



REGIÃO AUTÓNOMA DOS AÇORES
PRESIDÊNCIA DO GOVERNO
Gabinete do Subsecretário Regional da Presidência

Exmo. Senhor Chefe do Gabinete
De Sua Excelência o Presidente da Assembleia
Legislativa da Região Autónoma dos Açores
Rua Marcelino Lima
9901- 858 Horta

S/Referência	S/Comunicação	N/Referência	Data
S/17/2022	04/01/2022	SE/2022/26	07/01/2022

ASSUNTO: Requerimento ao Governo dos Açores n.º 266/XII (PPM) – Contrato de prestação de serviços de transporte regular coletivo de passageiros na ilha das Flores

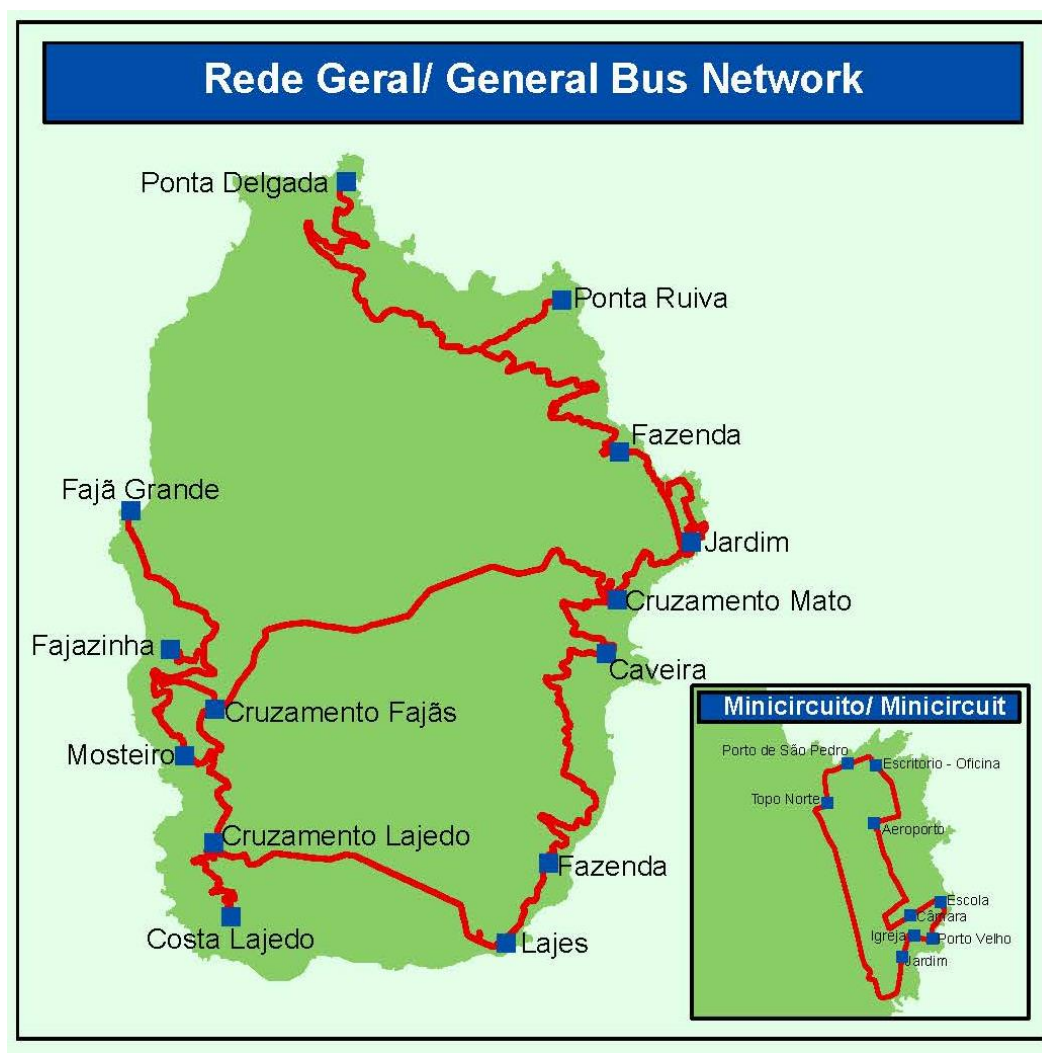
Em resposta ao requerimento mencionado em epígrafe, subscrito pelos Senhores Deputados Gustavo Alves e Paulo Dias, do grupo parlamentar do PPM, sem prescindir quanto ao teor do preâmbulo, encarrega-me o Senhor Subsecretário Regional da Presidência de remeter V. Exa., cópia do caderno de encargos de prestação de serviços associado ao contrato de prestação de serviços de transporte regular coletivo de passageiros na ilha das Flores, conforme o requerido.

Com os melhores cumprimentos,

Em anexo: o mencionado

CADERNO DE ENCARGOS

PRESTAÇÃO DE SERVIÇOS DE TRANSPORTE REGULAR COLETIVO DE PASSAGEIROS NA ILHA DAS FLORES



Fonte: Horários UTC

Índice

CLÁUSULAS JURÍDICAS	3
Cláusula 1. ^a Objeto	3
Cláusula 2. ^a Vigência do contrato	3
Cláusula 3. ^a Preço contratual	3
Cláusula 4. ^a Condições de pagamento	4
Cláusula 5. ^a Propriedade intelectual ou industrial	4
Cláusula 6. ^a Dever de sigilo	5
Cláusula 7. ^o Deveres de informação	6
Cláusula 8. ^a Revisão de preços e reposição do equilíbrio financeiro do contrato	7
Cláusula 9. ^a Sanções contratuais	8
Cláusula 10. ^a Seguros	10
Cláusula 11. ^a Cessão da posição contratual e subcontratação	11
Cláusula 12. ^a Contagem dos prazos	11
Cláusula 13. ^a Comunicações e notificações	11
Cláusula 14. ^a Foro competente	11
Cláusula 15. ^a Legislação aplicável	12
CLÁUSULAS TÉCNICAS	13
Cláusula 16. ^a Disposições gerais	13
Cláusula 17. ^a Horários e itinerários	14
Cláusula 18. ^a Publicitação de horários e carreiras	15
Cláusula 19. ^a Veículos afetos ao serviço	16
Cláusula 20. ^a Tarifas e Títulos de Transporte	16
Cláusula 21. ^a Pessoal afeto à prestação do serviço	17
Cláusula 22. ^a Locais de venda de títulos de transporte	17
Cláusula 23. ^a Sistema de bilhética	17
Cláusula 24. ^a Estatísticas e fiscalização	18
ANEXO I HORÁRIOS/ITINERÁRIOS/PARAGENS	19
ANEXO II TÍTULOS DE TRANSPORTE	43
ANEXO III TARIFÁRIO	46
ANEXO IV TABELA DESCONTOS PASSES MENSAIS	49
ANEXO V TIPOLOGIA DE VEÍCULOS	50
ANEXO VI QUILOMETRAGEM ANUAL	51
ANEXO VII QUADRO DE PESSOAL DA EMPRESA UNIÃO DE TRANSPORTES DOS CARVALHOS, LDA. (UTC – Açores)	52
ANEXO VIII BILHÉTICA	53



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

CLÁUSULAS JURÍDICAS

Cláusula 1.^a

Objeto

O presente caderno de encargos compreende as cláusulas a incluir no contrato a celebrar para a Prestação de Serviços de Transporte Regular Coletivo de Passageiros das Flores.

Cláusula 2.^a

Vigência do contrato

- 1 - O contrato produz efeitos ao 1.º dia útil do mês seguinte da data da comunicação da emissão do visto pelo Tribunal de Contas, o qual deverá ser notificado pela Secretaria Regional dos Transportes e Obras Públicas ao cocontratante no prazo de 5 dias a contar do conhecimento do mesmo.
- 2 - O contrato mantém-se em vigor pelo prazo de 10 anos, sem prejuízo das obrigações acessórias que devem perdurar para além da sua cessação, podendo ser prorrogado por períodos de 1 ano e até ao limite máximo de 5 anos, por opção do contraente público, a comunicar ao cocontratante através de carta registada com pelo menos 60 dias de antecedência relativamente ao termo do prazo inicial de vigência do contrato.

