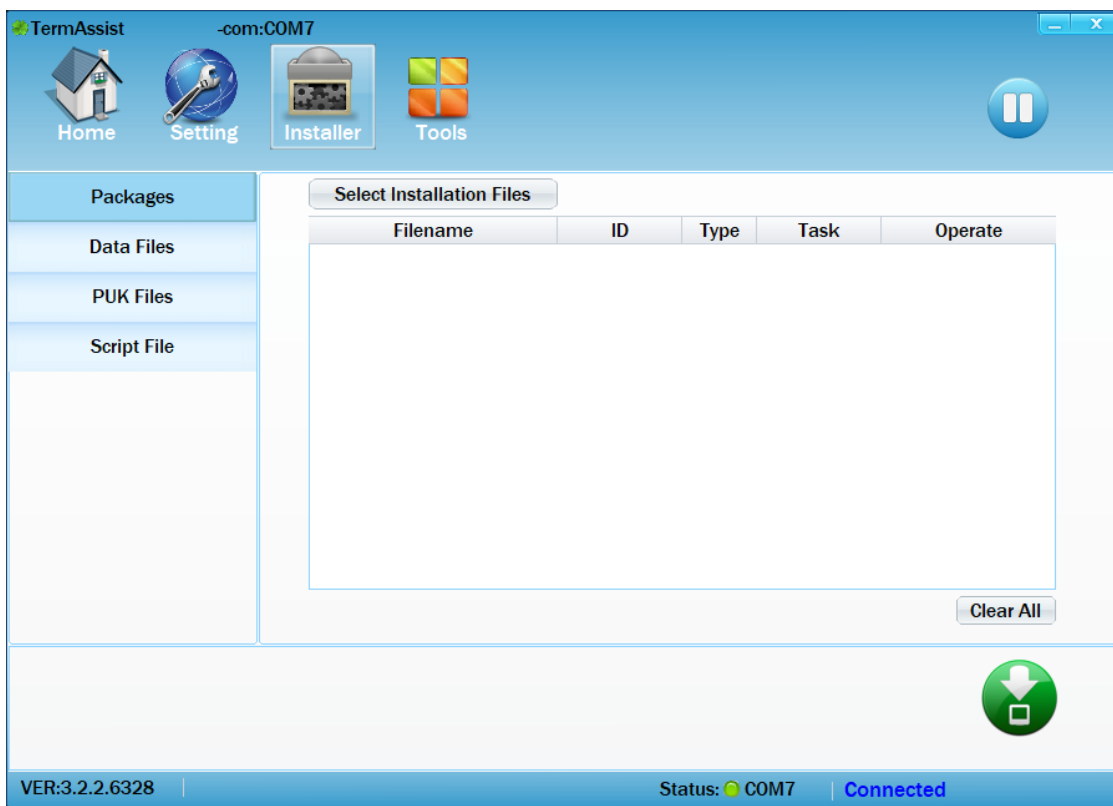


Caso o terminal não conectar, verifique se o mesmo está ligado.

Uma vez conectado, entre na opção *Installer*.



A tela de instalação aparecerá:

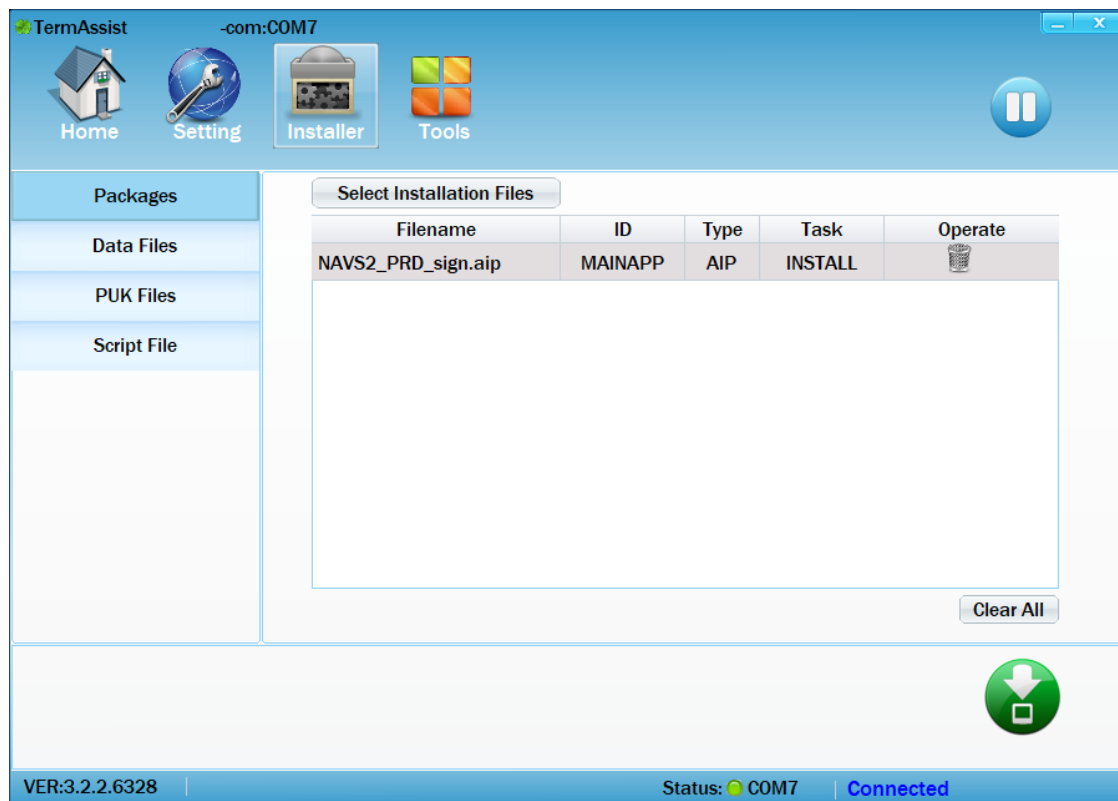


Para carregar o NAVS, a opção *Packages* deve estar seleccionada como na imagem acima.

Especificação de Interface

Clique no arquivo de carga do NAVS escolhido (SIMULADO ou PRODUÇÃO) e arraste para essa tela:

 NAVS2_PRD_sign.aip	10/11/2021 09:21	Arquivo AIP	6.838 KB
 NAVS2_SIM_sign.aip	10/11/2021 09:22	Arquivo AIP	6.828 KB



Caso tenha colocado o arquivo errado o queira trocar, clique na lixeira para excluí-lo.

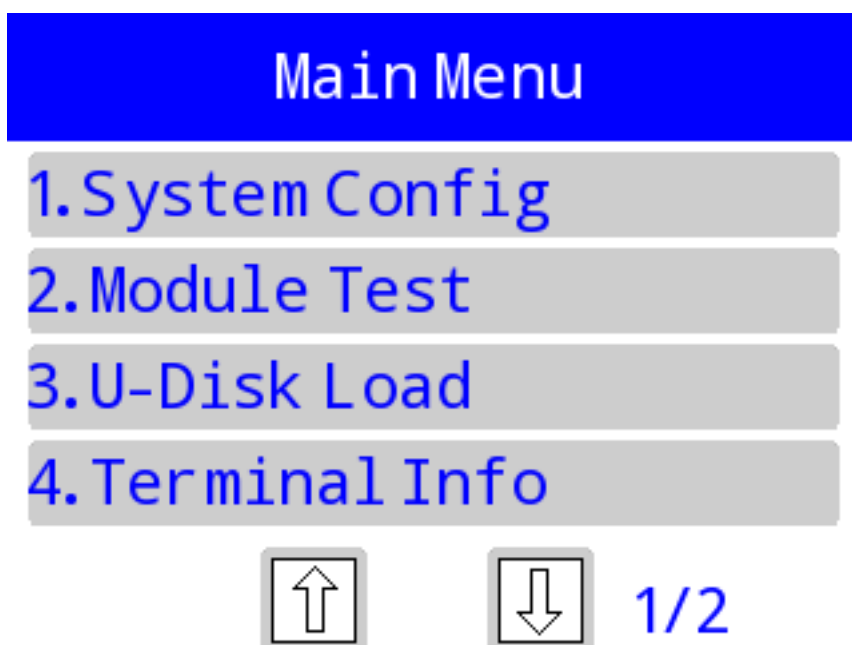
Finalmente, clique no ícone de *download* e espere a carga terminar:



Uma vez terminada a carga, reinicie o NAVS.

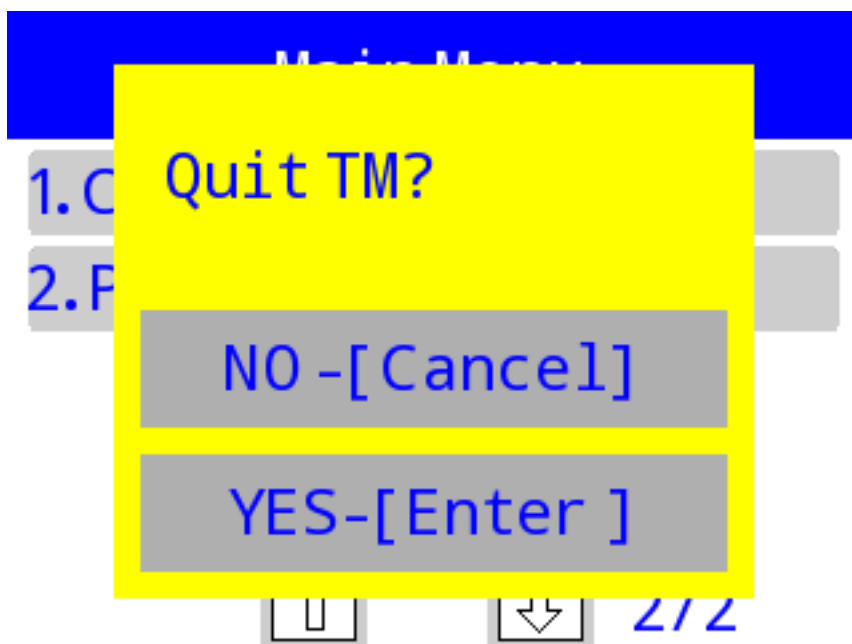
Caso esteja com o NAVS executando, tecle no terminal ENTER+7 e o mesmo voltará para a tela do sistema operacional semelhante à tela abaixo:

Especificação de Interface



Para voltar ao NAVS, tecle X (Vermelha):

Na tela abaixo tecle ENTER:

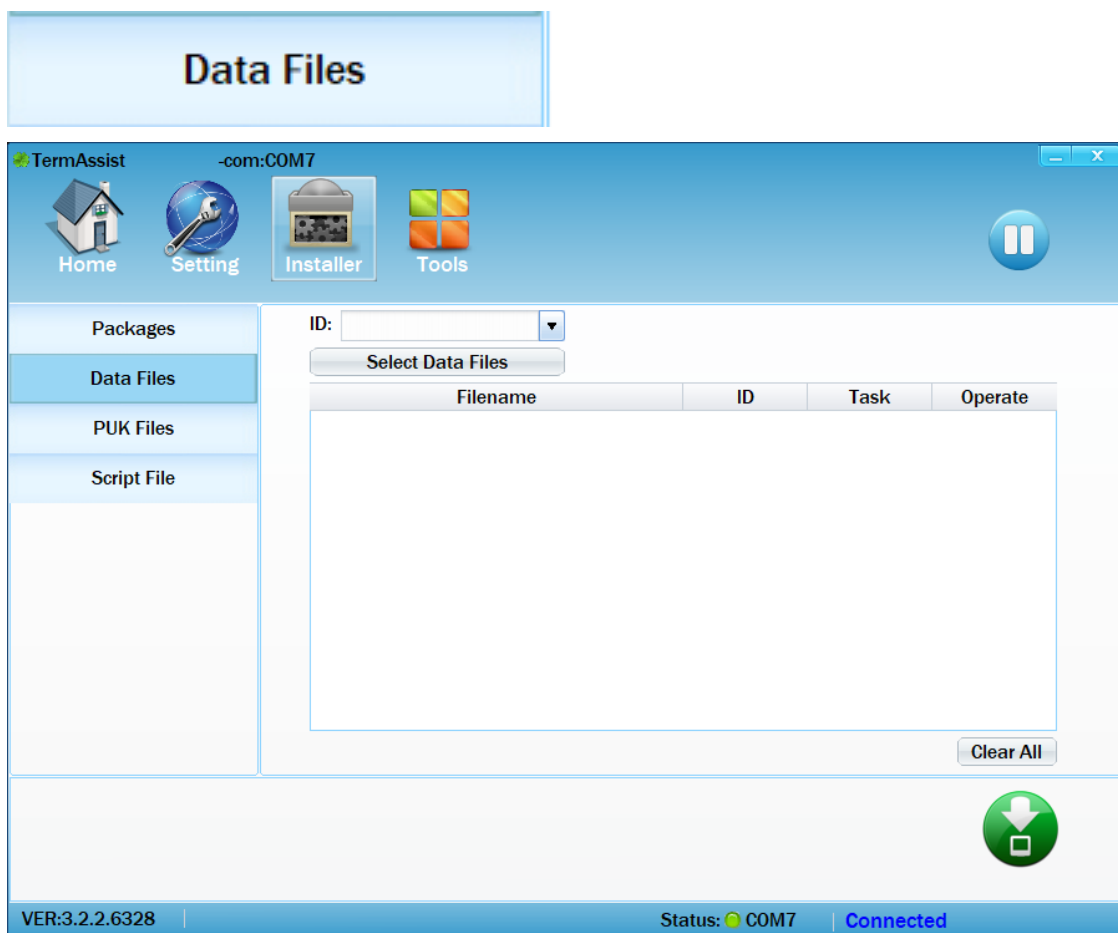


O NAVS entrará.