Cláusula 3.^a

Preço contratual

- 1 - O preço contratual é o que resulta da proposta adjudicada a qual não pode ser superior a 2.925.000,00€ (dois milhões novecentos e vinte e cinco mil euros), acrescido de IVA à taxa legal em vigor.
- 2 - O preço anual corresponde à divisão por 15 do preço constante da proposta adjudicada.
- 3 - O preço contratual inclui todos os custos, encargos e despesas cuja responsabilidade não esteja expressamente atribuída ao contraente público, incluindo todas as despesas com o seu pessoal, nomeadamente salários, contribuições obrigatórias para a Segurança Social, seguros de acidentes de trabalho ou outros que se revelem necessários e todas as outras despesas sociais obrigatórias, as despesas de alojamento, alimentação e deslocação de meios humanos, despesas de aquisição, transporte, armazenamento e manutenção de equipamentos e autocarros, reparação e manutenção



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

da frota de autocarros afetos à atividade, incluindo consumíveis (combustíveis e lubrificantes) e demais materiais e mão-de-obra necessários ao bom funcionamento da frota, bem como quaisquer encargos decorrentes da utilização de marcas registadas, patentes ou licenças e os custos de implementação e de funcionamento do sistema de bilhética desmaterializada.

Cláusula 4.^a
Condições de pagamento

- 1 - Não há lugar a pagamentos antecipados.
- 2 - Sem prejuízo do disposto no número seguinte, o pagamento mensal corresponde a 1/12 do preço anual.
- 3 - Se o início ou a conclusão dos serviços não coincidirem com o primeiro ou o último dia do mês correspondente, o valor mensal será dividido pelo número de dias do mês em apreço, de modo a se encontrar o valor diário e correspondente montante a pagar nesse mês.
- 4 - Os pagamentos são efetuados no prazo de 60 dias após a receção de cada fatura relativa aos serviços prestados no mês anterior a qual deve ser remetida ao Fundo Regional dos Transportes Terrestres, IPRA.
- 5 - Em caso de discordância por parte do contraente público quanto aos valores indicados na fatura, deve esta comunicar ao cocontratante, por escrito, os respetivos fundamentos, ficando o cocontratante obrigado a prestar os esclarecimentos necessários ou proceder à emissão de nova fatura corrigida.
- 6 - Desde que devidamente emitidas as faturas são pagas através de transferência bancária, para tal devendo o cocontratante indicar ao contraente público os dados necessários para o efeito.

Cláusula 5.^a
Propriedade intelectual ou industrial

- 1 - São da responsabilidade do cocontratante os encargos decorrentes da utilização, nos serviços em causa, de marcas registadas, patentes registadas ou licenças, bem como a obtenção das autorizações necessárias ao exercício da atividade objeto do presente contrato.
- 2 - O cocontratante é responsável pela violação de quaisquer direitos de patente, de conceção, de licenças, de projetos, de marcas, de nomes ou de quaisquer outros direitos de propriedade intelectual, industrial ou afins, respeitantes aos serviços objeto do contrato e aos bens envolvidos



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

na sua execução, nomeadamente projetos, estudos, programas informáticos, equipamento, materiais, documentação ou trabalhos realizados.

- 3 - O cocontratante é responsável por qualquer reclamação formulada perante o contraente público, resultante de violação dos direitos referidos nos números anteriores, adotando o contraente público o procedimento que se releve mais adequado para a intervenção plena do cocontratante na discussão e no esclarecimento, perante terceiros reclamantes ou quaisquer autoridades, das dúvidas que, neste âmbito, se coloquem.
- 4 - O material utilizado ou produzido no âmbito do presente contrato e entregue ao contraente público torna-se propriedade originária deste, ficando, em consequência, a única titular de todos os direitos de autor e conexos e de propriedade industrial inerentes aos mesmos e, neste sentido, fica-lhe reservada a faculdade de proceder à sua utilização, total ou parcial, tal como lhe foi entregue ou com as modificações que entenda convenientes fazer, após a conclusão ou em caso de resolução do contrato, não podendo o cocontratante fazer uso do mesmo fora do âmbito daquele sem o consentimento expresso do contraente público.
- 5 - O cocontratante responde, independentemente de culpa, pelos danos que sejam imputados ao contraente público e que se produzam perante terceiros, quando decorrentes de violação dos direitos a que alude o número anterior, devendo indemnizar o contraente público de todas as despesas que, em consequência, haja de fazer e de todas as quantias que tenha de pagar seja a que título for.

Cláusula 6.^a
Dever de sigilo

- 1 - O prestador deve guardar sigilo sobre toda a informação e documentação, técnica e não técnica, comercial ou outra de que possa ter conhecimento ao abrigo ou em relação com a execução do contrato.
- 2 - A informação e a documentação cobertas pelo dever de sigilo não podem ser transmitidas a terceiros, nem objeto de qualquer uso ou modo de aproveitamento que não o destinado direta e exclusivamente à execução do contrato.
- 3 - Exclui-se do dever de sigilo previsto a informação e a documentação que fossem comprovadamente do domínio público à data da respetiva obtenção pelo cocontratante ou que



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

este seja legalmente obrigado a revelar, por força da lei, de processo judicial ou a pedido de autoridades reguladoras ou outras entidades administrativas competentes.

- 4 - O dever de sigilo mantém-se em vigor até ao termo do prazo de 5 anos a contar do cumprimento ou cessação, por qualquer causa, do contrato, sem prejuízo da sujeição subsequente a quaisquer deveres legais relativos, designadamente à proteção de segredos comerciais ou da credibilidade, do prestígio ou da confiança devidos às pessoas coletivas.

Cláusula 7.º
Deveres de informação

- 1 - O contratante compromete-se a fornecer as informações necessárias ao cabal acompanhamento e fiscalização da execução do contrato, obrigando-se com o nível de detalhe e a periodicidade que lhe forem determinadas pelo contraente público, nomeadamente, a:
- a) Dar-lhe conhecimento imediato de todo e qualquer evento que possa vir a prejudicar ou impedir o cumprimento pontual e atempado de qualquer das obrigações emergentes do contrato;
 - b) Remeter-lhe, até ao dia 31 de maio de cada ano civil, a conta de exploração relativa à atividade objeto do contrato (com detalhe e desagregação de receitas e custos operacionais), bem como todas as demais informações relativas à respetiva condição financeira relativa ao ano civil anterior;
 - c) Remeter-lhe, nos 60 dias subsequentes ao termo de cada semestre, um relatório com informação detalhada e desagregada com as quantidades e faturação mensal para cada um dos diferentes títulos de transporte definidos no anexo II;
 - d) Fornecer-lhe toda a informação financeira e operacional que lhe for solicitada, nomeadamente indicadores de desempenho, no prazo de 15 dias e pelos meios informáticos adequados, salvo nos casos em que exista justificação em contrário;
 - e) Autorizar e colaborar na realização, pelo contraente público ou entidade por este designada, de auditorias financeiras e operacionais, sempre que tal lhe seja solicitado;
- 2 - A obrigação prevista no número anterior compreende o dever de o cocontratante participar em reuniões, com o contraente público ou com outras entidades, que se mostrem objetivamente necessárias em função do objeto do contrato.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

- 3 - O cocontratante deverá enviar ao contraente público informação sobre todas as carreiras e respetivos horários em formato de folha de cálculo até 30 dias após o início de vigência do contrato e até 15 dias após a ocorrência de qualquer alteração à informação prestada.
- 4 - O cocontratante obriga-se a comunicar ao contraente público e até 10 dias após ter conhecimento do início ou a iminência de qualquer processo judicial ou extrajudicial que possa conduzir à sua declaração de insolvência, a providência análoga à insolvência ou à sua extinção, bem como da verificação de qualquer outra circunstância que perturbe a execução do contrato.

Cláusula 8.ª

Revisão de preços e reposição do equilíbrio financeiro do contrato

- 1 - O preço anual é revisto em cada aniversário de vigência do contrato, tendo por base o índice de preço no consumidor (IPC), total com exclusão de habitação, registado na Região Autónoma dos Açores, exceto quando o mesmo seja negativo.
- 2 - O cocontratante tem direito à reposição do equilíbrio financeiro do contrato nos termos dos artigos 282.º e 314.º do Código dos Contratos Públicos (CCP), assim como nos seguintes casos e noutros que o contrato preveja:
 - a) Nas circunstâncias previstas no n.º 6 da cláusula 17.ª:

O cocontratante apresenta ao contraente público uma proposta de preço por km para realização dos itinerários ou carreiras em causa, devidamente justificada, nomeadamente em função das características dos itinerários a realizar, dos autocarros a utilizar e da estimativa de receita decorrente da venda de títulos de transporte. O preço por km a propor pelo cocontratante não pode ultrapassar em 10% o preço do km calculado a partir da relação entre o preço anual para o circuito e o número de km correspondente à soma da distância dos itinerários de todas as carreiras identificadas neste circuito.
 - b) Na implementação da Fase 2 do sistema de bilhética, conforme previsto na cláusula 23.ª:

O cocontratante deve demonstrar ao contraente público que os custos de implementação da fase 2 do sistema de bilhética são iguais ou superiores a 1,5% do preço contratual anual.
- 3 - Caso as circunstâncias previstas na alínea a) do n.º 2 implicarem a diminuição do número de circuitos das distâncias percorridas em cada itinerário, o preço anual devido é igualmente revisto em função do mesmo critério.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

- 4 - Sem prejuízo do referido no número anterior, a redução do preço contratual não pode ser superior a 20%.

Cláusula 9.^a
Sanções contratuais

- 1 - Pelo incumprimento de obrigações emergentes do contrato, o contraente público pode exigir ao cocontratante o pagamento de uma sanção pecuniária, num montante a fixar em função da gravidade do incumprimento, nos seguintes termos:

- a) Quando a chegada ocorra com mais de 20 minutos de atraso relativamente aos horários constantes do anexo I ou àqueles que no momento estiverem em vigor, uma penalidade calculada de acordo com a fórmula:

$$P = [0,8 * (V/Q)] * D$$

Em que:

P é igual ao valor da penalização

V é igual ao valor anual do contrato

Q é igual número de km anual previsto dos itinerários em vigor

D é o número de km do percurso mais longo em que se deu a mora.

- b) Quando o cocontratante não preste o serviço em qualquer das carreiras contratualmente previstas, nos termos dos itinerários constantes do anexo I ou àqueles que no momento estiverem em vigor, uma penalidade calculada de acordo com a seguinte fórmula:

$$P = [1,2 * (V/Q)] * D$$

Em que:

P é igual ao valor da penalização

V é igual ao valor anual do contrato

Q é igual número de km anual previsto dos itinerários em vigor

D é o número de km do (s) itinerário (s) não realizado (s).

- c) Quando o cocontratante não preste o serviço em dia em que estivesse obrigado a prestar, uma penalidade calculada de acordo com a seguinte fórmula:

$$P = (V * A / 365) \times 1,5$$



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Em que:

P é igual ao valor da penalização

V é igual ao valor anual do contrato

A é igual ao número de dias em que o contrato não foi cumprido.

- d) Quando o cocontratante não cumpra com requisito da idade média da frota exigida, uma penalidade calculada de acordo com a seguinte fórmula:

$$P = (IM1 - IM0) \times V \times 0,1$$

Em que:

P é igual ao valor da penalização

IM1 é igual à idade média da frota real = n.º anos + (n.º meses/12)

IM0 é igual à idade média da frota definida contratualmente (20 anos)

V é igual ao valor anual do contrato.

- e) Quando o cocontratante não cumpra com prazo de implementação do sistema de bilhética contratualmente exigido ou quando não proceda à disponibilização em sítio da internet da informação no prazo definido, uma penalidade calculada de acordo com a seguinte fórmula:

$$P = (B1 - B0) \times V \times 0,01$$

Em que:

P é igual ao valor da penalização

B1 é igual ao número de meses de incumprimento

B0 é igual ao número de meses definidos contratualmente

V é igual ao valor anual do contrato.

- f) Quando o cocontratante não cumpra com os deveres de informação, uma penalidade calculada de acordo com a seguinte fórmula:

$$P = [0,2 \times (V/Q)] \times D$$

Em que:

P é igual ao valor da penalização

V é igual ao valor anual do contrato



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Q é igual ao número total de dias anuais da prestação do serviço (250 dias)

D é o número de dias de incumprimento.

- 2 - O valor da penalidade prevista na alínea b) do número anterior é multiplicado pelo fator 1,5 no caso de a carreira em causa ser realizar entre as 7:30 horas e as 9:00 horas de 2ª feira a sábado.
- 3 - O fator 0,1 constante da formula referida na aliena d) do número anterior, passa a 0,15 quando o incumprimento seja igual ou superior a dois anos.
- 4 - As penalidades a que se referem os números anteriores são deduzidas do pagamento relativo à primeira fatura que vier a ser emitida após a ocorrência do facto que lhe deu origem.
- 5 - No caso de não existirem mais pagamentos a realizar, o cocontratante paga a penalidade aplicada no prazo máximo de 20 dias contados da data em que lhe foi notificada a respetiva aplicação.
- 6 - As penalidades previstas nos números anteriores são aplicadas até ao montante máximo de 20% do preço contratual.
- 7 - As penalidades previstas nos números anteriores têm a natureza de cláusula penal indemnizatória, mas não prejudicam a possibilidade de o contraente público ser ressarcido pelo dano excedente, no caso de existir.
- 8 - As penalidades são aplicadas por comunicação, por carta registada com aviso de receção, dirigida ao cocontratante.
- 9 - Quando as penalidades aplicadas excederem o limite previsto no n.º 6 pode o contraente público considerar o contrato como definitivamente incumprido, designadamente para efeitos de resolução.
- 10 - No caso de cumprimento defeituoso por causa imputável ao cocontratante, e sem prejuízo da aplicação das penalidades, o contraente público reserva-se ainda ao direito de requisitar a terceiros a prestação dos serviços objeto do contrato, correndo os respetivos custos por conta do cocontratante.

Cláusula 10.ª
Seguros

- 1 - É da responsabilidade do cocontratante a cobertura, através de contratos de seguro, de todos os riscos próprios da atividade de transporte público coletivo de passageiros, incluindo:
 - a) Seguro de responsabilidade civil por danos corporais e materiais;



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

- b) Seguros de acidentes de trabalho ou outros que se revelem necessários relativamente à mão-de-obra empregue na prestação de serviços.
 - c) Seguro automóvel efetuado nos termos da legislação em vigor.
- 2 - O contraente público pode, sempre que entender conveniente, exigir prova documental da celebração dos contratos de seguro referidos no número anterior, devendo o cocontratante fornecê-la no prazo 5 dias.

Cláusula 11.^a
Cessão da posição contratual e subcontratação

O cocontratante não pode ceder a sua posição no contrato ou subcontratar total ou parcialmente os serviços incluídos no mesmo sem autorização prévia do contraente público.

Cláusula 12.^a
Contagem dos prazos

Os prazos previstos no contrato são contínuos, correndo em sábados, domingos e dias feriados.

Cláusula 13.^a
Comunicações e notificações

- 1 - Sem prejuízo de acordo em contrário, as notificações e comunicações devem ser dirigidas, nos termos do Código dos Contratos Públicos, para o domicílio ou sede contratual de cada uma das partes, identificados no contrato.
- 2 - Qualquer alteração das informações de contacto constantes do contrato deve ser comunicada à outra parte.

Cláusula 14.^a
Foro competente

- 1 - Para resolução de quaisquer litígios emergentes do contrato, designadamente relativos à interpretação, execução, incumprimento, invalidade ou resolução do mesmo, é competente o tribunal administrativo de Ponta Delgada, com expressa renúncia a qualquer outro.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

2 - As partes no contrato podem derrogar o disposto no número anterior por acordo escrito, decidindo submeter à arbitragem algum litígio específico.