7.0 Carga de imagens e outros arquivos para uso do NAVS

Caso seja necessário a carga de imagens, dados etc, na tela do *Installer* selecionar a opção *Data files* clicando no ícone abaixo:

Especificação de Interface



O processo é semelhante arrastando os arquivos para essa tela e clicando no ícone download. Antes, porém, a aplicação tem que ser selecionada no menu *ID*:

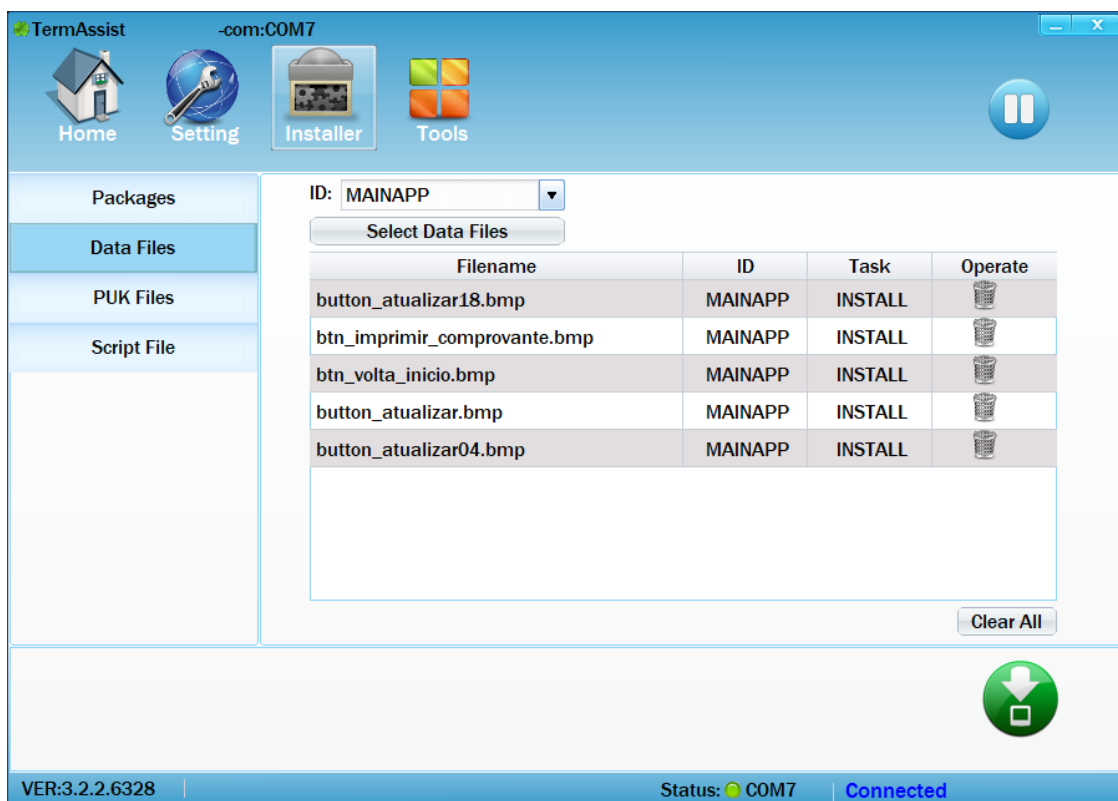
ID: ▼

clicando na seta para baixo:

ID: ▼

selecione MAINAPP

Especificação de Interface

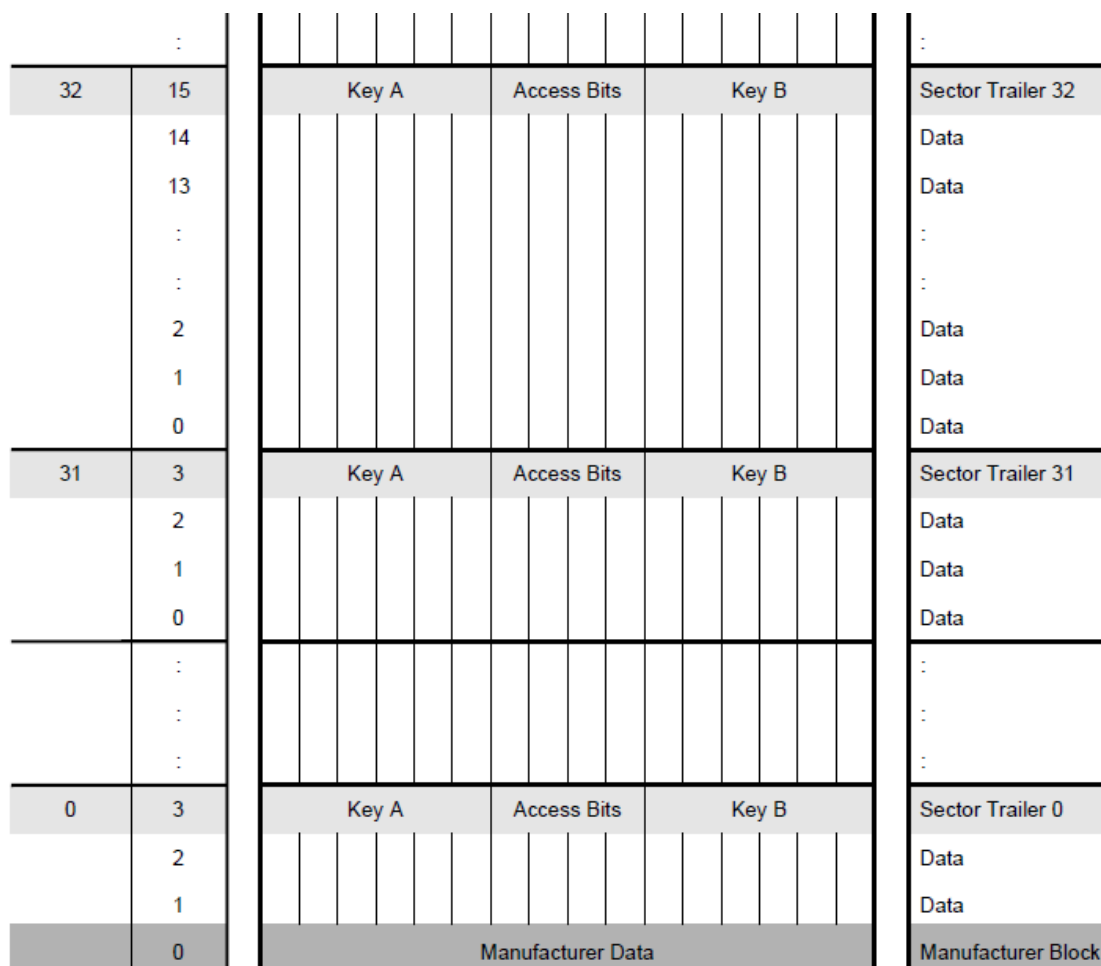


8.0 Mifare: gravação e leitura de dados

Os comandos **CTLSWRITE** e **GET TYPE=MIFARE_READ** são usados para, respectivamente, gravar e ler informações nos blocos de dados do cartão Mifare.

A imagem abaixo detalha a arquitetura interna dos cartões Mifare:

Especificação de Interface



Os blocos possuem o tamanho de 16 bytes e são agrupados de 4 em 4 em setores do setor 0 até o setor 31. A partir do setor 32, são agrupados de 16 em 16 blocos. A numeração dos blocos começa em 0 em cada setor.

O primeiro bloco (setor 0 - bloco 0) possui dados específicos do fornecedor e não é acessível pelos comandos NAVS.

Do mesmo modo, os últimos blocos do setor (bloco 3 dos setores 0 a 31 e bloco 15 dos setores 32 em diante) , possuem as chaves de acesso de cada bloco e também são inacessíveis pelos comandos.

Caso o comando informe um bloco inacessível os comandos retornará erro de bloco inválido.

Foi criado um parâmetro **DATABLOCK** nos comandos que, iniciando com o valor 0, acessa apenas os blocos disponíveis.

Como limitação, os blocos de dados acessíveis pelo parâmetro **DATABLOCK** vão de 0 a 40.

9.0 Detalhamento dos Comandos

Simbologia usada:

Especificação de Interface

Símbolo	Significado	Exemplo	Descrição do exemplo
	Escolha entre possíveis valores	ENABLE DISABLE	possibilidade de usar ENABLE ou DISABLE
[val]	Valor opcional	[NOENTER]	O argumento NOENTER pode ou não ser usado
[val1 val2]	Além de opcional, há mais de uma possibilidade de informar o valor	[NOENTER NOENTER=val]	possibilidade de usar NOENTER ou NOENTER=val ou não usá-los

9.1 LOCAL

#1

Executa os comandos NAVS a partir de um arquivo.

<LOCAL FILE=va1 [CONTINUE]>

- **FILE:** Nome do arquivo presente no terminal com os comandos NAVS a serem executados
- **CONTINUE:** usado para que o buffer de comandos não seja imediatamente enviado ao NAVS. Caso **CONTINUE** não seja usado o comando **LOCAL** será o último dessa interação sendo os demais comandos desprezados. O conteúdo do arquivo deve ter em algum ponto um comando **LOCAL, POST, DATA, LUA** para retornar o comando ao NAVS, caso contrário, o comando retorna ao arquivo **INIT.XML**.

9.2 DATA

#2

Executa *script* Lua.

<DATA [CONTINUE]>script</DATA>

O argumento **CONTINUA** tem uso semelhante ao do comando **LOCAL**.

9.3 LUA

#3

Executa *script* Lua mantendo o ambiente Lua intacto entre chamadas.

<LUA [CONTINUE]>script</LUA>

O argumento **CONTINUA** tem uso semelhante ao do comando **LOCAL**.

Especificação de Interface

9.4 CANCEL_KEY

#4

DEPRECADO: Usar **GET TYPE=CANCEL_KEY** e **SET TYPE=CANCEL_KEY**

Habilita ou desabilita a tecla VERMELHA como cancelamento da função NAVS executando.

<CANCEL_KEY TYPE=ENABLE|DISABLE>

ou:

<CANCEL_KEY ENABLE|DISABLE>

- **ENABLE:** *habilita a tecla Cancel (tecla VERMELHA).*
- **DISABLE:** *desabilita a tecla Cancel (tecla VERMELHA)*

9.5 POST

#5

Envia informações ao Servidor.

<POST [IP=val] [PORT=val] [RESOURCE=val] [HOST=val] [SSL=val] [SSL_METHOD=val] [SSL_SNI=val] [TIMEOUT=val] [RC_NAME=var] [ERR_FUNC=var] [ERR_FRET=var] [ERR_SSLRET=var] [ERR_ERROR=var]>

Parâmetro	Significado	Valores possíveis
IP	IP do servidor	
PORT	Porta da aplicação no servidor	
RESOURCE	Recurso no servidor entrada da aplicação	
HOST	Atributo HTTP HOST	
SSL	Indica protocolo HTTPS	'1' ou 'S'
SSL_METHOD	Indica o nível de SSL	1 = SSL V3 2 = TLS V1 3 = TLS V1.1 4 = TLS V1.2
SSL_SNI	Indica se o certificado é pra ser validado contra o servidor - SERVER NAME INDICATION	'1' ou 'S'
TIMEOUT	Tempo em segundos para a execução do comando	
RC_NAME	Variável que possui o retorno do comando POST	
ERR_FUNC	Em caso de erro, variável que possui a função TCPIP que deu erro	
ERR_FRET	Em caso de erro, variável que possui o retorno da função TCPIP que deu erro	

Especificação de Interface

Parâmetro	Significado	Valores possíveis
ERR_SSLRET	Em caso de erro de SSL, variável que possui o erro SSL	
ERR_ERROR	Em caso de erro, variável que possui o erro	

9.6 *RESET ou REBOOT*

#6

Reinicia o POS.

<RESET [COMM] [CONSOLE]>

ou

<REBOOT [COMM] [CONSOLE]>

- **COMM:** Parâmetro que reinicia a comunicação, sem reiniciar o POS. Usado para chaveamento de comunicação, WiFi, Chip, por exemplo.
- **CONSOLE:** Parâmetro que reinicia a console, sem reiniciar o POS. Para que as informações de console da comunicação sejam atualizadas, reinicie também a comunicação.

9.7 *PRINTER*

#7

Imprime conteúdo na impressora do terminal.

<PRINTER [DEFAULT] [DOUBLE_WIDTH] [DOUBLE_HEIGHT] [INVERSE] [WIDTH=w] [HEIGHT=h] [FONT=f]>

Conteúdo

</PRINTER>

- **DEFAULT:** Volta a configuração comando PRINTER ao definido na configuração do NAVS
- **DOUBLE_WIDTH:** Dobra a largura do fonte
- **DOUBLE_HEIGHT:** Dobra a altura do fonte
- **INVERSE:** Imprime branco em fundo preto
- **WIDTH:** define a largura do fonte em pixels. O modelo v240m não suporta essa opção, apenas o height.
- **HEIGHT:** define a altura do fonte em pixels
- **FONT:** define o nome do fonte a ser usado entre os carregados previamente no terminal

O conteúdo pode ter comandos XML que modificam o resultado do que é impresso:

Especificação de Interface

Causa um avanço de linha na impressora voltando a impressão para a primeira coluna

Conteúdo

Altera momentaneamente os parâmetros de impressão para o conteúdo selecionado entre as *tags*.

9.7.1 Configuração hierárquica do comando PRINTER

O comando PRINTER respeita as configurações de forma hierárquica onde o mais específico sobrepõe o mais genérico durante a duração do seu efeito. A ordem de efeito do mais genérico para o mais específico é a seguinte:

- Configuração do NAVS
 - nunca expira a validade e é a instância mais genérica
- Comando SET TYPE=PRINTER
 - expira no próximo comando SET TYPE PRINTER tendo duas possibilidades:
 - volta a valer o definido na configuração caso o comando use o argumento DEFAULT: <SET TYPE=PRINTER DEFAULT>
 - passa a valer o indicado no novo comando SET TYPE=PRINTER
- Comando PRINTER
 - expira no fim do comando ao encontrar o </PRINTER>
- Comando SPAN
 - expira no fim do comando ao encontrar o

9.8 TONE

#8

Emite um **beep** sonoro.

Exemplo de uso:

<TONE TYPE=NORMAL | ERROR>

- **NORMAL:** *emite um beep indicando normalidade.*

Especificação de Interface

- **ERROR:** *emite um beep indicando anormalidade.*
-

9.9 DELAY

#9

Aguarda um tempo antes de continuar o próximo comando.

<DELAY TIME=**t**>

- **TIME:** tempo de espera em segundos

Ideal para mensagens para a Console temporizadas.

Exemplo para uso com mensagem para console temporizada em 3 segundos:

<CONSOLE> Texto para display </CONSOLE>

<DELAY TIME=**3**>

9.10 PRNLOGO

#10

Imprimir imagens a partir de um arquivo na impressora do terminal.

<PRNLOGO NAME=**n** [SPACES=**s**] [RC_NAME=**r**]>

- **NAME:** nome do arquivo da imagem a ser impressa já presente no terminal
- **SPACES:** deslocamento a direita da imagem (em pixels)
- **RC_NAME:** nome da variável que receberá o código de retorno do comando. Os valores que podem ser retornados são:
 - 0 Sucesso
 - -1 Arquivo não encontrado
 - -2 Parâmetro NAME não especificado
 - -3 Erro ao abrir o dispositivo de impressão

**** Nota**** : O arquivo de imagem deve ser do tipo .BMP monocromático (intensidade de 1 bit por pixel). Além disso, a largura da imagem deve ser até 384 pixels.