Cláusula 15.^a
Legislação aplicável

O contrato é regulado pelo disposto no Decreto Legislativo Regional n.º 27/2015/A, de 29 de dezembro, no Código dos Contratos Públicos, o Regulamento (CE) n.º 1370/2007, do Parlamento Europeu e do Conselho, de 23 de outubro de 2007, o artigo 59.º do Decreto Legislativo Regional n.º 1/2009/A, de 7 de janeiro, que adapta à Região Autónoma dos Açores a Lei n.º 52/2015, de 9 de junho, que aprova o Regime Jurídico do Serviço Público de Transporte de Passageiros e demais legislação aplicável.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

CLÁUSULAS TÉCNICAS

Cláusula 16.^a
Disposições gerais

- 1 - Sem prejuízo de outras obrigações previstas na legislação aplicável, o cocontratante fica obrigado, a título principal, a efetuar os serviços de transporte regular coletivo de passageiros na Ilha das Flores, nos termos e nas condições previstas no presente caderno de encargos e respetivos anexos.
- 2 - O serviço de transporte regular coletivo de passageiros tem por objetivo servir as populações dos locais e percursos identificados com um serviço regular e contínuo de transportes públicos de passageiros.
- 3 - O serviço de transporte regular coletivo de passageiros só pode ser prestado por empresas de transportes públicos coletivos rodoviários de passageiros exercida por sociedades comerciais ou cooperativas, licenciadas mediante alvará comprovativo para o exercício da atividade, obtido nos termos do artigo 3.º do Decreto-Lei n.º 3/2001, de 10 de janeiro.
- 4 - Para efeitos do presente contrato, adotam-se os seguintes conceitos:
 - a) “Transporte público ou por conta de outrem”: o transporte de passageiros, oferecido ao público ou a certas categorias de utentes, que não se classifique como particular, nos termos previstos no Decreto-Lei n.º 3/2001, de 10 de janeiro;
 - b) “Serviços regulares”: aqueles que asseguram o transporte de passageiros segundo itinerário, frequência, horário e tarifas predeterminados e em que podem ser tomados e largados passageiros em paragens previamente estabelecidas, nos termos definidos no Decreto-Lei n.º 3/2001, de 10 de janeiro;
 - c) “Autocarro”: o veículo automóvel construído ou adaptado para o transporte de passageiros com lotação superior a nove lugares, incluindo o condutor, nos termos definidos no Decreto-Lei n.º 3/2001, de 10 de janeiro;
 - d) “Intermunicipal”: serviço público de transporte de passageiros que visa satisfazer as necessidades de deslocação entre diferentes municípios ou concelhos de uma ilha, nos termos definidos no artigo 59.º do Decreto Legislativo Regional n.º 1/2019/A, de 7 de janeiro;
 - e) “Municipal suburbano”: o serviço público de transporte de passageiros que visa satisfazer as necessidades de deslocação fora da área urbana de um município, entendendo-se como tal o que se desenvolve integralmente ou maioritariamente fora da respetiva área urbana da sede de



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

concelho, nos termos definidos no artigo 59.º do Decreto Legislativo Regional n.º 1/2019/A, de 7 de janeiro;

- f) “Municipal urbano”: o serviço público de transporte de passageiros que visa satisfazer as necessidades de deslocação dentro da área urbana de um município, entendendo-se como tal o que se desenvolve integralmente ou maioritariamente dentro da respetiva área urbana da sede de concelho, nos termos definidos no artigo 59.º do Decreto Legislativo Regional n.º 1/2019/A, de 7 de janeiro.

Cláusula 17.ª
Horários e itinerários

- 1 - Os itinerários e a cadência horária (horários) referentes a cada carreira (circuito) são os apresentados no anexo I.
- 2 - A quilometragem anual prevista para realização dos horários e itinerários a que se refere o número anterior são apresentados no anexo VI.
- 3 - As carreiras e horários identificados só podem ser alterados pelo contraente público, mediante comunicação escrita ao cocontratante, com a antecedência mínima de 8 dias úteis.
- 4 - As carreiras e horários poderão ser pontualmente alterados sempre que por motivos não imputáveis ao cocontratante, tal se justifique.
- 5 - Considera-se motivos não imputáveis ao cocontratante a impossibilidade de circulação numa via (imposta por sinalização ou motivos de obras) ou por motivos de força maior.
- 6 - Sempre que a alteração dos percursos implicar uma alteração contratual em mais de 10% da quilometragem o contrato poderá ser revisto nos termos da cláusula 8.ª.
- 7 - O cocontratante pode efetuar, pontualmente, ou a título permanente, novos horários e carreiras além dos contratualmente estabelecidos, decorrendo neste caso por conta do mesmo qualquer custo e responsabilidade pelo serviço.
- 8 - O cocontratante poderá ajustar os itinerários e horários das carreiras regulares a realizar em período escolar, em função das necessidades do sistema escolar, devendo para tal, no início de cada ano letivo, reunir com os conselhos executivos das respetivas escolas.
- 9 - O cocontratante obriga-se a reforçar o número de lugares/autocarros de um circuito, sempre que a procura pelos utentes ultrapasse a capacidade em termos de lugares oferecidos, com especial incidência no período escolar.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

- 10 - Para efeitos do disposto no número anterior, o cocontratante pode realizar carreiras em regime de desdobramento.
- 11 - A prestação de serviços ocorre de 2.^a feira a 6.^a feira, com exceção dos feriados nacionais e regionais.
- 12 - Considera-se horário diurno o período compreendido entre as 7:00 horas e as 20:00 horas.
- 13 - Sem prejuízo do disposto nos números 1, 11 e 12, a linha 6 pode contemplar serviços durante o fim de semana e em horário noturno, sendo anualmente revisto o horário em função da operação de Verão da Atlânticoline.

Cláusula 18.^a

Publicitação de horários e carreiras

- 1 - O cocontratante obriga-se a fixar nas paragens os horários e itinerários das carreiras objeto da prestação de serviços, os quais devem ser intuitivos e perceptíveis à generalidade dos utentes, devendo ainda conter a indicação da rede, redigidos em Português e Inglês.
- 2 - Os horários e itinerários das carreiras a que alude o número anterior deverão possuir as dimensões adequadas, fixadas em local visível e devidamente protegidos contra intempéries e atos de vandalismo.
- 3 - O cocontratante, no prazo máximo de 30 dias após entrada em vigor do contrato, entregará ao contraente público para efeitos de aprovação, o projeto de horários e itinerários e a indicação dos materiais a utilizar, o qual deverá ser aprovado por esta no prazo máximo de 30 dias.
- 4 - Os horários e itinerários aprovados pelo contraente público têm de estar fixados pelo cocontratante no prazo máximo de 30 dias após a referida aprovação.
- 5 - O cocontratante obriga-se ainda, no prazo máximo de 6 meses após a entrada em vigor do contrato, à divulgação dos horários, itinerários e tarifários em sítio da Internet a indicar/criar para o efeito.
- 6 - No caso previsto no n.º 3 da cláusula 17.^a, o cocontratante dispõe de 8 dias, após a comunicação do contraente público, para fixar os novos horários e itinerários.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Cláusula 19.^a
Veículos afetos ao serviço

- 1 - Os veículos a utilizar na prestação de serviços devem estar devidamente licenciados e homologados nos termos da legislação em vigor (Decreto-Lei n.º 58/2004, de 19 de março) para a atividade de transporte coletivo de passageiros.
- 2 - Os autocarros a afetar à prestação de serviços deverão ser os adequados aos circuitos e população a servir e deverão obedecer às tipologias de veículos conforme anexo V.
- 3 - O contraente público pode solicitar a substituição de autocarro, caso verifique a desadequação, em termos de características e lotação do mesmo, ao circuito.
- 4 - É da responsabilidade do cocontratante a identificação dos autocarros utilizados na execução do contrato, mediante inscrição de logótipo e/ou denominação social da empresa.
- 5 - É da responsabilidade do cocontratante a identificação, em zona frontal do autocarro, da carreira a que o veículo se encontra afeto.
- 6 - O cocontratante obriga-se a manter os autocarros utilizados no serviço em boas condições de limpeza e higiene, bem como a cumprir todos os requisitos de segurança legalmente estabelecidos.
- 7 - A idade média da frota de autocarros a afetar à exploração não poderá ultrapassar os 20 anos, contados da data de atribuição da primeira matrícula.

Cláusula 20.^a
Tarifas e Títulos de Transporte

- 1 - O cocontratante fica obrigado a disponibilizar os títulos de transporte e a praticar os descontos comerciais indicados no anexo II.
- 2 - As tarifas de bordo aprovadas para cada carreira, em vigor para o ano de 2019, são as constantes do anexo III.
- 3 - Considera-se percurso efetuado a distância percorrida pelo utente entre o ponto de partida e o local de chegada (local de destino).



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Cláusula 21.^a

Pessoal afeto à prestação do serviço

- 1 - O cocontratante fica obrigado a admitir no quadro da empresa os funcionários da empresa União dos Transportes dos Carvalhos, Lda., atual prestadora do serviço público, constantes do anexo VII.
- 2 - O cocontratante fica ainda obrigado a cumprir os regimes salariais dos funcionários a que alude o número anterior, nomeadamente em termos de vencimento base e diuturnidades, de acordo com a Convenção Coletiva de Trabalho do sector.
- 3 - O cocontratante complementa o seu quadro de pessoal com a contratualização de outros funcionários que se revelem necessários à boa prossecução do objeto da presente prestação de serviços.
- 4 - Os condutores afetos à prestação de serviço deverão possuir as necessárias habilitações profissionais, comprovadas pela titularidade de título de condução válido para a categoria do veículo a que conduz.