Para os modelos Prolin - PAX, é necessário converter o arquivo de imagem .BMP para um formato específico antes de carregá-lo no POS. Segue as instruções de como gerar esse arquivo de imagem no formato correto:

- O arquivo .BMP deve ser monocromático de 1 bit de intensidade.
- Abra a ferramenta Bitmap Converter (fornecido pela PAX) para converter o arquivo em um header .H.

Especificação de Interface

OBS: O tamanho da imagem deve ter até 384 pixels por largura. A altura é ilimitada.

Exemplo de uso:

```
<PRNLOGO NAME=filename.bmp SPACES=10 RC_NAME=ret>
```

9.11 CONLOGO

#11

Exibir imagens a partir de um arquivo na tela do terminal.

```
<CONLOGO NAME=n X=n Y=m [NOCLS|NOCLS=x]>
```

- **NAME** Especifica o nome do arquivo de imagem.
- **X** Especifica a posição em pixels no eixo X (horizontal). A origem (0, 0) fica no canto superior esquerdo.
- **Y** Especifica a posição em pixels no eixo Y (vertical). A origem (0, 0) fica no canto superior esquerdo.
- **NOCLS** Não limpa a tela antes de exibir a imagem. Se for informado um valor, este deve ser 1.

Nota: Para os modelos da PAX (Prolin), o arquivo de imagem pode ser do formato BMP ou PNG. Como os arquivos .PNG pode ter 32 bits por pixel de intensidade, é possível exibir imagens com transparência.

Exemplo de uso:

```
<CONLOGO NAME=filename.png X=1 Y=1 NOCLS>
```

9.12 DEVICE

#12

Ler e gravar em um dispositivo externo ao terminal

```
<DEVICE TYPE=t DEV=d FILE=f NAME=n ERR=e RC_NAME=r>
```

- **TYPE:** tipo de ação, sendo:
 - **COPYTO:** para cópia de arquivo do terminal para o dispositivo externo
 - **COPYFROM:** para cópia de arquivo do dispositivo externo para o terminal
 - **DELETE:** para apagar o arquivo do dispositivo externo
- **DEV:** indica o tipo de dispositivo externo, podendo ser:
 - **USB**
 - **FLASHCARD**
 - **SD**

Especificação de Interface

- **SDCARD**
 - **FILE:** o nome do arquivo no terminal a ser gravado no dispositivo
 - **NAME:** parâmetro antigo - **DEPRECADO** - sendo no formato:
 - **D:NNNN** onde:
 - **D** pode ser:
 - **M** para pendrive (USB)
 - **L** para cartão SD
 - **NNNN** o nome do arquivo
 - **ERR:** nome da variável receberá o retorno do comando - **DEPRECADO** - sendo:
 - 0 para sucesso
 - 1 para falha
 - **RC_NAME:** nome da variável que receberá o resultado do comando com mais detalhes sendo 0 para sucesso.
-

9.13 *FTP*

#13

Executa transferência de dados via protocolo FTP.

```
<FTP TYPE=GET|PUT HOST=h [PORT=p] [USERID=u] [PASSW=pw] LOCALFILE=lf REMOTEFILE=rf [TIMEOUT=to]
[RETURN=v] RCNAME=v>
```

- **TYPE:** tipo de chamada:
 - **GET:** para receber no terminal um arquivo do servidor
 - **PUT:** para enviar do terminal um arquivo para o servidor
 - **HOST:** endereço IP do servidor FTP
 - **PORT:** número da porta de comandos (default=21)
 - **USERID:** usuário autorizado
 - **PASSW:** senha do usuário
 - **LOCALFILE:** nome do arquivo no terminal a ser recebido (no caso de GET) ou enviado (no caso de PUT)
 - **REMOFILE:** nome do arquivo no servidor
 - **RETURN:** nome da variável que receberá o resultado do comando: 0 OK, 1 ERRO
 - **RCNAME:** nome da variável que receberá o código de retorno mais específico de erro
-

Especificação de Interface

9.14 UNZIP

#14

Descomprime um arquivo no formato **zip** presente no terminal.

<UNZIP NAME=**n** KEEP_FILE VERBOSE ERR=**v**>

- **NAME**: nome do arquivo a ser descomprimido presente no terminal
 - **KEEP_FILE**: indica que o arquivo não será excluído após o processo (default é ser excluído)
 - **VERBOSE**: mostra na tela do terminal o andamento da execução do comando
 - **ERR**: nome da variável que receberá o resultado do comando
 - 0 para sucesso
 - 1 para falha
-

9.15 CONSOLE

#15

Exibe texto na tela do terminal

<CONSOLE [DEFAULT] [CONTINUE] [NOCLS|NOCLS=**x**] [COLOR=**c**] [BGCOLOR=**b**] [FONT=**f**] [JUSTIFY=**j**]>

texto a exibir

</CONSOLE>

- **DEFAULT**: Volta a configuração comando CONSOLE ao definido na configuração do NAVS
- **CONTINUE**: Se informado, mostra o texto a partir da última linha da tela, caso contrário, da primeira
- **NOCLS**: Não limpa a tela antes de exibir o texto. Se for informado um valor, este deve ser 1.
- **COLOR**: indica a cor do fonte a ser usada. Pode ser do seguinte formato:
 - BRANCO
 - WHITE
 - PRETO
 - BLACK
 - CINZA CLARO
 - LIGHTGRAY
 - CINZA
 - GRAY
 - DARKGRAY

Especificação de Interface

- CINZAESCURO
- RED
- GREEN
- BLUE
- VERMELHO
- VERDE
- AZUL
- AMARELO
- YELLOW
- CIANO
- CYAN
- PURPURA
- MAGENTA
- RGB(r,g,b) onde:
 - r,g e b são valores de 0 a 100 indicando a percentagem dos componentes azul, verde e vermelho respectivamente
- **BGCOLOR**: indica a cor do fundo de fonte a ser usada. Os valores possíveis são os mesmos do parâmetro **COLOR**.
- **FONT**: define o nome do fonte a ser usado entre os carregados previamente no terminal*
- **JUSTIFY**: Indica o alinhamento do texto, sendo possível:
 - **LEFT**: Alinhado à esquerda
 - **RIGHT**: Alinhado à direita
 - **CENTER**: Centralizado
 - **CLOSE**: Logo após o último caractere do último comando **CONSOLE** executado.

O conteúdo pode ter comandos XML que modificam o resultado do que é mostrado na tela:

Causa um avanço de linha na console movendo o próximo conteúdo para a primeira coluna

Conteúdo

Altera momentaneamente os parâmetros de exibição na console para o conteúdo selecionado entre as *tags*.

9.15.1 Configuração hierárquica do comando **CONSOLE**

O comando **CONSOLE** respeita as configurações de forma hierárquica onde o mais específico sobrepõe o mais genérico durante a duração do seu efeito. A ordem de efeito do mais genérico para o mais específico é a seguinte:

- Configuração do **NAVS**
 - nunca expira a validade e é a instância mais genérica
- Comando **SET TYPE=CONSOLE**
 - expira no próximo comando **SET TYPE CONSOLE** tendo duas possibilidades:
 - volta a valer o definido na configuração caso o comando use o argumento **DEFAULT**: `<SET TYPE=CONSOLE DEFAULT>`
 - passa a valer o indicado no novo comando **SET TYPE=CONSOLE**
- Comando **CONSOLE**
 - expira no fim do comando ao encontrar o `</CONSOLE>`
- Comando **SPAN**
 - expira no fim do comando ao encontrar o ``

9.16 **INIT**

#16

Marca um ponto de retomada para quando pressionada a tecla **CANCEL**.

`<INIT>`

IMPORTANTE - Caso o comando **SET TYPE=CANCEL_KEY** tenha sido executado com a opção de file, o comando **INIT** é desprezado.

9.17 **RECTANGLE**

#17

Desenha um retângulo na tela e prepara o funcionamento do comando **GET TYPE=TOUCH**.

`<RECTANGLE NAME=n FILE=f X=x Y=y HEIGHT=h WIDTH=w COLOR=c BGCOLOR=b FILLED|FILLED=f VISIBLE|VISIBLE=v BLE=v>`

- **NAME**: conteúdo que será passado para o resultado do comando **GET TYPE=TOUCH** na variável indicada por **NAME**.

Especificação de Interface

- **FILE**: indica que o arquivo que será executado por um comando **LOCAL** caso o retângulo seja tocado
- **X** Especifica a posição em pixels no eixo X (horizontal). A origem (0, 0) fica no canto superior esquerdo.
- **Y** Especifica a posição em pixels no eixo Y (vertical). A origem (0, 0) fica no canto superior esquerdo.
- **HEIGHT**: Indica a altura do retângulo em pixels.
- **WIDTH**: Indica a largura do retângulo em pixels.
- **COLOR**: indica a cor da borda do retângulo conforme o padrão do comando **CONSOLE**
- **BGCOLOR**: indica a cor do interior do retângulo conforme o padrão do comando **CONSOLE** caso o parâmetro **FILLED** esteja presente
- **FILLED**: indica se o retângulo será preenchido pela cor **BGCOLOR**
- **VISIBLE**: indica se o retângulo será mostrado na tela

IMPORTANTE - O parâmetro **FILE** tem precedência sobre o parâmetro **NAME**.

9.18 LAUNCH_ITEM

#18

Define um item de execução para ser usado com o comando **GET TYPE=LAUNCH**

<LAUNCH_ITEM KEY=k FILE=f>

- **KEY**: indica a tecla que, ao ser pressionada, causara a execução do arquivo definido pelo parâmetro FILE
 - **FILE**: indica que o arquivo que será executado por um comando **LOCAL** a tecla correspondente seja pressionada
-

9.19 CAPTURE

#19

Permite a entrada de dados por múltiplos meios

<CAPTURE NAME=var RETURN=r><get type=...><get type=...>/CAPTURE>

- **NAME**: indica o nome da variável que receberá a resposta
- **RETURN**: indica o nome da variável que receberá o nome do comando GET que foi selecionado

Os seguintes comandos **GET** podem ser usados

- CARD

Especificação de Interface

- FIELD
- VALUE
- PASS
- KEY
- ANYKEY (usar KEY)
- SPECIALKEY (usar KEY)
- TOUCH
- LAUNCH
- IDCTLS
- MIFARE
- INSERTCARD

IMPORTANTE: Ao usar o comando CAPTURE a variável que recebe o retorno é a **NAME** do comando **CAPTURE**. Caso informado NAME nos comandos GET, ela será desprezada.

9.20 *WRITE_AT (deprecado)*

#20

Exibe texto na tela do terminal a partir de uma linha e coluna especificado pelo comando.

`<WRITE_AT [LINE=1] [COLUMN=c] [X=x] [Y=y]>texto a exibir</WRITE_AT>`

- **LINE:** Especifica a linha onde o texto será exibido
- **COLUMN:** Especifica a coluna onde o texto será exibido
- **X:** Especifica o pixel horizontal onde o texto será exibido
- **Y:** Especifica o pixel vertical onde o texto será exibido

Este comando utiliza configurações definidas pelo comando CONSOLE.

9.21 *RELICENSE*

#21

Remove a licença do NAVS, numa próxima reinicialização uma nova licença será requisitada.

Especificação de Interface

<RELICENSE>

9.22 Web Services - Comandos WS_GET, WS_POST e WS

#22

Executa uma chamada de *web service*

Os comandos WS_GET e WS_POST existem mas foram **DEPRECADOS** . Usar o comando **WS**

```
<WS METHOD={GET|POST|PUT|DELETE} IP=val PORT=val PATH=val [HOST=val] [SSL=val] [TIMEOUT=val] [RC_NAME=var] [ERR_DESCR=var] [HTTP_RESP=var] [ADD=hd1,hd2,hd3...] [hd1=val] [hd2=val] [hd3=val] [NAME=val] [BODY_IN=val] [BODY_OUT=val]>
conteúdo da requisição
</WS>
```

Parâmetro	Significado	Valores possíveis
IP	IP do servidor	
PORT	Porta da aplicação no servidor	
PATH	Recurso do webservices - deve incluir, se for o caso, os parâmetros a serem passados	
HOST	Atributo HTTP HOST	
SSL	Indica protocolo HTTPS	'1' ou 'S'
TIMEOUT	Tempo em segundos para a execução do comando	
RC_NAME	Variável que possui o retorno do comando WS	
ERR_DESCR	Em caso de erro, variável que possui a descrição do erro	
HTTP_RESP	Variável que possui o retorno HTTP do comando WS	
ADD	Indicar quais os headers opcionais a serem incluídos separados por vírgulas	
hd1, hd2, hd3 ...	Os headers opcionais a serem usados	
NAME	Nome da variável a receber o conteúdo da resposta	
BODY_IN	Nome do arquivo que contém o conteúdo da requisição. Caso este parâmetro não seja informado, o conteúdo é obtido do corpo do comando. IMPORTANTE: para conteúdos XML é obrigatório o uso de arquivo caso contrário, o NAVS apresentará resultado imprevisível	
BODY_OUT	Nome do arquivo que contém o conteúdo da resposta em substituição à variável NAME	

Exemplos

```
<ws method=get ip=www.receitaws.com.br porta=443 host=www.receitaws.com.br name=JSON path=/v1/cnpj/00000000000191 ssl=1 rc_name=RC err_descr=RESP timeout=60>
</ws>
```

Especificação de Interface

```
<ws method=post ip=postman-echo.com name=JSON path=/post add=Content-Type Content-Type=application/json ssl=1 rc_name=RC err_descr=RESP timeout=60>
{
  "oi":"mundo"
}
</ws>
```

```
<ws method=put ip=postman-echo.com name=JSON path=/put add=Content-Type Content-Type=application/json ssl=1 rc_name=RC err_descr=RESP timeout=60 body_in=requisicao.xml body_out=resposta.xml>
</ws>
```

9.23 *DELFILE*

#23

Remove arquivos do terminal.

```
<DELFILE NAME=n ERR=e>
```

- **NAME:** indica o nome do arquivo que será removido do POS
- **ERR:** indica o nome da variável que receberá o código de retorno:
 - 0 para sucesso
 - 1 para falha

Exemplo de uso:

```
<DELFILE NAME=delfileTestFile ERR=ERROVAR>
```

9.24 *CHGCONFNT*

#24

Comando que permite a mudança de fontes para o console.

```
<CHGCONFNT IDFONT=FONT.TTF SIZE=10 COLOR=RGB COLORBG=RGB INVERSE UNDERLINE KEEPBG >
```

Este comando está depreciado e recomendamos a utilização do comando SET TYPE=CONSOLE para conseguir o mesmo efeito.

Observe que este comando poderá sobrescrever a configuração do comando SET TYPE=CONSOLE anterior.

9.25 *FILE*

#25

Comando que permite baixar para um terminal um arquivo a partir de um servidor.