Cláusula 22.^a

Locais de venda de títulos de transporte

- 1 - O cocontratante fica obrigado a dispor, pelo menos, de um posto de atendimento, de informação ao utente e de venda de passes e demais bilhetes pré-adquiridos.
- 2 - O posto de atendimento a que alude o número anterior, quando integrado em edifício com outras valências da empresa adjudicatária, deverá possuir espaço próprio, delimitado e reservado ao atendimento ao público.

Cláusula 23.^a

Sistema de bilhética

- 1 - Sem prejuízo de outras obrigações previstas na legislação aplicável, o cocontratante fica obrigado a implementar o sistema de bilhética desmaterializada cuja as especificações técnicas e arquitetura são as constantes do anexo VIII.
- 2 - O sistema de bilhética será implementado por fases, correspondendo a Fase 0 ao início do contrato, a Fase 1 aos pressupostos desse sistema no 4.º ano de vigência do contrato e a Fase 2



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

deverá ser implementada até 6 meses após indicação do contraente público, conforme previsto no anexo VIII.

- 3 - A título acessório, o cocontratante fica ainda obrigado, designadamente, a recorrer a todos os meios humanos, materiais e informáticos que sejam necessários e adequados à prestação do serviço, bem como ao estabelecimento do sistema de organização necessário à perfeita e completa execução do sistema a implementar.

Cláusula 24.^a
Estatísticas e fiscalização

- 1 - O contraente público procede ao acompanhamento da execução do contrato, podendo designadamente:
 - a) Solicitar esclarecimentos quanto à execução do contrato;
 - b) Efetuar vistorias ou inspeções para verificar o estado dos autocarros ou as condições em que o serviço de transporte público coletivo é executado;
 - c) Propor medidas de melhoramento da execução do contrato.
- 2 - O cocontratante faculta, gratuitamente, a entrada dos representantes do contraente público em ação de fiscalização nas carreiras objeto do contrato, bem como a informação necessária aos trabalhos de fiscalização.
- 3 - O cocontratante obriga-se ainda a fornecer a informação do sistema de bilhética no modelo de dados referido no anexo VIII.
- 4 - Até à entrada em funcionamento do sistema de bilhética a que se refere o número anterior, o cocontratante enviará mensalmente o resumo dos passageiros transportados por mês nas diferentes carreiras.



REGIÃO AUTÓNOMA DOS AÇORES

SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

ANEXO I

HORÁRIOS/ITINERÁRIOS/PARAGENS

Linhas/Horários

Santa Cruz – Ponta Ruiva – Ponta Delgada

LINHA/LINE 1: SANTA CRUZ - PONTA DELGADA

Sentido Ida/Going Way					
	A 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª	B 2ª 4ª 6ª	E 2ª 3ª 4ª 5ª 6ª	A 2ª 3ª 4ª 5ª 6ª
Jardim Santa Cruz	07:05	-	13:00	-	16:50
Topo Norte	07:07	-	13:02	-	16:52
Porto de São Pedro	07:08	-	13:03	-	16:53
Escritorio - Oficina	07:09	-	13:04	-	16:54
Aeroporto	07:10	-	13:05	-	16:55
Camara	07:11	-	13:06	-	16:56
Escola	07:12	-	13:07	-	16:57
Porto Velho	07:13	-	13:08	-	16:58
Igreja	07:14	-	13:09	-	16:59
Jardim Santa Cruz	07:15	08:30	13:10	15:00	17:00
Monte	07:18	08:34	13:13	15:04	17:04
Fazenda	07:20	08:37	13:15	15:07	17:07
Cedros 1	07:27	-	13:22	-	17:16
Cedros 2	07:28	-	13:23	-	17:18
Ponta Ruiva Entrada	07:32	-	13:27	-	17:24
Ponta Ruiva	07:36	-	13:31	-	17:30
Ponta Ruiva Entrada	07:40	-	13:35	-	17:34
Ponta Delgada Entrada	07:56	-	13:51	-	17:50
Ponta Delgada 2	07:58	-	13:53	-	17:52
Ponta Delgada 3	07:59	-	13:54	-	17:53
Ponta Delgada Fim	08:00	-	13:55	-	17:55

A = Efetuado anualmente/Performed annually

B = Efetuado Anualmente às 2ª, 4ª e 6ª feira/ Performed annually at Monday, Wednesday and Friday

E = Efetuado em período escolar/Performed in school season.

- = Sem passagem nesta paragem/No service at this stop

 Minicircuito/Minicircuit S. Cruz



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Sentido Volta/ Return Way					
	A 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª	B 2ª 4ª 6ª	E 2ª 3ª 4ª 5ª 6ª	A 2ª 3ª 4ª 5ª 6ª
Ponta Delgada Fim	08:00	-	14:00	-	17:45
Ponta Delgada 3	08:02	-	14:01	-	17:46
Ponta Delgada 2	08:04	-	14:02	-	17:47
Ponta Delgada Entrada	08:08	-	14:03	-	17:48
Ponta Ruiva Entrada	08:10	-	14:20	-	17:49
Ponta Ruiva	08:16	-	-	-	18:05
Ponta Ruiva Entrada	08:22	-	-	-	18:10
Cedros 2	08:26	-	14:25	-	18:16
Cedros 1	08:28	-	14:26	-	18:17
Fazenda	08:47	08:40	14:34	15:10	18:25
Monte	08:49	08:43	14:36	15:14	18:26
Topo Norte	08:50	08:44	-	15:15	18:30
Porto de São Pedro	08:51	08:45	-	15:16	18:31
Escritório - Oficina	08:52	08:46	-	15:17	18:32
Aeroporto	08:54	08:47	-	15:18	18:34
Camara	08:56	08:48	-	15:19	18:36
Escola	08:57	08:49	-	15:20	18:37
Porto Velho	08:58	08:50	-	15:21	18:38
Igreja	08:59	08:51	-	15:22	18:39
Jardim Santa Cruz	09:00	08:52	14:40	15:23	18:40

A = Efetuado anualmente/Performed annually

B = Efetuado Anualmente às 2ª, 4ª e 6ª feira/ Performed annually at Monday, Wednesday and Friday

E = Efetuado em período escolar/Performed in school season.

- = Sem passagem nesta paragem/No service at this stop

Minicircuito/Minicircuit S. Cruz



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Itinerário e Paragens

Linha 1

Santa Cruz - Ponta Ruiva - Ponta Delgada

P	Itinerário	GPS	Km	Acumul.
1	Jardim S. Cruz	39.27.08 31.07.70	0,00	
2	Câmara	27.20 07.68	3,50	3,50
3	Escola	27.15 07.57	0,60	4,10
4	Jardim		0,30	4,40
5	Monte	27.71 08.20	2,00	6,40
6	Fazenda	28.09 08.72	1,30	7,70
7	Cedros 1	28.86 09.12	4,40	12,10
8	Cedros 2	28.88 09.46	0,60	12,70
9	P. Ruiva Entrada	29.03 09.98	2,70	15,40
10	P. Ruiva	29.78 09.53	2,20	17,60
11	P. Ruiva Entrada		2,20	19,80
12	P. Delgada Entrada	30.67 12.76	9,00	28,80
13	P. Delgada 2	30.81 12.60	0,60	29,40
14	P. Delgada 3	30.95 12.57	0,30	29,70
15	P. Delgada Fim	31.16 12.57	0,50	30,20

Total do Percurso 1 30,20 Km

P - Percurso

GPS - Coordenadas das paragens existentes



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A

Linha 1A

Santa Cruz - Fazenda de S. Cruz (Período Escolar)

P	Itinerário	GPS	Kms	Acumul.
1	Jardim S. Cruz	39.27.08 31.07.70	0,00	
2	Câmara	27.20 07.68	3,50	3,50
3	Escola	27.15 07.57	0,60	4,10
4	Jardim		0,30	4,40
5	Monte	27.71 08.20	2,00	6,40
6	Fazenda	28.09 08.72	1,30	7,70
7	Fazenda 1	27.42 09.25	1,38	9,08

Fazenda de S. Cruz - Santa Cruz (Período Escolar)

P	Itinerário	GPS	Kms	Acumul.
1	Fazenda 1	27.42 09.25	0,00	
2	Fazenda	28.09 08.72	1,38	1,38
3	Monte	27.71 08.20	1,30	2,68
4	Jardim		2,00	4,68
5	Escola	27.15 07.57	0,30	4,98

Nota: Coordenadas de GPS Norte/Oeste



REGIÃO AUTÓNOMA DOS AÇORES

SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

FUNDO REGIONAL DOS TRANSPORTES TERRESTRES, I. P. R. A



Fonte: Horários UT

Linhas/Horários

LINHA/LINE 2: SANTA CRUZ - LAJES

Sentido Ida/Going Way							
	E 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª	A1 2ª 3ª 4ª 5ª 6ª	C 3ª 5ª	E 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª	A 2ª 3ª 4ª 5ª 6ª
Jardim Santa Cruz	-	-	-	-	-	-	16:50
Topo Norte	-	-	-	-	-	-	16:52
Porto de São Pedro	-	-	-	-	-	-	16:53
Escritório - Oficina	-	-	-	-	-	-	16:54
Aeroporto	-	-	-	-	-	-	16:55
Camara	-	-	-	-	-	-	16:56
Escola	-	-	-	-	-	-	16:57
Porto Velho	-	-	-	-	-	-	16:58
Igreja	-	-	-	-	-	-	16:59
Jardim Santa Cruz	-	09:00	10:00	11:15	15:45	-	17:00
Rua Barqueiros	-	09:02	10:02	11:17	15:47	-	17:02
Boavista 1	-	09:04	10:04	11:18	15:49	-	17:04
Boavista 2	-	09:06	10:06	11:19	15:52	-	17:06
Rua da Aresta	-	09:08	10:08	11:21	15:58	-	17:08
Cruzamento Mato	-	09:12	10:12	11:23	16:02	-	17:12
Caveira	-	09:22	10:22	11:31	16:12	-	17:22
Lomba	08:45	09:35	10:35	11:44	16:27	17:00	17:35
Fazenda Jardim	08:53	09:42	10:42	11:50	16:35	17:08	17:42
Fazenda Agrícolas	08:55	09:44	10:44	11:51	16:36	17:10	17:44
Lajes Loura	08:57	09:47	10:47	11:53	16:38	17:12	17:47
Lajes Ponte	08:58	09:49	10:49	11:55	16:40	17:13	17:49
Lajes Escola	08:59	09:53	10:53	11:57	16:42	17:14	17:53
Lajes Germano	-	09:55	10:55	11:58	16:43	-	17:55
Lajes PSP	09:00	10:00	11:00	12:00	16:45	17:15	18:00

A = Efetuado anualmente/Performed annually

A1 = Exceto período escolar/ Except school season

C= Efetuado anualmente às 3ª e 5ª feira/ Performed annually at Tuesday and Thursday

E = Efetuado em período escolar/Performed in school season.

- = Sem passagem nesta paragem/No service at this stop

Minicircuito/Minicircuit S. Cruz

Sentido Volta/ Return Way						
	E 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª	A 2ª 3ª 4ª 5ª 6ª	C 3ª 5ª	A 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª
Lajes PSP	08:00	08:35	09:00	14:00	16:00	16:45
Lajes Germano	08:03	-	09:03	14:03	16:03	-
Lajes Escola	08:06	08:36	09:06	14:06	16:06	16:46
Lajes Ponte	08:09	08:39	09:09	14:09	16:09	16:49
Fazenda	08:10	08:40	09:10	14:10	16:10	16:50
Lomba	08:15	08:45	09:15	14:15	16:15	17:00
Caveira	08:23		09:23	14:23	16:23	
Cruzamento Mato	08:29		09:29	14:29	16:29	
Rua da Aresta	08:31		09:31	14:31	19:31	
Boavista	08:32		09:32	14:32	16:32	
Rua Barqueiros	08:35		09:35	14:35	16:35	
Topo Norte - S. Cruz	-		09:36	-	-	
Porto de São Pedro	-		09:36	-	-	
Escritório - Oficina	-		09:38	-	-	
Aeroporto	-		09:39	-	-	
Camara	-		09:41	-	-	
Escola	-		09:42	-	-	
Porto Velho	-		09:43	-	-	
Igreja	-		09:44	-	-	
Jardim Santa Cruz	08:40		09:45	14:40	16:40	

A = Efetuado anualmente/Performed annually

A1 = Exceto período escolar/ Except school season

C= Efetuado anualmente às 3ª e 5ª feira/ Performed annually at Tuesday and Thursday

E = Efetuado em período escolar/Performed in school season.

- = Sem passagem nesta paragem/No service at this stop

 Minicircuito/Minicircuit S. Cruz

Nota: em função do número de alunos a transportar em cada ano letivo poderá revelar-se necessário efetuar desdobramento (no percurso total ou parcial) da Carreira de Lajes/Santa Cruz (08:00h) e S. Cruz/Lajes (17:00h)

Itinerário e Paragens

Linha 2 Santa Cruz - Lajes

P	Itinerário	GPS	Km	Acumul.
1	Jardim S. Cruz	39.27.08 31.07.70	0,00	
2	Câmara	27.20 07.68	3,50	3,50
3	Escola	27.15 07.57	0,60	4,10
4	Jardim		0,30	4,40
5	R. Barqueiros	39.26.84 31.08.10	1,20	5,60
6	Boavista 1	26.83 08.30	0,30	5,90
7	Boavista 2	26.93 08.32	0,20	6,10
8	R. Aresta	26.73 08.32	0,80	6,90
9	Cruzamento Mato	25.90 08.86	0,80	7,70
10	Caveira	25.84 08.98	3,50	11,20
11	Lomba Terra Chã	24.28 09.32		
12	Lomba	24.10 09.27	5,90	17,10
13	Fazenda Barreira - Vermelha	23.39 09.48		
14	Fazenda Jardim	23.51 09.87	3,10	20,20
15	Fazenda S. Agrícolas	23.21 10.04	0,60	20,80
16	Lajes - Lourã	22.93 10.25	0,80	21,60
17	Lajes - Ponte	22.72 10.45	0,50	22,10
18	Lajes - Escola	22.63 10.61	0,50	22,60
19	Lajes Germano	22.75 10.89	0,40	23,00
20	Lajes PSP	22.64 10.50	1,00	24,00

Total do Percurso 2 24,00 Km
--

P - Percurso

GPS - Coordenadas das paragens existentes

Linha 2A
Lajes - Lomba

P	Itinerário	GPS	Kms	Acumul.
1	Lajes - PSP	39.22.64 31.10.50	0,00	
2	Lajes - Escola	22.63 10.61	0,20	0,20
3	Lajes - Ponte	22.72 10.45	0,50	0,70
4	Lajes - Lourã	22.93 10.25	0,50	1,20
5	Fazenda S. Agrícolas	23.21 10.04	0,80	2,00
6	Fazenda Jardim	23.51 09.87	0,60	2,60
7	Lomba - Centro	24.10 09.27	3,30	5,90

Total do Percurso 2A 5,90 Km

P - Percurso

GPS - Coordenadas das paragens existentes



Fonte: Horários UTC

Linhas/Horários

Linha/Line 3: Lajes - Fajã Grande

Sentido Ida/Going Way				
	A	E	A1	E
	2ª 3ª 4ª	2ª 3ª 4ª	2ª 3ª 4ª	2ª 3ª 4ª
	5ª 6ª	5ª 6ª	5ª 6ª	5ª 6ª
Lajes PSP	11:00	16:50	18:00	17:45
Lajes Escola	11:01	16:51	18:01	17:46
Cruzamento Lajedo	11:15	17:05	18:15	18:00
Lajedo	11:19	17:09	18:19	18:04
Costa Lajedo	11:23	17:13	18:32	18:08
Lajedo	11:27	17:17	18:26	-
Cruzamento Lajedo	11:30	17:20	18:29	-
Cruzamento Mosteiro	11:35	17:25	18:34	-
Mosteiro 1	11:36	17:26	18:35	-
Mosteiro 2	11:38	17:28	18:37	-
Cruzamento da Caldeira	11:42	17:32	18:41	-
Cruzamento Fajazinha	11:46	17:35	18:44	-
Fajazinha	11:48	14:38	18:47	-
Cruzamento Fajazinha	11:50	17:40	18:49	-
Cuada	11:53	17:43	18:52	-
Fajã Grande	11:55	17:45	18:55	-

A = Efetuado anualmente/Performed annually

A1 = Exceto período escolar/ Except school season

E = Efetuado em período escolar/Performed in school season.

- = Sem passagem nesta paragem/No service at this stop.