Especificação de Interface

<FILE NAME=n ADDR=a ERR=e RC_HTTP=e WRT=w IP=i PORT=p HOST=h SSL=s SSL_METHOD=m SSL_SNI=i TIMEOU
T=t SIZE=s LIN=l COL=c REFRESH_PERIOD=r MD5=m>

- **NAME:** indica o nome do arquivo que será gravado no terminal
- **ADDR:** indica o endereço e o nome do arquivo no servidor
- **ERR:** indica o nome da variável que receberá o resultado da operação
- Possíveis retornos
 - OK
 - NOT_WRT
 - NAME_NOT_INFORMED
 - MD5_ERROR
 - WRITE_ERROR
 - NO_NAME
 - HTTP_ERROR
 - SIZE_ERROR
 - CURL_ERROR
- **RC_HTTP:** indica o nome da variável que receberá o retorno HTTP da operação
- **WRT:** indica se o arquivo é para ser sobrescrito no terminal
- 0, N ou NAO para não sobrescrever
- 1, S ou SIM para sobrescrever
- **IP, PORT, HOST:** para usar um servidor diferente do definido na configuração do NAVS
- **SSL, SSL_METHOD, SSL_SNI:** indicadores de uso de HTTPS (ver comando POST)
- **TIMEOUT:** indica em segundos o tempo para esperar a cada iteração com o servidor
- **SIZE:** indica o tamanho esperado para o arquivo. Caso o recebido seja maior que esse valor, o comando é interrompido com erro
- **LIN, COL:** indica a linha e coluna para que o comando mostre mensagem sobre o andamento da recepção do arquivo
- **REFRESH_PERIOD:** indica o intervalo de tempos em ms (mili segundos) para que a mensagem de andamento da recepção do arquivo seja atualizado. Caso não seja informado, é assumido 1s (1000 ms)
- **MD5:** valor original do MD5 do arquivo para que o comando cheque o recebido automaticamente.

Especificação de Interface

9.26 SLEEP

#26

Suspende a captura de dados pelo terminal para permitir a economia de energia.

Esse comando é executado também internamente conforme a configuração específica do NAVS.

`<SLEEP MODE={IMAGE|TEXT} CONTENT=c>`

- **MODE:** indica se o modo utilizado será imagem ("IMAGE") ou texto ("TEXT").
- **CONTENT:** indica, no caso da utilização de imagem, o nome do arquivo utilizado, e no caso da utilização de texto, o texto que deve ser mostrado na tela (coordenadas 0 x 0).

9.27 GPRS_SWITCH_SIM

#27

Efetua a troca do chip GPRS (*SIM card*) ativo

`<GPRS_SWITCH_SIM x>`

Onde **x** pode ser:

- **SIM1:** chip presente no *slot 1*
- **SIM2:** chip presente no *slot 2*
- **OTHER:** inverte o *slot*
- **DEFAULT:** o mesmo que **SIM1**

Importante: Usar o comando **GET TYPE=SIM_SLOTS** para certificar-se se o terminal possui mais de um slot para chip GPRS.

9.28 SELECT

#28

Mostra menu de seleção de opções a partir de lista separada por vírgulas.

```
<SELECT LIN=1 COL=1 SIZE=s QTD=q UP=u DOWN=d RIGHT=r LEFT=l NAME=n TYPE_RETURN=t INDEX=n1,n2,n3,
..,nX>
item1,
item2,
item3,
...,
itemn
</SELECT>
```

- **LIN:** linha inicial em que a lista será mostrada

Especificação de Interface

- **COL:** coluna inicial em que a lista será mostrada
- **SIZE:** Tamanho da máscara de seleção que será mostrada em inverso
- **QTD:** quantidade de itens a serem mostradas por vez na tela
- **UP:** tecla de navegação para cima na lista
- **DOWN:** tecla de navegação para baixo na lista
- **LEFT:** tecla de navegação para esquerda na lista
- **RIGHT:** tecla de navegação para direita na lista
- **NAME:** nome da variável que receberá a seleção
- **TYPE_RETURN:** tipo de retorno que será provido, sendo:
 - **1** - o próprio item selecionado da lista
 - **2** - o índice do item selecionado da lista partindo de 1
 - **3** - o item correspondente do parâmetro **INDEX** descrito abaixo
- **INDEX:** lista de retornos separados por vírgulas para o tipo de retorno **3**

Código de Teclas:

As teclas podem ser indicadas de **0** a **9** para o uso do teclado.

As teclas usadas no NAVS1 serão convertidas para o uso no NAVS2 conforme o seguinte:

- Para as teclas **B0** a **B9** serão usadas as teclas de **0** a **9**
- Para a tecla **AA** será usada a tecla ***** e para a tecla **A3** será usada a tecla **#**
- Para as teclas **E1** e **FA** será usada a tecla de **2**
- Para as teclas **E2** e **FB** será usada a tecla de **8**
- Para as teclas **E3** e **FC** será usada a tecla de **4**
- Para as teclas **E4** e **FD** será usada a tecla de **6**

COMO DEFAULT, SERÃO CONSIDERADAS AS SEGUINTESS PARAMETRIZAÇÕES DO COMANDO:

- **UP=2 DOWN=8 LEFT=4 RIGHT=6**

Exemplos:

```
<set type=console bgcolor=preto color=rgb(100,50,0)>
<CONSOLE>Escolha a opcao: 2/8/4/6</CONSOLE>
<SELECT name=SELECT lin=3 col=2 qtd=5 size=30 type_return=3 up=2 down=8 left=4 right=6 index=01
,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16>
a-01234567890123456789012345,
b-01234567890123456789012345,
c-01234567890123456789012345,
d-01234567890123456789012345,
```

Especificação de Interface

```
e-01234567890123456789012345,
f-01234567890123456789012345,
g-01234567890123456789012345,
h-01234567890123456789012345,
i-01234567890123456789012345,
j-01234567890123456789012345,
k-01234567890123456789012345,
l-01234567890123456789012345,
m-01234567890123456789012345,
n-01234567890123456789012345,
o-01234567890123456789012345,
p-01234567890123456789012345,
</SELECT>
<set type=console default>
```

9.29 PRNFEED

#29

Avança o papel da impressora de forma a facilitar o destaque

```
<PRNFEED>
```

9.30 GENERATE_QR_CODE

#30

Gera, imprime e/ou salva e mostra na tela um código de barras bidimensional no padrão QR.

```
<GENERATE_QR_CODE SIZE=s QR_ECLEVEL=1 [KEEP_FILE=0|1] [SPACES=s] [FILE_NAME=f] [NO_PRINT=0|1] [D
ISPLAY=0|1] [X=x] [Y=y] [ERR_QR=v] [RC_NAME=v]>
texto a codificar
</GENERATE_QR_CODE>
```

- **SIZE:** espessura do traço a ser usado em número de pixels
- **QR_ECLEVEL:** controla a resolução do do QR code gerado (de 0 a 4)
- **KEEP_FILE:** Indica se o arquivo gerado é para permanecer no terminal (1=sim, 0=não - default)
- **SPACES:** quantidade de pixels a ser deixado como espaço à esquerda da impressão do QR code (default=0)
- **FILE_NAME:** nome do arquivo de imagem a ser salvo (bmp)
- **NO_PRINT:** determina se não é para imprimir (1=não imprime, 0=imprime - default)
- **DISPLAY:** determina se é para mostrar o QR code na tela (1=mostra, 0=não mostra - default)
- **X:** no caso de display, qual a coordenada horizontal da imagem
- **Y:** no caso de display, qual a coordenada vertical da imagem

Especificação de Interface

- **ERR_QR**: nome da variável que receberá o retorno do comando (1=ok, 0=erro)
 - **RC_NAME** - nome da variável que receberá o código de retorno mais preciso do comando, sendo zero para sem erro
-

9.31 **CODE128**

#31

Gera e imprime um código de barras no padrão CODE 128.

```
<CODE128 WIDTH=w HEIGHT=h [ORIENTATION=HORIZONTAL|VERTICAL] [SPACES=s] [ERROR=v] [RC_NAME=v]>
texto a codificar
</CODE128>
```

- **WIDTH**: espessura do traço a ser usado em número de pixels
 - **HEIGHT**: altura em pixels do código de barras gerado
 - **ORIENTATION**: orientação da impressão (HORIZONTAL (default) ou VERTICAL)
 - **SPACES**: quantidade de pixels a ser deixado como espaço à esquerda da impressão (default=0)
 - **ERROR**: nome da variável que receberá o retorno do comando (1=ok, 0=erro)
 - **RC_NAME** - nome da variável que receberá o código de retorno mais preciso do comando, sendo zero para sem erro
-

9.32 **CODE_BAR_3_9**

#32

Gera e imprime um código de barras no padrão CODE 3 de 9 na vertical como padrão.

```
<CODE_BAR_3_9 HEIGHT=h WIDTH=w [ORIENTATION=VERTICAL|HORIZONTAL] [SPACES=s] [ERROR=v] [RC_NAME=v]
]>
texto a codificar
</CODE_BAR_3_9>
```

- **HEIGHT**: espessura do traço a ser usado em número de pixels
 - **WIDTH**: largura em pixels do código de barras gerado
 - **ORIENTATION**: orientação da impressão (VERTICAL(default) ou HORIZONTAL)
 - **SPACES**: quantidade de pixels a ser deixado como espaço à esquerda da impressão (default=0)
 - **ERROR**: nome da variável que receberá o retorno do comando (1=ok, 0=erro)
 - **RC_NAME** - nome da variável que receberá o código de retorno mais preciso do comando, sendo zero para sem erro
-

Especificação de Interface

9.33 CODEBAR

#33

Gera e imprime um código de barras no padrão EAN-13 na horizontal.

<CODEBAR VALUE=v HEIGHT=h SIZE=s SPACES=s] [RC_NAME=v]>

- **HEIGHT:** altura em pixels do código de barras gerado
 - **SIZE:** espessura do traço a ser usado em número de pixels
 - **SPACES:** quantidade de pixels a ser deixado como espaço à esquerda da impressão (default=0)
 - **RC_NAME** - nome da variável que receberá o código de retorno mais preciso do comando, sendo zero para sem erro
-

9.34 CODE_BAR_2_5

#34

Gera e imprime um código de barras no padrão ITF na horizontal.

<CODE_BAR_2_5 VALUE=v HEIGHT=h WIDTH=s SPACES=s] [ERROR=e] [RC_NAME=v]>

- **HEIGHT:** altura em pixels do código de barras gerado
 - **WIDTH:** espessura do traço a ser usado em número de pixels
 - **SPACES:** quantidade de pixels a ser deixado como espaço à esquerda da impressão (default=0)
 - **ERROR:** nome da variável que receberá o retorno do comando (1=ok, 0=erro)
 - **RC_NAME** - nome da variável que receberá o código de retorno mais preciso do comando, sendo zero para sem erro
-

9.35 CTLSWRITE

#35

Grava blocos de dados em cartão Mifare. Veja esclarecimentos sobre o uso do Mifare no capítulo **Mifare: gravação e leitura de dados**

<CTLSWRITE [DATABLOCK=b] [SECTOR=s] [BLOCK=b] [BCOUNT=c] [KEY=k] [RC_NAME=v]>

dados em hexadecimal em múltiplos de 32 dígitos

</CTLSWRITE>

- **DATABLOCK:** número do bloco de dados a partir de 0 até 40 (default=0)
- **SECTOR:** número do setor do cartão
- **BLOCK:** número do bloco de dados dentro do setor
- **BCOUNT:** número de blocos a serem gravados (default=1)

Especificação de Interface

- **REMOVETABLE**: Apaga os dados da tabela carregada para uma ou todas as adquirentes
- **GETCARD**: Solicita o cartão e captura seus dados
- **REMOVECARD**: Solicita e espera a retirada do cartão
- **NAME**: indica o nome da variável que receberá a resposta
- **RAW**: se igual a 1 indica que os dados passados para o comando estão na forma bruta; se igual a 0 (default) indica que os dados estão na forma aberta
- **REDEADQ**: indica qual rede adquirente usar no comando. Deve ter dois dígitos. Se informado 00 (default) significa todas as redes adquirentes.
- **RC_NAME**: nome da variável que receberá o código de retorno do comando, sendo zero para sem erro

Os parâmetros a serem informados para o comando dependem da função a ser executada e do parâmetro **RAW** passado:

- **OPEN**: Sem parâmetros
- **CLOSE**: Sem parâmetros

-
- **GETINFO, GETTIMESTAMP e REMOVETABLE**:
 - **RAW=1**
 - **INPUT=aa** onde **aa** é a rede adquirente ou 00 para informações gerais ou todas as adquirentes
 - **RAW=0**
 - Os dados vêm do parâmetro **REDEADQ**

-
- **LOADTABLE**:
 - **RAW=1**
 - **INPUT=aaDDMMAAAAss**, onde:
 - **aa** é a rede adquirente ou 00 para todas
 - **DDMMAAAA** é o *timestamp* para a tabela
 - **ss** é um número sequencial para o referido *timestamp*
 - **INPUT2=nnSSS...**,
 - **INPUT2=nnSSS...**,
 - **INPUT2=nnSSS...** onde:
 - **nn** é a quantidade de registros nessa entrada normalmente 01
 - **SSS** é o tamanho do registro a seguir

Especificação de Interface

- **APPVERSION1**: *Application Version Number* (Terminal) - opção #1
- **APPVERSION2**: *Application Version Number* (Terminal) - opção #2
- **APPVERSION3**: *Application Version Number* (Terminal) - opção #3
- **TERMCOUNTRYCODE**: *Terminal Country Code*, **076** para o Brasil
- **TRANSCURRENCYCODE**: *Transaction Currency Code*, **986** para o Real (R\$)
- **TRANSCURRENCYEXP**: *Transaction Currency Exponent*
- **MERCHANTID**: *Merchant Identifier*
- **MERCHANTCODE**: *Merchant Category Code*
- **IDTERM**: identificação do terminal
- **CAPABILITIES**: Mapa de bits indicando as capacidades do terminal, com o formato:
 - abcxxxxx defgkxxx hijmxxxx, sendo:
 - a = Digitação de número do cartão.
 - b = Cartão magnético.
 - c = Cartão com chip de contatos.
 - d = Verificação em cartão com chip de PIN “em aberto”.
 - e = Verificação online de PIN com criptografia.
 - f = Assinatura em papel.
 - g = Verificação em cartão com chip de PIN com criptografia.
 - k = Aceita o método “No CVM” (sem verificação do portador).
 - h = SDA - Autenticação estática de dados do cartão com chip.
 - i = DDA - Autenticação dinâmica de dados do cartão com chip.
 - j = Captura de cartão.
 - m = Autenticação offline CDA.
- **ADDCAPABILITIES**: Capacidades adicionais do terminal
- **TERMTYPE**: tipo de terminal:
- **TACDEFAULT**: *Terminal Action Code – Default*
- **TACDENIAL**: *Terminal Action Code – Denial*
- **TACONLINE**: *Terminal Action Code – Online*
- **FLOORLIMIT**: *Terminal Floor Limit*
- **TRANSCATCODE**: *Transaction Category Code*
- **CTLSACTION**: ação para cartão com chip sem contato se o valor da transação estiver zerado:

Especificação de Interface

- 0 = Não suporta;
- 1 = Suporta, porém somente online.
- **CTLSAIDCAP**: Capacidade de tratamento do terminal para o referido AID, para CTLS:
 - 0 = Não suporta;
 - 1 = Suporta VISA MSD;
 - 2 = Suporta VISA qVSDC;
 - 3 = Suporta MasterCard PayPass Mag Stripe;
 - 4 = Suporta MasterCard PayPass M/Chip;
 - 5 = Suporta Amex Expresspay Magstripe Mode; e
 - 6 = Suporta Amex Expresspay EMV Mode.
- **CTLSLIMIT**: *CTLS Transaction Limit*
- **CTLSFLOORLIMIT**: *CTLS Floor Limit*
- **CVMLIMIT**: *CVM Required Limit*
- **APPMAGVERSION**: *PayPass Mag Stripe Application Version Number*
- **CTLSAPPSELECT**: forma de seleção da aplicação CTLS:
 - 0 = A aplicação é selecionada automaticamente pela prioridade; ou
 - 1 = Deve ser mostrado menu de seleção caso exista outra aplicação compatível.
- **TDOLDEFAULT**: *TDOL*
- **DDOLDEFAULT**: *DDOL*
- **OFFLINEAUTHRESPCODE**: fixo **Y1Z1Y3Z3**
- **CTLSTACDEFAULT**: *CTLS Terminal Action Code – Default*
- **CTLSTACDENIAL**: *CTLS Terminal Action Code – Denial*
- **CTLSTACONLINE**: *CTLS Terminal Action Code – Online*

IMPORTANTE: Nenhum campo deve deixar de ser informado mesmo se o mesmo não tenha valor sob pena de formar registros com valores inválidos

- Exemplo:
- `<emv_bc func=loadtable name=RESP rc_name=RC1 raw=0 retheadq=00>`
`timestamp=2408202000,`

`REGISTROSIZE=284,`
`IDTABLE=1,`
`IDREGISTER=01,`
`AIDSIZE=07,`
`AID=A0000001544442000000000000000000,`
`TIPOAPP=02,`
`LABEL='Banrisul',`

Especificação de Interface

- **RAW=0**
 - **REDEADQ** - indica a rede adquirente ou 00 para todas. Caso não informado, o comando usa o parâmetro REDEADQ do corpo do comando, ou assume 00 caso nenhum for informado
 - **TIPOAPP** -tipo de aplicação (01=crédito, 02=débito etc). Para qualquer aplicação, informar 99, para selecionar de uma lista, usar 00.
 - **VALOR** - valor em centavos com 12 bytes
 - **DATA** - data da transação no formato AAMMDD
 - **HORA** - hora da transação no formato HHMMSS
 - **TIMESTAMP** - *timestamp* da tabela carregada no formato DDMMAAAASS. Usar a função **GETTIMESTAMP** para obtê-lo.
 - **QTDENTRADAS** - quantidade de entradas de 4 bytes a seguir para selecionar redes adquirentes específicas e *AIDs* de aplicações. Caso deseje usar a tabela completa, informar 00.
 - **AAIL** - repetir esse parâmetro para a quantidade informada. **AA** é a rede adquirente e **II** é o índice na tabela carregada
 - **CTLS** - habilita o cartão sem contato (1=sim, 0=não)
 - Exemplo:
 - ```
<emv_bc func=getcard name=RESP rc_name=RC raw=0>
REDEADQ=00,
TIPOAPP=01,
VALOR=000000000010,
DATA=200818,
HORA=110000,
TIMESTAMP=1408202000,
QTDENTRADAS=02,
AAIL=0102,AAIL=0103,
CTLS=1
</emv_bc>
```

- 
- **GETPIN:**
  - **RAW=1**
    - **INPUT='mllww...wwLLpp...ppQssSSmsg1... msg2...'**
    - Descrição do parâmetro:
      - **m** - modo de criptografia, sendo:
        - “0” - Master Key / Working DES (8 bytes)
        - “1” - Master Key / Working 3DES (16 bytes)



## Especificação de Interface

- “2” - DUKPT DES
- “3” - DUKPT 3DES
- **II** - índice da Master Key ou da DUKPT
- **ww...ww** - Para os modos “0” e “1”, é a Working Key (criptografada pela Master Key) com 32 bytes hexadecimais
- **LL** - tamanho do PAN de 13 a 19
- **pp...pp** - PAN de 19 bytes alinhado à esquerda completados com zeros à esquerda
- **Q** - quantidade capturas a seguir, de 0 a 9
  - **ss** - tamanho mínimo do dado a ser capturado
  - **SS** - tamanho máximo do dado a ser capturado
  - **msg1** - mensagem 1 a ser usada na captura (16 bytes)
  - **msg2** - mensagem 2 a ser usada na captura (16 bytes)
- Exemplo:

```

• <emv_bc func=getpin name=RESP rc_name=RC raw=1>
 INPUT= '301 13000000000000000000010412Digite a
 sua SENHA '
 </emv_bc>

```

- **RAW=0**
- **MODOCRIPTO** - modo de criptografia
- **IDDUKPT** - índice da Master Key ou da DUKPT
- **WORKINGKEY** - Working Key
- **PANSIZE** - tamanho do PAN de 13 a 19
- **PAN** - PAN de 19 bytes alinhado à esquerda completados com zeros à esquerda
- **QTDCAPTURAS** - quantidade capturas a seguir, de 0 a 9
  - **MINSIZE** - tamanho mínimo do dado a ser capturado
  - **MAXSIZE** - tamanho máximo do dado a ser capturado
  - **MSG1** - mensagem 1 a ser usada na captura (até 16 bytes)
  - **MSG2** - mensagem 2 a ser usada na captura (até 16 bytes)
- Exemplo:

```

– <emv_bc func=getpin name=RESP rc_name=RC1 raw=0>
 MODOCRIPTO=3,
 IDDUKPT=01,
 WORKINGKEY=,
 PANSIZE=13,
 PAN=0000001234567890123,

```

## Especificação de Interface

---

```

QTDCAPTURAS=1,
MINSIZE=04,
MAXSIZE=12,
MSG1='Entre com a ',
MSG2='sua Senha '
</emv_bc>

```

- **GOONCHIP:**
- **RAW=1**
  - **INPUT='vv...vvTT...TTnCcMiiWW...WWrFFFFFFFppTTTTTTTTmmSSS...'** ,
  - **INPUT2='ssst...tt'**
  - Descrição dos parâmetros:
    - **vv...vv** - novo valor da transação em centavos
    - **TT...TTT** - valor do troco em centavos
    - **n** - resultado da consulta à lista negra
    - **C** - obrigatoriedade de conexão
    - **p** - requer verificação de PIN
    - **M** - modo de criptografia, sendo:
      - “0” - Master Key / Working DES (8 bytes)
      - “1” - Master Key / Working 3DES (16 bytes)
      - “2” - DUKPT DES
      - “3” - DUKPT 3DES
    - **ii** - índice da Master Key ou da DUKPT
    - **WW...WW** - Para os modos “0” e “1”, é a Working Key (criptografada pela Master Key) com 32 bytes hexadecimais
    - **r** - gerenciamento de risco
    - **FFFFFFF** - *Terminal Floor Limit* (8 bytes)
    - **pp** - *Target Percentage*
    - **TTTTTTTT** - *Threshold Value* (8 bytes)
    - **mm** - *Maximum Target Percentage*
    - **SSS** - Tamanho dos dados específicos a seguir (3 bytes)
    - **...** - Parâmetros específicos da rede adquirente
    - **sss** - tamanho total das TAGs a seguir em bytes

## Especificação de Interface

- **tt...tt** - TAGs para montagem do campo 55 da mensagem ISO8583 em hexadecimal (tamanho = o dobro de sss)

- Exemplo:

```
- <emv_bc func=goonchip name=RESP rc_name=RC raw=1>
 INPUT='0000000001000000000000000011301
 000000000000000000000000',
 INPUT2=0045F205F28
</emv_bc>
```

0

- **RAW=0**

- Descrição dos parâmetros:

- **VALOR** - novo valor da transação em centavos
- **TROCO** - valor do troco em centavos
- **LISTANEGRA** - resultado da consulta à lista negra
- **COMCONEXAO** - obrigatoriedade de conexão
- **COMPIN** - requer verificação de PIN
- **MODOCRIPTO** - modo de criptografia, sendo:
  - "0" - Master Key / Working DES (8 bytes)
  - "1" - Master Key / Working 3DES (16 bytes)
  - "2" - DUKPT DES
  - "3" - DUKPT 3DES
- **IDDUKPT** - índice da Master Key ou da DUKPT
- **WORKINGKEY** - Para os modos "0" e "1", é a Working Key (criptografada pela Master Key) com 32 bytes hexadecimais
- **RISCO** - gerenciamento de risco
- **FLOORLIMIT** - Terminal Floor Limit (8 bytes)
- **PERCENT** - Target Percentage
- **THRESHOLD** - Threshold Value (8 bytes)
- **MAXPERCENT** - Maximum Target Percentage
- **SIZEDADOS** - Tamanho dos dados específicos a seguir (3 bytes)
- **PARAMS** - Parâmetros específicos da rede adquirente
- **SIZETAGS** - tamanho total das TAGs a seguir em bytes
- **TAGS** - TAGs para montagem do campo 55 da mensagem ISO8583 em hexadecimal (tamanho = o dobro de SIZETAGS)
- Exemplo:

## Especificação de Interface

- `<emv_bc func=goonchip name=RESP rc_name=RC raw=0>`  
 VALOR=00000000100,TROCO=00000000000,  
 LISTANEGRA=0,  
 COMCONEXAO=1,  
 COMPIN=1,  
 MODOCRIPTO=3, IDDUKPT=01, WORKINGKEY=,  
 RISCO=0,  
 FLOORLIMIT=00000000, PERCENT=00,  
 THRESHOLD=00000000, MAXPERCENT=00,  
 SIZEDADOS=000', PARAMS=,  
 SIZETAGS=004, TAGS=5F205F28  
`</emv_bc>`

- **FINISHCHIP:**
- **RAW=1**
  - **INPUT='SeAASSVv...vvSSS...'**,
  - **INPUT2='ssstt...tt'**
  - Descrição dos parâmetros:
    - **S** - status da comunicação com o *Host*
    - **e** - tipo de emissor
    - **AA** - *Athorization Response Code*
    - **SSS** - tamanho em bytes dos dados do bit 55 (3 bytes)
    - **vv...vv** - dados do bit55 no formato TLV (em hexadecimal)
    - **SSS** - Tamanho dos dados específicos a seguir (3 bytes)
    - **...** - Parâmetros específicos da rede adquirente
    - **sss** - tamanho total das *TAGs* a seguir em bytes
  - **tt...tt** - *TAGs* para montagem dos dados finais em hexadecimal (tamanho = o dobro de sss)
- Exemplo:
 

```
<emv_bc func=finishchip name=RESP rc_name=RC raw=1>
INPUT= '00000109108330D56C80029FC3A000',
INPUT2=0045F205F28
</emv_bc>
```
- **RAW=0**
- Descrição dos parâmetros:
  - **STATUSHOST** - status da comunicação com o *Host*
  - **TIPOEMISSOR** - tipo de emissor
  - **AUTHRC** - *Athorization Response Code*

## Especificação de Interface

---

- **SIZEBIT55** - tamanho em bytes dos dados do bit 55 (3 bytes)
- **BIT55** - dados do bit55 no formato TLV (em hexadecimal)
- **SIZEDADOS** - Tamanho dos dados específicos a seguir (3 bytes)
- **PARAMS** - Parâmetros específicos da rede adquirente
- **SIZETAGS** - tamanho total das TAGs a seguir em bytes
- **TAGS** - TAGs para montagem dos dados finais em hexadecimal (tamanho = o dobro de sss)
- Exemplo:
- ```
<emv_bc func=finishchip name=RESP rc_name=RC raw=1>
STATUSHOST=0,
TIPOEMISSOR=0,
AUTHRC=00,
SIZEBIT55=010,BIT55=9108330D56C80029FC3A
SIZEDADOS=000,PARAMS=,
SIZETAGS=004,TAGS=5F205F28
</emv_bc>
```

-
- **REMOVECARD:**
 - **RAW=1**
 - **INPUT='mensagem 1 mensagem 2 '**
 - Descrição do parâmetro:
 - São duas mensagens de, exatamente, 16 bytes cada uma concatenadas
 - Usar aspas para delimitar o parâmetro
 - Exemplo:
 - ```
<emv_bc func=removecard name=RESP rc_name=RC raw=1>
INPUT='Retire o Cartao Imediatamente..'
</emv_bc>
```
  - **RAW=0**
    - **MSG1=mensagem 1 ,**
    - **MSG2=mensagem 2**
    - Descrição dos parâmetros:
    - São duas mensagens de, no máximo, 16 bytes cada uma
    - Usar aspas para delimitar o parâmetro caso haja espaço em branco nele
    - Exemplo:
      - ```
<emv_bc func=removecard name=RESP rc_name=RC raw=0>
MSG1='Retire o Cartao',
```


Especificação de Interface

```
REDEADQ=x AID=y TIPOAPP=z LABEL=1;
</EMV_ID>
```

- **NAME:** indica o nome da variável que receberá a resposta, no caso, os dados do cartão
- **RC_NAME:** nome da variável que receberá o código de retorno do comando, sendo zero para sem erro
- **NOWAIT:** indica, caso informado o valor **1**, se o comando não espera o usuário retirar o cartão

Os parâmetros a serem informados para o comando são os seguintes:

- **REDEADQ:** Informa o código, com 2 bytes, da rede adquirente
- **AID:** *Application Identifier* usado para identificar o tipo de cartão e a aplicação dele
- **TIPOAPP:**
- - **01** - Crédito
 - **02** - Débito
- **LABEL:** Rótulo para a AID com até 16 bytes
-

– Exemplo