Sentido Volta/ Return Way				
	E 2ª 3ª 4ª 5ª 6ª	E 2ª 3ª 4ª 5ª 6ª	A1 2ª 3ª 4ª 5ª 6ª	A 2ª 3ª 4ª 5ª 6ª
Fajã Grande Fim	-	07:45	08:00	15:00
Cuada	-	07:47	08:02	15:02
Cruzamento Fajazinha	-	07:50	08:05	15:05
Fajazinha	-	07:52	08:07	15:07
Cruzamento Fajazinha	-	07:54	08:09	15:09
Cruzamento da Caldeira	-	08:58	08:13	15:13
Mosteiro 2	-	08:02	08:17	15:17
Mosteiro 1	-	08:04	08:19	15:19
Cruzamento Mosteiro	-	08:05	08:20	15:20
Cruzamento Lajedo	-	08:18	08:25	15:25
Lajedo	-	08:21	08:28	15:28
Costa Lajedo	07:45	08:26	08:34	15:34
Lajedo	07:50	08:29	08:37	15:37
Cruzamento Lajedo	07:53	08:32	08:41	15:41
Lajes Escola	07:59	08:35	08:54	15:54
Lajes PSP	08:00	08:35	08:55	15:50

A = Efetuado anualmente/Performed annually

A1 = Exceto período escolar/ Except school season

E = Efetuado em período escolar/Performed in school season.

- = Sem passagem nesta paragem/No service at this stop.

Itinerário e Paragens

Linha 3

Lajes - Fajã Grande

P	Itinerário	GPS	Km	Acumul.
1	Lajes - PSP	39.22.64 31.10.50	0,00	
2	Lajes Escola	22.63 10.61	0,20	0,20
3	Lajes Germano	22.75 10.84	0,50	0,70
4	Lajes ER Morros e Monte	22.54 11.06		
5	Cruzamento Lajedo	23.81 14.68	6,80	7,50
6	Lajedo	23.54 14.68	1,50	9,00
7	Costa do Lajedo	22.98 14.45	2,20	11,20
8	Lajedo		2,20	13,40
9	Cruzamento Lajedo		1,50	14,90
10	Cruzamento Mosteiro		2,20	17,10
11	Mosteiro 1	24.72 14.90	0,60	17,70
12	Mosteiro 2	24.79 15.08	1,40	19,10
13	Cruzamento Caldeira	25.69 15.41	1,80	20,90
14	Cruzamento Fajazinha	25.93 14.78	1,90	22,80
15	Fajazinha	25.97 15.25	1,20	24,00
16	Cruzamento Fajazinha		1,20	25,20
17	Cuada	26.51 15.15	1,70	26,90
18	Fajã Grande 1	27.03 15.39		
19	Fajã Grande 2	27.20 15.72	1,70	28,60
20	Fajã Grande Fim	27.49 15.80	0,90	29,50

Total do Percurso 3

29,50 Km

P - Percurso

GPS - Coordenadas das paragens existentes



Fonte: Horários UTC



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Linhas/Horários

LINHA/LINE 4: SANTA CRUZ - FAJÃ GRANDE (VIA COSTA)

Sentido Ida/Going Way	
	E
	2ª 3ª 4ª 5ª 6ª
Jardim Santa Cruz	17:00
Rua Barqueiros	17:02
Boavista	17:03
Rua da Aresta	17:04
Cruzamento Mato	17:06
Cruzamento Fajãs	17:18
Cruzamento Mosteiro	17:21
Mosteiro 1	17:22
Mosteiro 2	17:24
Cruzamento da Caldeira	17:28
Cruzamento Fajazinha	17:34
Fajazinha	17:38
Cruzamento Fajazinha	14:40
Cuada	17:43
Fajã Grande	17:45

Sentido Volta/ Return Way	
	E
	2ª 3ª 4ª 5ª 6ª
Fajã Grande	08:00
Cuada	08:02
Cruzamento Fajazinha	08:09
Fajazinha	08:11
Cruzamento Fajazinha	08:17
Cruzamento da Caldeira	08:20
Mosteiro 2	08:24
Mosteiro 1	08:26
Cruzamento Mosteiro	08:27
Cruzamento Fajãs	08:30
Cruzamento Mato	08:32
Rua da Aresta	08:44
Boavista	08:45
Rua Barqueiros	08:46
Jardim Santa Cruz	08:50

E = Efetuado em período escolar/Performed in school season.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Itinerário e Paragens

Linha 4
Santa Cruz - Fajã Grande (Via Costa)

P	Itinerário	GPS	Kms	Acumul.
1	Jardim S. Cruz	39.27.08 31.07.70	0,00	
2	Câmara	27.20 07.68	3,50	3,50
3	Escola	27.15 07.57	0,60	4,10
4	Jardim		0,30	4,40
5	R. Barqueiros	39.26.84 31.08.10	1,20	5,60
6	Boavista 1	26.83 08.30	0,30	5,90
7	Boavista 2	26.93 08.32	0,20	6,10
8	R. Aresta	26.73 08.32	0,80	6,90
9	Cruzamento Mato	25.90 08.86	0,80	7,70
10	Cruzamento Fajãs	25.30 14.62	11,40	19,10
11	Cruzamento Mosteiro	24.71 14.67	1,70	20,80
12	Cruzamento Lajedo	23.81 14.68	2,20	23,00
13	Lajedo	23.54 14.68	1,50	24,50
14	Costa do Lajedo	22.98 14.45	2,20	26,70
15	Lajedo		2,20	28,90
16	Cruzamento Lajedo		1,50	30,40
17	Cruzamento Mosteiro		2,20	32,60
18	Mosteiro 1	24.72 14.90	0,60	33,20
19	Mosteiro 2	24.79 15.08	1,40	34,60
20	Cruzamento Caldeira	25.69 15.41	1,80	36,40
21	Cruzamento Fajazinha	25.93 14.78	1,90	38,30
22	Fajazinha	25.97 15.25	1,20	39,50



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

23	Cruzamento Fajazinha		1,20	40,70
24	Cuada	26.51 15.15	1,70	42,40
25	Fajã Grande 1	27.03 15.39	1,70	44,10
26	Fajã Grande 2	27.20 15.72		
27	Fajã Grande Fim	27.49 15.80	0,90	45,00

Total do Percurso 4 45,00 Km

P - Percurso

GPS - Coordenadas das paragens existentes



Fonte: Horários UTC



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Linhas/Horários

LINHA/LINE 5: SANTA CRUZ - FAJÃ GRANDE (VERÃO/SUMMER)

Sentido Ida/Going Way		
	V 2ª 3ª 4ª 5ª 6ª	V 2ª 3ª 4ª 5ª 6ª
Jardim Santa Cruz	-	-
Topo Norte	-	-
Porto de São Pedro	-	-
Escritório - Oficina	8:30	18:00
Aeroporto	8:31	18:01
Camara	8:32	18:02
Escola	8:32	18:02
Porto Velho	8:33	18:03
Igreja	8:34	18:04
Jardim Santa Cruz	08:35	18:05
Rua Barqueiros	08:37	18:07
Boavista	08:39	18:09
Rua da Aresta	08:41	18:11
Cruzamento Mato	08:45	18:15
Cruzamento Fajãs	08:55	18:25
Cruzamento da Caldeira	08:58	18:28
Cruzamento Fajazinha	09:03	18:33
Cuada	09:05	18:35
Fajã Grande	09:10	18:40



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Sentido Volta/ Return Way		
	V 2 ^a 3 ^a 4 ^a 5 ^a 6 ^a	V 2 ^a 3 ^a 4 ^a 5 ^a 6 ^a
Fajã Grande	09:15	18:45
Cuada	09:20	18:50
Cruzamento Fajazinha	09:22	18:52
Cruzamento da Caldeira	09:27	18:57
Cruzamento Fajãs	09:30	19:00
Cruzamento Mato	09:40	19:10
Rua da Aresta	09:44	19:14
Boavista	09:46	19:16
Rua Barqueiros	09:47	19:17
Jardim Santa Cruz	09:50	19:20

V = Efetuado em julho e agosto/ Performed in July and August.