```
<EMV_ID NAME=RESP RC_NAME=RC NOWAIT=1>
REDEADQ=4 AID=A0000000031010 TIPOAPP=2 LABEL=DEBVISA;
REDEADQ=4 AID=A0000000041010 TIPOAPP=2 LABEL=DEBMASTER;
REDEADQ=4 AID=A00000004942010 TIPOAPP=2 LABEL=DEBELO;
REDEADQ=4 AID=F0000000030001 TIPOAPP=2 LABEL=BRDESCARD;
REDEADQ=4 AID=A00000004271010 TIPOAPP=2 LABEL=DEBHIPER;
REDEADQ=4 AID=A0000000250000 TIPOAPP=2 LABEL=AMEX,
REDEADQ=4 AID=A00000002501 TIPOAPP=2 LABEL=AMEX;
REDEADQ=4 AID=A0000001544442 TIPOAPP=2 LABEL=BANRISUL;
REDEADQ=02 AID=A0000000041010 TIPOAPP=01 label=Credito;
REDEADQ=02 AID=A0000000041010 TIPOAPP=02 label=Debito;
REDEADQ=02 AID=A0000000031010 TIPOAPP=01 label=Credito;
REDEADQ=02 AID=F0000000030001 TIPOAPP=01 label=Credito;
REDEADQ=02 AID=A00000000305076010 TIPOAPP=01 label=Credito;
REDEADQ=02 AID=A00000000305076010 TIPOAPP=02 label=Debito;
REDEADQ=02 AID=F0000000030001 TIPOAPP=02 label=Debito;
REDEADQ=02 AID=A0000000031010 TIPOAPP=02 label=Debito;
REDEADQ=02 AID=A0000000032010 TIPOAPP=02 label=Debito;
REDEADQ=02 AID=A0000000043060 TIPOAPP=02 label=Debito;
</EMV_ID>
```

9.40 PAGS

#40

Especificação de Interface

Executa a nova modalidade de pagamento pelo NAVS

```
<PAGS ENGINE=eng ENV=env VAR_RC_DESCR=erd VAR_RC=er SERVER_NAME=serv PAYLOG_NAME=n RESPAG_NAME=r
>
dados de pagamento
</PAGS>
```

- **ENGINE:** Indica qual mecanismo de pagamento a ser usado, podendo atualmente ser apenas **ISO_APSK**
- **ENV:** Indica o ambiente presente no arquivo de configuração PAGS.CFG, podendo ser: **SANDBOX**, **HOMOLOGACAO** ou **PRODUCAO**
- **SERVER_NAME:** É possível no arquivo de configuração PAGS.CFG indicar mais de um servidor no mesmo ambiente. Este parâmetro indica qual o servidor usar na execução do comando. O conteúdo desse parâmetro deve estar de acordo com o valor informado no arquivo de configuração
- **PAYLOG_NAME** e **RESPAG_NAME:** Indicam, respectivamente, os nomes dos arquivos de LOG (PAYLOG_NAME) e RESULTADO (RESPAG_NAME). O Arquivo de LOG é crescente e contém todas as variáveis de entrada e de saída da execução do comando. O seu expurgo é por conta da aplicação através do comando DELFILE do NAVS. O arquivo de RESULTADO contém sempre a última execução.
- **VAR_RC** e **VAR_RC_DESCR:** Indicam os nomes das variáveis que receberão o *return code* do comando PAGS e sua descrição:

Descrição	RC
ERROR_NO_ERROR	0
ACAO_APROVADO	10
ACAO_NAO_APROVADO	11
ACAO_CONFIRMAR	12
ACAO_RETRY	13
ACAO_EXECUTAR	14
ACAO_NEXT_FUNCTION	15
ERROR_GENERICO	-9
ERROR_NO_RESPONSE	-10
ERROR_NO_HTTP_RESPONSE	-11
ERROR_MISS_VAR	-12
ERROR_MISS_HEADER_RESPONSE	-13
ERROR_MISS_VAR_OR_ZERO	-14
ERROR_MISS_CONFIG	-15
ERROR_MISS_CONFIG_OR_ZERO	-16
ERROR_NOT_JSON	-17

Especificação de Interface

Descrição	RC
ERROR_JSON_CREATE	-18
ERROR_JSON_PARSE	-19
ERROR_HTTP_400	-20
ERROR_HTTP_401	-21
ERROR_HTTP_403	-22
ERROR_HTTP_404	-23
ERROR_HTTP_OUTRO	-24
ERROR_COMUNICACAO	-25
ERROR_PACK	-26
ERROR_UNPACK	-27

Arquivo de configuração PAGES.CFG

Deve ser carregado no terminal um arquivo texto com a configuração do comando PAGES com o nome de *PAGES.CFG*. Esse arquivo é um arquivo JSON com o seguinte formato:

```
{
  "connection" :
  {
    "time-out" : 30
  },
  "iso_apsk":
  {
    "sandbox":
    {
      "endpoint" : "192.168.0.111",
      "port"      : "2048"
    }
  },
  "dtef" :
  {
    "models":
    [
      {
        "name"   : "C680",
        "pin_x"  : 24,
        "pin_y"  : 180
      },
      {
        "name"   : "VX690",
        "pin_x"  : 24,
        "pin_y"  : 180
      }
    ],
    "tp_comm" : 1,
    "autom_name": "IPDEMO",
    "autom_version": "0.0.1",
  }
}
```

Especificação de Interface

```

    "sandbox" :
    [
      {
        "name" : "desenvolvimento",
        "URL" : "https:// 192.168.0.112",
        "password" : "01234567890ABCDEF01234567890ABCDEF"
      },
      {
        "name" : "desenvolvimento1",
        "URL" : "https:// 192.168.0.113",
        "password" : "01234567890ABCDEF01234567890ABCDEF"
      }
    ]
  }
}

```

Para cada ambiente, é possível definir vários servidores que são diferenciados pela entrada **name**. Para indicar o servidor correto, usar, além do parâmetro **ENV**, o parâmetro **SERVER_NAME** com o **name** da configuração. Caso seja necessário mais de um ambiente, incluir o ambiente **PRODUCAO/HOMOLOGACAO/SANDBOX** sob o item **ISO_APSK**.

9.41 GET TYPE

9.41.1 CARD

#101

Capturar os dados do cartão magnético

<GET TYPE=CARD NAME=var TRACKS=t>

- **NAME**: indica o nome da variável que receberá a resposta
- **TRACKS**: indica quais trilhas serão devolvidas conforme a tabela abaixo:

TRACKS	Trilha 1	Trilha 2	Trilha 3
1	X		
2		X	
3	X	X	
4			X
5	X		X
6		X	X
7	X	X	X

outro valor

Especificação de Interface

TRACKS	Trilha 1	Trilha 2	Trilha 3
ausente	X	X	X

9.41.2 MODEL

#102

Capturar o modelo do terminal POS

<GET TYPE=MODEL NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

9.41.3 SERIALNO

#103

Capturar o modelo do terminal POS

<GET TYPE=SERIALNO NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

9.41.4 STSCOMM - Deprecado - usar COMM_STATUS

#104

Capturar o status da comunicação

<GET TYPE=STSCOMM NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

Resposta	Significado
CX..	Conectando na rede.
Rdok:	Rede OK, indica conexão física com sucesso.
StNw	Inicia conexão lógica.
Cxok	Conexão lógica com sucesso, conectado com a operadora.
Wfok	Encontrou a rede Wi-Fi.
Htok	Conectado com o Servidor.
HOST	Servidor OK.
Hoff	Servidor offline.
RX..	Recebendo dados.
Rxok	Dados do Servidor recebidos.

Especificação de Interface

Resposta	Significado
TX..	Enviando dados ao Servidor.
CXer	Erro de conexão.
DNSe	Erro de DNS, tentando encontrar página.
....	Reconectando na rede.

9.41.5 COMM_STATUS

#105

Recupera o status da comunicação.

Capturar o status da comunicação

<GET TYPE=COMM_STATUS NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

Resposta	Significado
0	UP
1	STOP
2	START
3	INIT
4	LINK
5	OPEN
-1	FAIL

9.41.6 HIDDEN

#106

Passar valores por uma variável de uma iteração à próxima.

<GET TYPE=HIDDEN NAME=v VALUE=v>

- **NAME:** Nome da variável que receberá o valor passado
 - **VALUE:** valor a ser passado na variável
-

9.41.7 VAR

#107

Especificação de Interface

Recupera o valor de uma variável da configuração do NAVS

<GET TYPE=VAR VARNAME=v NAME=v>

- **NAME:** Nome da variável que receberá o conteúdo da variável solicitada.
- **VARNAME:** Nome da variável da configuração do NAVS.

Variáveis usadas pelo NAVS

Variável	Conteúdo	Valores possíveis
COMM_METHOD	Método primário de comunicação	G=GPRS, W=WiFi, E=Ethernet, O=Offline
COMM_METHOD2	Método secundário de comunicação	G=GPRS, W=WiFi, E=Ethernet, O=Offline
COMM_GPRS_APN	APN do chip usado	
COMM_GPRS_USR	Usuário da APN	
COMM_GPRS_PWD	Senha do usuário da APN	
COMM_GPRS_APN2	APN do chip 2 usado	
COMM_GPRS_USR2	Usuário da APN do chip 2	
COMM_GPRS_PWD2	Senha do usuário da APN do chip 2	
COMM_WIFI_SSID	SSID do Access Point WiFi	
COMM_WIFI_KEY	Chave de acesso ao Access Point WiFi	
COMM_WIFI_DHCP	Se usa ou não DHCP para WiFi	1=Sim, 0=Não (IP estático)
COMM_WIFI_IP	IP Estático para WiFi	
COMM_WIFI_MASK	Máscara de rede para IP Estático para WiFi	
COMM_WIFI_GATEWAY	IP do Gateway de rede para IP Estático para WiFi	
COMM_WIFI_DNS1	IP do Servidor DNS1 de rede para IP Estático para WiFi	
COMM_WIFI_DNS2	IP do Servidor DNS2 de rede para IP Estático para WiFi	
POST_IP	Endereço IP para o comando POST	
POST_PORT	Port para o comando POST	
POST_HOST	Header HOST para o comando POST	
POST_RESOURCE	Recurso a ser acessado pelo comando POST	

Especificação de Interface

Variável	Conteúdo	Valores possíveis
POST_SSL	Indicador de protocolo HTTPS para o comando POST	1=Sim, 0=Não
POST_SSL_METHOD	Método de criptografia a ser usado no comando POST	1=SSSV3, 2=TLSv1, 3=TLSv1.1, 4=TLSv1.2
POST_SSL_SNI	Valor a ser usado como SNI (Server Name Indication) para protocolo HTTPS	
POST_TIMEOUT	Tempo máximo em segundos de espera para conclusão do comando POST	
LOL_IP	Endereço IP para o licenciamento online	
LOL_PORT	Port para o licenciamento online	
LOL_HOST	Header HOST para o licenciamento online	
LOL_RESOURCE	Recurso a ser acessado pelo licenciamento online	
TCPIP_TIMEOUT	Tempo máximo padrão em segundos de espera para conclusão de requisições TCPIP	
PAY_SSL		
PAY_RECIBO		
PAY_PTO_IMP		
PAY_FORCA_DESC		
PRINTER_INVERSE	Indicador de impressão inversa	1=Sim, 0=Não
PRINTER_DOUBLE_WIDTH	Indicador de impressão em largura dupla	1=Sim, 0=Não
PRINTER_DOUBLE_HEIGHT	Indicador de impressão altura dupla	1=Sim, 0=Não
PRINTER_FONT	Nome do fonte de impressão padrão	
PRINTER_WIDTH	Largura em pixels dos caracteres de impressão	
PRINTER_HEIGHT	Altura em pixels dos caracteres de impressão	
PRINTER_OLD_FONTE1	Nome do fonte para o parâmetro FONTE1 do antigo comando CHGPRNFNT	
PRINTER_OLD_FONTE2	Nome do fonte para o parâmetro FONTE2 do antigo comando CHGPRNFNT	
PRINTER_OLD_FONTE3	Nome do fonte para o parâmetro FONTE3 do antigo comando CHGPRNFNT	

Especificação de Interface

Variável	Conteúdo	Valores possíveis
PRINTER_OLD_FONTE4	Nome do fonte para o parâmetro FONTE4 do antigo comando CHGPRNFNT	
PRINTER_OLD_FONTE5	Nome do fonte para o parâmetro FONTE5 do antigo comando CHGPRNFNT	
PRINTER_OLD_FONTE6	Nome do fonte para o parâmetro FONTE6 do antigo comando CHGPRNFNT	
PRINTER_OLD_SIZE2	Tamanho percentual do caractere para o parâmetro SIZE2 do antigo comando CHGPRNFNT	
PRINTER_OLD_SIZE3	Tamanho percentual do caractere para o parâmetro SIZE2 do antigo comando CHGPRNFNT	
PRINTER_OLD_SIZE4	Tamanho percentual do caractere para o parâmetro SIZE2 do antigo comando CHGPRNFNT	

9.41.8 CANCEL_KEY

#108

Recupera se a tecla CANCEL (vermelha) está ou não habilitada

<GET TYPE=CANCEL_KEY NAME=v>

- **NAME:** Nome da variável que receberá a informação se a tecla CANCEL está habilitada, sendo:
 - ENABLE
 - DISABLE

9.41.9 FIELD

#109

Capturar uma entrada numérica, não secreta, por meio do teclado

<GET TYPE=FIELD NAME=v SIZE=s LIN=1 COL=c [NOENTER|NOENTER=ne] [ALPHA|ALPHA=a1] [FILLER=f1] [LABEL=1b] [VALUE=v1] ZL>

Parâmetro	Significado	Valores possíveis
NAME	Nome da variável que receberá o campo capturado	
SIZE	Quantidade de bytes do valor a ser obtido do teclado	
LIN	Linha da console do primeiro byte do campo ou do valor de LABEL	
COL	Coluna da console do primeiro byte do campo ou do valor de LABEL	

Especificação de Interface

Parâmetro	Significado	Valores possíveis
NOENTER	Ao completar a quantidade de dígitos igual ao SIZE informado, o comando retorna ao NAVS com o valor digitado	'1' ou 'S'
ALPHA	Permite a entrada de valores alfanuméricos	'1' ou 'S'
FILLER	Indica o caracter de preenchimento do campo diferente de brancos	
LABEL	Rótulo a ser inserido à esquerda do campo a ser digitado. Será posicionado em LIN e COL informado	
VALUE	Valor a ser preenchido e mostrado previamente no campo a ser digitado	
ZL	Despreza a digitação de zeros não significativos	
MASK	Leia instruções abaixo	

Parâmetro MASK do comando GET TYPE=FIELD

Para captura de entrada com máscara predefinida na tela.

O valor retornado virá sem a máscara.

Sintaxe:

Na máscara podem ser usados os caracteres:

- 9, 0, x, X, a e A, com as seguintes funções:
 - 9 – Só aceita dígitos numéricos.
 - 0 – Só aceita dígitos numéricos e coloca zero onde ainda não estiver preenchido.
 - a – Só aceita caracteres alfabéticos, maiúsculos ou minúsculos.
 - A – Só aceita caracteres alfabéticos, forçando para maiúsculos.
 - x – Aceita qualquer caractere.
 - X – Aceita qualquer caractere, forçando os alfabéticos para maiúsculos.

Alguns caracteres especiais podem ser usados na primeira posição da máscara, para alterar algumas características da entrada de dados:

- * - Digitação protegida. Todos os caracteres digitados aparecerão como asteriscos.
- ! – Enter automático. Ao se digitar o último caractere da máscara, é gerado um Enter.
- D – A digitação é alinhada à direita.
- E – A digitação é alinhada à esquerda (default).
- _ - O espaço para entrada do valor é preenchido com o caractere underline.

Especificação de Interface

- C – AutoClear – Ao editar um campo com um valor já definido, caso a primeira tecla digitada seja uma tecla numérica, o campo é limpo. Caso contrário (um BACKSPACE, ou ENTER, por exemplo) o valor do campo é preservado.

Exemplos:

Digitação de senha: “*999999”

Digitação de data: “99/99/9999”

Escolha de opções: “@Um,Dois,Três,Quatro”

Qualquer outro caractere que apareça na máscara será apresentado na posição onde aparece.

9.41.10 VALUE

#110

Capturar um valor moeda numérica, não secreta, por meio do teclado

O valor mostrado será prefixado com R\$ e preenchido da direita para a esquerda com 2 casas decimais e separação de milhar. Exemplo: R\$1.000.000,00.

```
<GET TYPE=VALUE NAME=v SIZE=s LIN=1 COL=c [NOENTER|NOENTER=ne] [FILLER=f1] [LABEL=1b] [VALUE=v1]>
```

Os parâmetros são os mesmos do comando GET TYPE=FIELD com exceção do argumento ALPHA

9.41.11 IDCTLS ou IDMIFARE

#111

Capturar a identificação de cartão do tipo ISO ou MIFARE Apenas para os modelos de POS que tenham essa interface.

```
<GET TYPE=IDCTLS|IDMIFARE NAME=v>
```

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.12 INSERTCARD

#112

Aguardar a inserção de um cartão sem capturar qualquer informação.

```
<GET TYPE=INSERTCARD>
```

Especificação de Interface

9.41.13 KEY

#113

Captura uma tecla

<GET TYPE=KEY [NAME=v]>

- **NAME:** indica o nome da variável que receberá a resposta

Valores retornados

Valor retornado	Tecla
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
1B	CANCEL (Vermelha) caso o CANCEL_KEY esteja desabilitado
08	BACKSPACE (Amarela)
0D	ENTER (Verde)

9.41.14 WIFI_LIST

#114

Recupera a lista de **Access Points WiFi** disponíveis.

<GET TYPE=WIFI_LIST NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

A lista virá no formato de vetor Lua:

```
{ [1]="AP_1", [2]="AP_2", ..., [n]="AP_n" }
```

9.41.15 ANYKEY - Deprecado - Usar KEY

#115

Especificação de Interface

Aguardar o pressionamento da tecla ENTER

<GET TYPE=ANYKEY>

9.41.16 SPECIALKEY - Deprecado - Usar KEY

#116

Aguardar o pressionamento de teclas especiais. Este comando pode possuir três parâmetros.

Exemplo de uso:

<GET TYPE=SPECIALKEY NAME=RET [NOENTER=1] [NOBKSPC=1]>

- **NAME:** nome da variável que receberá a tecla especial pressionada.
- **NOENTER:** desabilita o pressionamento da tecla ENTER
- **NOBKSPC:** desabilita o pressionamento da tecla BACKSPACE

Valores retornados

TECLA	STRING RETORNADA
FUNC	FA
# (MENU)	A3
BACKSPACE	88
ENTER	8D

9.41.17 VERSION

#118

Recupera a versão do NAVS.

<GET TYPE=VERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.18 DESNBS

#119

Criptografa no padrão DES .

<GET TYPE=DESNBS CODOP=c KEY=k IN=i NAME=v RC_NAME=r>

- **CODOP:** Indica qual operação a executar, sendo:
 - '0' = cifra ou criptografa

Especificação de Interface

- '1' = decifra ou decriptografa
 - **KEY:** chave de criptografia com 8 bytes em hexadecimal (16 caracteres)
 - **IN:** campo a (de)criptografar com 8 bytes em hexadecimal (16 caracteres)
 - **NAME:** indica o nome da variável que receberá a resposta
 - **RC_NAME:** resposta do comando:
 - '00' = comando executado com sucesso
 - '-1' = erro, verifique o conteúdo passado na variável passada em NAME
-

9.41.19 PASS

#120

Capturar um valor moeda numérica, não secreta, por meio do teclado

O valor mostrado será prefixado com cada dígito substituído por '*'

```
<GET TYPE=PASS NAME=v SIZE=s LIN=1 COL=c [NOENTER|NOENTER=ne] [FILLER=f1] [LABEL=1b] [VALUE=v1]>
```

Os parâmetros são os mesmos do comando GET TYPE=FIELD com exceção do argumento ALPHA

9.41.20 EMV

#121

Capturar os dados de um cartão EMV

```
<GET TYPE=EMV NAME=v RC_NAME=r REDEADQ=q AID=a TIPOAPP=t>
```

- **REDEADQ:** Indica a rede adquirente, sendo:
 - 1 (Amex),
 - 2 (Rede),
 - 3 (Cielo)
- **AID:** Identificador da aplicação, exemplo: A0000000031010.
- **TIPOAPP:** Tipo de aplicação:
 - 1 (Crédito),
 - 2 (Débito),
 - 3 (Moedeiro).
- **NAME:** indica o nome da variável que receberá a resposta
- **RC_NAME:** resposta do comando:

Especificação de Interface

- '00' = comando executado com sucesso
 - '-1' = erro nos parâmetros passados
 - Outro valor: erro nas funções internas EMV. A função e o erro retornarão na variável indicada por NAME.
-

9.41.21 TOUCH

#122

Capturar o retângulo pressionado na **TOUCH SCREEN** e executar a ação programada na figura.

<GET TYPE=TOUCH NAME=n>

- **NAME:** indica o nome da variável que receberá a resposta

Caso o valor definido no retângulo for um FILE, o NAVS executa um comando LOCAL com o arquivo indicado no parâmetro. Caso o valor definido no comando RECTANGLE for um NAME, esse valor será retornado na variável indicada pelo parâmetro NAME acima.

9.41.22 LAUNCH

#123

Capturar a tecla pressionada e executar a ação programada pelo comando LAUNCH_ITEM correspondente

<GET TYPE=LAUNCH>

9.41.23 HOUR

#124

Capturar uma entrada no formato de hora (hh:mm:ss). A formatação é feita automaticamente pelo comando.

<GET TYPE=HOUR NAME=v LIN=1 COL=c [NOENTER|NOENTER=ne] [FILLER=f1] [LABEL=lb] [VALUE=v1]>

Parâmetro	Significado	Valores possíveis
NAME	Nome da variável que receberá o campo capturado	
LIN	Linha da console do primeiro byte do campo ou do valor de LABEL	
COL	Coluna da console do primeiro byte do campo ou do valor de LABEL	
NOENTER	Ao completar a quantidade de 6 dígitos, o comando retorna ao NAVS com o valor digitado	'1' ou 'S'
FILLER	Indica o caracter de preenchimento do campo diferente de brancos	

Especificação de Interface

Parâmetro	Significado	Valores possíveis
LABEL	Rótulo a ser inserido à esquerda do campo a ser digitado. Será posicionado em LIN e COL informado	
VALUE	Valor a ser preenchido e mostrado previamente no campo a ser digitado	

9.41.24 DATETIME

#125

Recupera a data e hora do terminal no formato AAAAMMDDHHMMSS

<GET TYPE=DATETIME NAME=v>

- NAME:** indica o nome da variável que receberá a resposta

9.41.25 CPF

#126

Capturar uma entrada no formato de CPF (xxx.xxx.xxx-xx). A formatação é feita automaticamente pelo comando. Caso o conteúdo seja inconsistente (erro nos dígitos verificadores), o comando recomeça a captura.

<GET TYPE=CPF NAME=v LIN=1 COL=c [NOENTER|NOENTER=ne] [FILLER=f1] [LABEL=1b] [VALUE=v1]>

Parâmetro	Significado	Valores possíveis
NAME	Nome da variável que receberá o campo capturado	
LIN	Linha da console do primeiro byte do campo ou do valor de LABEL	
COL	Coluna da console do primeiro byte do campo ou do valor de LABEL	
NOENTER	Ao completar a quantidade de dígitos, o comando retorna ao NAVS com o valor digitado	'1' ou 'S'
FILLER	Indica o caracter de preenchimento do campo diferente de brancos	
LABEL	Rótulo a ser inserido à esquerda do campo a ser digitado. Será posicionado em LIN e COL informado	
VALUE	Valor a ser preenchido e mostrado previamente no campo a ser digitado	

9.41.26 CNPJ

#127

Especificação de Interface

Capturar uma entrada no formato de CNPJ (xx.xxx.xxx/xxxx-xx). A formatação é feita automaticamente pelo comando. Caso o conteúdo seja inconsistente (erro nos dígitos verificadores), o comando recomeça a captura.

```
<GET TYPE=CNPJ NAME=v LIN=1 COL=c [NOENTER|NOENTER=ne] [FILLER=f1] [LABEL=lb] [VALUE=v1]>
```

Parâmetro	Significado	Valores possíveis
NAME	Nome da variável que receberá o campo capturado	
LIN	Linha da console do primeiro byte do campo ou do valor de LABEL	
COL	Coluna da console do primeiro byte do campo ou do valor de LABEL	
NOENTER	Ao completar a quantidade de dígitos, o comando retorna ao NAVS com o valor digitado	'1' ou 'S'
FILLER	Indica o caracter de preenchimento do campo diferente de brancos	
LABEL	Rótulo a ser inserido à esquerda do campo a ser digitado. Será posicionado em LIN e COL informado	
VALUE	Valor a ser preenchido e mostrado previamente no campo a ser digitado	

9.41.27 BATTERY_STATUS

#128

Recupera o status da carga da bateria podendo ser um valor de 0 a 100 representando o percentual da carga ou o valor **CHR** indicando que a mesma está em carga

```
<GET TYPE=BATTERY_STATUS NAME=v>
```

- **NAME:** indica o nome da variável que receberá a resposta

9.41.28 IMEI

#129

Recupera o valor do IMEI (*International Mobile Equipment Identity*) do terminal.

```
<GET TYPE=IMEI NAME=v>
```

- **NAME:** indica o nome da variável que receberá a resposta

9.41.29 ICCID

#130

Recupera o valor do ICCID (*Integrated Circuit Card Identifier*) do chip GPRS presente no terminal.

Especificação de Interface

<GET TYPE=ICCID NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.30 IMSI

#131

Recupera o valor do IMSI (*International mobile subscriber identity*) do chip GPRS presente no terminal.

<GET TYPE=IMSI NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.31 SSLKEYS

#132

Recupera o as chaves DUKPT e MASTERKEY presentes no terminal

<GET TYPE=SSLKES NAME=v LUATABLE=x RETURN=r>

- **NAME:** indica o nome da variável que receberá a resposta
- **LUATABLE:** indica se a saída é no formato de tabela LUA ou não ('1' ou '0'). Por padrão, caso não fornecido, é não.
- **RETURN:** indica o tipo de retorno que o comando dará:
 - **ALL:** todas as informações serão retornadas. É o *default*.
 - **TYPE:** só o tipo das chaves
 - **KSI:** só o *KSI* das chaves
 - **OWNER:** só o *owner* das chaves (5 últimos bytes do *KSI*).

Formatos de saída:

- LUATABLE=0
 - (1)DUKPT_PS=01,(1)DUKPT_T=3,(1)DUKPT_KSI=FFFFFF12345,(2)DUKPT_PS=02,(2)DUKPT_T=3,(2)DUKPT_KSI=FFFFFF12345,(3)DUKPT_PS=03,(3)DUKPT_T=3,(3)DUKPT_KSI=FFFFFF12345,(1)MASTER_PS=01,(1)MASTER_T=3,(2)MASTER_PS=02,(2)MASTER_T=3,(3)MASTER_PS=03,(3)MASTER_T=3
 - LUATABLE=1
 - {DUKPT={[1]={PS="01",T="3",KSI="FFFFFF12345"},[2]={PS="02",T="3",KSI="FFFFFF12345"},[3]={PS="03",T="3",KSI="FFFFFF12345"}},MASTER={[1]={PS="01",T="3"},[2]={PS="02",T="3"},[3]={PS="03",T="3"}}}
-

Especificação de Interface

9.41.32 WIFIMAC

#133

Retorna o endereço mac do Wifi ou um código erro (-1).

<GET TYPE=WIFIMAC NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.33 MD5

#134

Retorna o *hash* MD5 do arquivo indicado no comando.

<GET TYPE=MD5 NAME=v FILE=f>

- **NAME:** indica o nome da variável que receberá a resposta
 - **FILE:** Indica o nome do arquivo presente no terminal a calcular o MD5
-

9.41.34 OSVERSION

#135

Retorna a versão do sistema operacional.

<GET TYPE=OSVERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.35 LAC

#136

Recupera o LAC (Location Area Code) da conexão GPRS.

<GET TYPE=LAC NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

Caso não seja possível obter o valor de LAC com sucesso, o comando retornará os seguintes códigos de erro:

-1 - Erro ao obter informações do módulo Wireless -2 - O tipo de comunicação atual não é GPRS.

9.41.36 CID

#137

Especificação de Interface

Recupera a identificação da célula (Cell ID) da conexão GPRS.

<GET TYPE=**CID** NAME=**v**>

- **NAME:** indica o nome da variável que receberá a resposta

Caso não seja possível obter o valor de CID com sucesso, o comando retornará os seguintes códigos de erro:

-1 - Erro ao obter informações do módulo Wireless -2 - O tipo de comunicação atual não é GPRS.

9.41.37 SIGNAL_LEVEL

#138

Recupera o nível de sinal da conexão GPRS ou WiFi.

<GET TYPE=**SIGNAL_LEVEL** NAME=**v** FORMAT=**f**>

- **NAME:** indica o nome da variável que receberá a resposta
- **FORMAT:** pode ser:
 - **PERC:** nível percentual de 0 a 100% onde 100% indica aproximadamente -59 dBm
 - **ABS:** nível absolute do sinal onde 30 indica aproximadamente -53 dBm
 - **DBM:** nível em dBm

OBS: Para os modelos da PAX, o parâmetro FORMAT não é utilizado. O nível é sempre retornado como percentual.

9.41.38 DATE ou DATA

#139

Capturar uma entrada no formato de DATA (dd/mm/aa ou dd/mm/aaaa) com 6 ou 8 bytes dependendo do parâmetro size. A formatação é feita automaticamente pelo comando. Caso o conteúdo seja inconsistente, o comando recomeça a captura.

O comando GET TYPE=DATA está depreciado.

<GET TYPE=**DATE** NAME=**v** SIZE=**s**>

- **NAME:** indica o nome da variável que receberá a resposta
 - **SIZE:** Indica o tamanho do campo podendo ser:
 - 6 para o formato dd/mm/aa
 - 8 para o formato dd/mm/aaaa que é o assumido caso o parâmetro SIZE não seja fornecido ou informar outro valor que não 6 ou 8.
-

Especificação de Interface

9.41.39 NONINT

#140

Capturar uma entrada numérica com casas decimais. O valor mostrado será preenchido da direita para a esquerda com o número de casas decimais informados e separação de milhar. Exemplo: R\$1.000.000,00.

```
<GET TYPE=NONINT NAME=v SIZE=s LIN=1 COL=c [NOENTER|NOENTER=ne] [FILLER=f1] [LABEL=1b] [VALUE=v1] [DECIMALS=d] [ZL]>
```

Os parâmetros são os mesmos do comando GET TYPE=FIELD com exceção do argumento ALPHA e com o acréscimo do seguinte:

- **DECIMALS:** indica a quantidade de casas decimais a serem consideradas

9.41.40 TIMEOUT

#141

Utilizado em conjunto com o comando **CAPTURE** para captura múltiplas de entradas, permite abortar o comando se nada for capturado no intervalo de tempo informado nos parâmetros.

```
<GET TYPE=TIMEOUT TIME=t ACTION=a VALUE=v>
```

- **TIME:** indica a quantidade de segundos para esperar antes de abortar
- **ACTION:** determina o tipo de ação quando o tempo expirar sem nenhuma captura, sendo:
 - **CANCEL:** termina o comando **CAPTURE** como se a tecla **CANCEL** fosse pressionada
 - **SEND:** terminal o comando **CAPTURE** enviando na variável indicada pelo parâmetro **NAME** do comando **CAPTURE** com o valor indicado abaixo:
 - **VALUE:** valor a ser passado para a **ACTION=SEND** caso o tempo expire sem captura

Exemplos:

```
<CAPTURE NAME=RESPOSTA>
<GET TYPE=TIMEOUT TIME=20 ACTION=CANCEL>
<GET TYPE=CARD TRACKS=7>
<GET TYPE=IDMIFARE>
</CAPTURE>
```

```
<CAPTURE NAME=RESPOSTA>
<GET TYPE=TIMEOUT TIME=20 ACTION=SEND VALUE=CANCELADO>
<GET TYPE=CARD TRACKS=7>
<GET TYPE=IDMIFARE>
</CAPTURE>
```

Especificação de Interface

9.41.41 SWEVERSION

#142

Recupera a versão CliSitef usada no NAVS.

<GET TYPE=SWEVERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.42 SWEPSCVERSION

#142

Recupera a versão PosCore usada no NAVS.

<GET TYPE=SWEPSCVERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.43 BCVERSION

#143

Recupera a versão da BC (Biblioteca Compartilhada) usada no NAVS.

<GET TYPE=BCVERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.44 LIBLOLVERSION

#144

Recupera a versão biblioteca de licenciamento (LOL) usada no NAVS.

<GET TYPE=LIBLOLVERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.45 LIBPAYVERSION

#144

Recupera a versão biblioteca de pagamento usada no NAVS.

<GET TYPE=LIBPAYVERSION NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta

Especificação de Interface

9.41.46 SIM_SLOTS

#145

Recupera a quantidade de *slots* disponíveis para *SIM cards*

<GET TYPE=SIM_SLOTS NAME=v>

- **NAME:** indica o nome da variável que receberá a resposta
-

9.41.47 MIFARE_READ

#146

Le blocos de dados de cartão Mifare. Veja esclarecimentos sobre o uso do Mifare no capítulo **Mifare: gravação e leitura de dados**

<GET TYPE=MIFARE_READ NAME=n [DATABLOCK=b] [SECTOR=s] [BLOCK=b] [BCOUNT=c] [KEY=k] [RC_NAME=v]>

- **NAME:** indica o nome da variável que receberá a resposta
 - **DATABLOCK:** número do bloco de dados a partir de 0 até 40 (default=0)
 - **SECTOR:** número do setor do cartão
 - **BLOCK:** número do bloco de dados dentro do setor
 - **BCOUNT:** número de blocos a serem lidos (default=1)
 - A cada bloco lido, 32 bytes hexadecimais serão concatenados nos dados a serem recuperados
 - **KEY:** chave de autorização ao(s) bloco(s) com 6 bytes hexadecimais (12 dígitos). Como default, é assumido FFFFFFFF
 - **RC_NAME** - nome da variável que receberá o código de retorno do comando, sendo zero para sem erro
-

9.42 SET TYPE

9.42.1 VAR

#201

Altera o valor de uma variável da configuração do NAVS

<SET TYPE=VAR VARNAME=v VALUE=v1>, onde:

- **VARNAME:** nome da variável do Config.
- **VALUE:** valor a ser atribuído à variável.

9.42.2 CANCEL_KEY

#202

Define o valor da tecla de cancelamento de função NAVS.

```
<SET TYPE=CANCEL_KEY ENABLE|DISABLE FILE=f>
```

- **DISABLE:** desabilita a tecla CANCEL
 - **ENABLE:** habilita a tecla CANCEL - é o valor **DEFAULT**
 - **FILE:** indica o arquivo a ser executado via comando LOCAL ao ser pressionada a tecla CANCEL. Só é eficaz junto com a opção ENABLE
-

9.42.3 PRINTER

#204

Altera as configurações para o comando PRINTER.

```
<SET TYPE=PRINTER [DEFAULT] [DOUBLE_WIDTH] [DOUBLE_HEIGHT] [INVERSE] [WIDTH=w] [HEIGHT=h] [FONT=f]>
```

Ver comando **PRINTER** para o uso deste comando bem como de seus parâmetros.

9.42.4 CONSOLE

#205

Altera as configurações para o comando CONSOLE.

```
<SET TYPE=CONSOLE [DEFAULT] [CONTINUE] [NOCLS|NOCLS=x] [COLOR=c] [BGCOLOR=b] [FONT=f] [JUSTIFY=j]>
```

Ver comando **CONSOLE** para o uso deste comando bem como de seus parâmetros.

9.42.5 TIME

#206

Comando para alterar a data e/ou hora do POS:

```
<SET TYPE=TIME HOUR=223400 DATE=16102002 HDSTS=STSSTT>
```

Onde:

Especificação de Interface

- **HOUR:** É a hora que atribuiremos ao POS, se for “0” será mantida a hora atual. Deverá estar no formato hhmmss (hh = hora, mm= minuto, ss = segundos).
- **DATE:** É a data que atribuiremos ao POS, se for “0” será mantida a data atual. Deverá estar no formato ddmmaaaa (dd = dia, mm= mês, aaaa = ano (exemplo:2013)).
- **HDSTS:** é o retorno do status da operação, que pode ser:
 - OK;
 - ERRO_LEITURA, é erro de leitura de dados do POS quando um dos parâmetros (HOUR ou DATE) vai zerado;
 - ERRO_ENTRADA, é erro no formato enviado para o POS em HOUR ou DATE;
 - ERRO_GRAVAÇÃO, quando o POS não conseguiu efetuar a gravação dos novos parâmetros, portanto a operação não foi executada.

OBS.: As datas devem ser a partir do ano 2000, caso contrário é provável que ocorra “ERRO_GRAVAÇÃO”.

10.0 Programação por eventos

O NAVS, nesta versão 2, permite a programação por eventos. Entradas de teclado e de *touch screen* podem ser direcionadas diretamente para arquivos XML que contenham comandos NAVS definidos pelo usuário e carregados no terminal indicados através dos comandos **RECTANGLE** e **LAUNCH_ITEM**.

Exemplo 1:

```
<RECTANGLE x=1 y=1 height=50 width=50 visible file=acao1.xml>
<RECTANGLE x=61 y=1 height=50 width=50 visible file=acao2.xml>
<GET TYPE=TOUCH>
```

Exemplo 2:

```
<LAUNCH_ITEM key=1 file=acao1.xml>
<LAUNCH_ITEM key=2 file=acao2.xml>
<GET TYPE=LAUNCH>
```

Exemplo 3:

```
<RECTANGLE x=1 y=1 height=50 width=50 visible file=acao1.xml>
<RECTANGLE x=61 y=1 height=50 width=50 visible file=acao2.xml>
<LAUNCH_ITEM key=1 file=acao1.xml>
<LAUNCH_ITEM key=2 file=acao2.xml>
<CAPTURE>
  <GET TYPE=LAUNCH>
  <GET TYPE=TOUCH>
</CAPTURE>
```

Arquivo **acao1.xml**

Especificação de Interface

```
<lua>
  dofile 'main.lua'
    acao1()
</lua>
```

Arquivo **acao2.xml**

```
<lua>
  dofile 'main.lua'
    acao2()
</lua>
```

Agregado a isso, o comando abaixo pode ser usado para direcionar a tecla **CANCEL** para um arquivo específico:

```
<SET TYPE=CANCEL_KEY ENABLE FILE=acao_cancel.xml>
```

Arquivo **acao_cancel.xml**

```
<lua>
  dofile 'main.lua'
    acao_cancel()
</lua>
```

11.0 Histórico das versões

Versão	Data	Conteúdo	Autor
2.0.0	01/07/2019	Versão inicial	Rivera
2.1.0	09/09/2019	Inclusão do Curl para acesso HTTP	Rivera
2.2.0	23/12/2019	Comandos de webservices	Rivera
2.1.3.0	15/07/2020	Reestruturação do código de versão Atualização das libs LOL e PAY Reestruturação do processo de captura de dados Reestruturação do processo de acesso web via Curl Comandos presentes e consolidados nesta versão: LOCAL, DATA, LUA, CANCEL_KEY, POST, RESET ou REBOOT, PRINTER, TONE, DELAY, PRNLOGO, CONLOGO, DEVICE, FTP, UNZIP, CONSOLE, INIT, RECTANGLE, LAUNCH_ITEM, CAPTURE, WRITE_AT, RELICENSE, Web Services, DELFILE, CHGCONFNT, FILE, SLEEP, SELECT GET TYPE: CARD, MODEL, SERIALNO, STSCOMM, COMM_STATUS, HIDDEN, VAR, CANCEL_KEY, FIELD, VALUE, IDCTLS ou IDMIFARE, INSERTCARD, KEY, WIFI_LIST, ANYKEY, SPECIALKEY, VERSION, DESNBS, PASS, EMV, TOUCH,	Rivera

Especificação de Interface

		LAUNCH, HOUR, DATETIME, CPF, CNPJ, BATTERY_STATUS, IMEI, ICCID, IMSI, SSLKEYS, WIFIMAC, MD5, OSVERSION, LAC, CID, SIGNAL_LEVEL, DATE ou DATA, NONINT, TIMEOUT SET TYPE: VAR, CANCEL_KEY, PRINTER, CONSOLE, TIME	
2.1.3.1	25/07/2020	Correções e tratamento de dois SIM cards Comando GET TYPE=SIM_SLOTS Suporte ao comando GET TYPE=FIELD ALPHA para terminais PAX D195	Rivera
2.1.4.0	27/08/2020	Versão da liblol 1.0.6 Comandos de impressão de códigos de barras e afins: PRNFEED, BARCODE, GENERATE_QR_CODE, CODE128, BAR_CODE_3_9 e BAR_CODE_2_5 CTLSWRITE e GET TYPE=MIFARE_READ ANIMATE EMV_BC	Rivera
2.1.4.1	21/09/2020	Adicionado o parâmetro MASK no comando GET TYPE=FIELD	Douglas
2.2.0.0	28/08/2020	Clisitef 7.0.116.7.r1	Douglas
2.2.3.0	03/03/2022	Comando CONFIG_NAVS	Rivera
2.2.4.0	17/03/2022	Incluído <i>log</i> de utilização de memória por comandos	Rivera
2.2.5.0	24/03/2022	Incluído o comando NAVS_UPDATE para atualização do NAVS	Rivera
2.2.5.1	25/03/2022	Corrigido <i>memory leak</i> do comando POST	Rivera
2.2.5.2	04/04/2022	Corrigido <i>memory leak</i> do comando CONLOGO	Rivera
2.2.5.3	07/04/2022	Corrigido <i>crash</i> do comando FILE e incluído <i>log</i> do LOL	Rivera
2.2.5.4	11/04/2022	Corrigido problema com expiração de licença	Rivera
2.2.5.5	20/04/2022	Incluído o parâmetro TYPE no comando NAVS_UPDATE para permitir uso com pacote AUP	Rivera
2.2.5.6	28/04/2022	Acerto da decodificação URL nos headers dos comandos <i>web-services</i>	Rivera
2.2.6.0	02/06/2022	Inclusão do comando EMV_ID	Rivera
2.2.7.0	21/06/2022	Inclusão do comando PAGS suportando o ISO do APSK	Rivera
2.2.8.0	27/06/2022	Inclusão dos comandos LUA para tratamento de bits	Rivera
2.3.0.0	30/06/2022	Novas versões de CliSitef (7.0.117.50.r1), PosCore (9.0.2) e BC (1.45_210201)	Rivera
2.3.0.2	30/07/2022	Correções no comando PAGAMENTO / PAY	Rivera
2.3.0.3	13/07/2022	Correções nos dados enviados no licenciamento Tratamento para SSID WiFi com espaços em branco	Rivera
2.3.0.4	20/08/2022	Acerto ssid wifi com brancos	Rivera

Especificação de Interface

2.3.0.5	27/09/2022	Nova versão da lib PAGES 1.8.2 para acerto bit 35 do ISO APSK	Rivera
2.3.0.6	04/10/2022	Impressão das chaves DUKPT e Master Keys na configuração do NAVS	Rivera
2.3.1.0	21/10/2022	Nova versão da lib de pagamento (1.1.0) para corrigir problemas do envio do CNPJ e CNPJ PDV Nova versão da lib de licenciamento (1.2.0) para permitir o envio dos KSI das chaves de criptografia Nova versão da lib PAGES (1.9.0) com as funções de criptografia NEWDES e DESNBS	Rivera
2.3.2.0	25/11/2022	Inclusão dos parâmetros X e Y no comando WRITE_AT Suporte ao novo terminal PAX Q92	Rivera
2.3.2.1	23/06/2023	Correção de tamanho dos headers para o <u>comando</u> WS	Rivera