 Minicircuito/Minicircuit S. Cruz



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Itinerário e Paragens

Linha 5
Santa Cruz - Fajã Grande (Horário Verão)

P	Itinerário	GPS	Kms	Acumul.
1	Jardim S. Cruz	39.27.08 31.07.70	0,00	
2	R. Barqueiros	26.84 08.10	1,20	1,20
3	Boavista 1	26.83 08.30	0,30	1,50
4	Boavista 2	26.93 08.32	0,20	1,70
5	R. Aresta	26.73 08.32	0,80	2,50
6	Cruzamento Mato	25.90 08.86	0,80	3,30
7	Cruzamento Fajãs	25.30 14.62	11,40	14,70
8	Cruzamento Caldeira	25.69 15.41	1,70	16,40
9	Cruzamento Fajazinha	25.93 14.78	1,70	18,10
10	Cuada	26.51 15.15	1,70	19,80
11	Fajã Grande 1	27.03 15.39		
12	Fajã Grande 2	27.20 15.72	1,70	21,50
13	Fajã Grande Fim	27.49 15.80	0,90	22,40

Total do Percorso 5 22,40 Kms

P - Percorso

GPS - Coordenadas das paragens existentes



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes



Fonte: Horários UTC



Linhas/Horários

LINHA/LINE 6: Verão Atlânticoline

Esta Linha efetua-se nos meses de julho e agosto, nos dias de chegada e partida do navio da Atlânticoline (linha Amarela), nas condições abaixo indicadas:

- A Linha 6 é realizada com recurso a duas Linhas já existentes:
 - Linha 2 - Lajes/Santa Cruz e
 - Linha 3 Lajes/Fajã Grande



Linhas Lajes/Fajã Grande e Lajes/Santa Cruz (Ida)

As carreiras iniciam-se nas Lajes à hora de chegada do navio. Os inícios das carreiras efetuam-se quando os passageiros já tiverem desembarcado.

Fajã Grande/Lajes e Santa Cruz/Lajes (volta)

As carreiras iniciam-se na Fajã Grande e Santa Cruz uma 1:30 h antes da hora prevista e chegada do Navio seguindo o mesmo itinerário no regresso às Lajes.

Notas:

- A carreira da Linha 3 Lajes/Fajã Grande (e circuito inverso) não efetua passagem na Costa e Lajedo.
- As carreiras com origem e destino a Santa Cruz efetuam o Minicircuito de Santa Cruz



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Horários Atlânticoline 2019 (Flores)

Operação Sazonal 2019

Ida			Volta		
Data	Trajetos	Hora de Chegada	Data	Trajetos	Hora de Partida
18/07/2019	Faial/Flores	09:30	22/07/2019	Flores/Faial	21:30
01/08/2019	Faial/Flores	09:30	05/08/2019	Flores/Faial	21:30
12/08/2019	Faial/Flores	12:30	17/08/2019	Flores/Faial	21:30
23/08/2019	Faial/Flores	09:30	26/08/2019	Flores/Faial	21:30
31/08/2019	Faial/Flores	09:45			



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Minicircuito/Minicircuit S. Cruz

Algumas linhas constantes do presente anexo (Linhas 1, 2 e 6) apresentam, à chegada/partida de Santa Cruz uma expansão de itinerário designado por “Minicircuito de Santa Cruz”.

Com exceção da Linha 6 – Altanticoline (cujos horários podem variar em função da programação anual do serviço de verão da Altanticoline), as carreiras e horários que efetuam o minicircuito estão identificados com:

Minicircuito/Minicircuit S. Cruz



Fonte: UTC Horários

O minicircuito tem uma extensão aproximada de 4,5 km.

Linha 1 – Santa Cruz/Ponta delgada

As carreiras efetuam o minicircuito no seguinte itinerário:

Sentido Ida (S. Cruz/P. Delgada)

Jardim Santa Cruz
Topo Norte
Porto de São Pedro
Escritório - Oficina
Aeroporto
Camara
Escola
Porto Velho
Igreja
Jardim Santa Cruz
Monte/Ponta Delgada

Sentido Volta (P. Delgada/S. Cruz)

Ponta Delgada/Monte
Topo Norte
Porto de São Pedro
Escritório - Oficina
Aeroporto
Camara
Escola
Porto Velho
Igreja
Jardim Santa Cruz



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

Linhas 2 e 6 - Santa Cruz/Lajes

As carreiras efetuam o minicircuito no seguinte itinerário:

Sentido Ida (S. Cruz/Lajes)

Jardim Santa Cruz
Topo Norte
Porto de São Pedro
Escritório - Oficina
Aeroporto
Câmara
Escola
Porto Velho
Igreja
Jardim Santa Cruz
Barqueiros/Lajes

Sentido Volta (P. Delgada/S. Cruz)

Lajes/Barqueiros
Topo Norte
Porto de São Pedro
Escritório - Oficina
Aeroporto
Câmara
Escola
Porto Velho
Igreja
Jardim Santa Cruz

Linhas 5- Santa Cruz/Fajã Grande (Verão)

Sentido Ida (S. Cruz/Fajã Grande)

Escritório
Aeroporto
Câmara
Escola
Porto Velho
Igreja
Jardim Santa Cruz
Barqueiros/Fajã Grande



ANEXO II TÍTULOS DE TRANSPORTE

1- O cocontratante fica obrigado a disponibilizar os títulos de transporte e praticar os descontos comerciais abaixo indicados:

- a) **Bilhete simples de bordo (BS)** – Bilhete simples válido para uma viagem, para determinada origem e destino;
- b) **Meio bilhete de bordo (MB)** – Bilhete corresponde a 50% do BS, arredondado para o valor da tarifa em vigor que lhe for mais próxima, não podendo, contudo, ser inferior ao mínimo de cobrança fixado para cada ano (ano de 2019 o mínimo de cobrança é de € 0,70), sendo válido para crianças dos 4 aos 11 anos, inclusive);
- c) **Pré-comprado de 10 viagens 3ª Idade e Pensionista por Invalidez (PC 10V 3IP)** – Bilhete adquirido previamente à sua utilização, em local de venda do cocontratante, em conjuntos de 10, para uma dada origem e destino, calculados com base num desconto de 50% sobre o BS, arredondado o valor de cada bilhete ao cêntimo de euro;

Considera-se:

- 3ª Idade – Os utentes com idade igual ou superior a 65 anos;
 - Pensionista por Invalidez – Os utentes que apresentem o respetivo Cartão de Pensionista por Invalidez
- d) **Pré-comprado de 10 viagens (PC 10V)** – Bilhete adquirido previamente à sua utilização, em conjuntos de 10, para uma dada origem e destino, calculados com base num desconto de 10% sobre o BS, arredondado o valor de cada bilhete ao cêntimo de euro;
 - e) **Passe rede geral 1 dia (P1 Dia)** – Válido para toda a rede, durante 24h após a primeira validação, calculado com base no dobro do valor do preço do BS mais elevado em vigor, arredondado ao cêntimo de euro;
 - f) **Passe rede geral 3 dia (P3 Dia)** – Válido para toda a rede, durante 72 horas após a primeira validação, calculado com base no preço do P1 Dia multiplicado por 3, com desconto de 10%, arredondado ao cêntimo de euro;
 - g) **Passe rede geral 5 dia (P5 Dia)** – Válido para toda a rede, durante 120 horas após a primeira validação, calculado com base no preço do P1 Dia multiplicado por 5, com desconto de 15%, arredondado ao cêntimo de euro;
 - h) **Passe 30 Dias (P30 Dias)** – Modalidade válida para um número ilimitado de viagens, calculado com base na tarifa simples (BS) entre a zona de origem e destino, válido durante



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

30 dias após a primeira validação. O valor deste passe é calculado com base no valor de 44 viagens (BS) com uma redução de acordo com a percentagem de desconto constante da tabela constante do Anexo IV

- i) **Passe para desempregados (P DES)** – Válido para um número ilimitado de viagens, dentro da origem e destino a que se refere o passe, durante um mês após a primeira validação. Poderá ser renovado até 2 meses e não deverá ser emitido mais do que uma vez por ano por pessoa. O valor deste passe é calculado com base no valor P30Dias com descontos de 25%, arredondado ao cêntimo de euro.
- j) **Passe de 3ª idade, Pensionistas e Invalidez (P 3IP)** – Válido para um número ilimitado de viagens, dentro da origem e destino a que se refere o passe, durante um mês após a primeira validação. O valor deste passe é calculado com base no valor do P30 Dias com uma redução de 25% sobre o preço final, arredondado ao cêntimo de euro, destinado a utentes com idade igual ou superior a 65 anos, aposentados e pensionistas ou aposentados por invalidez.
- k) **Passe Escolar (P EST)** – Modalidade destinada a alunos dos Estabelecimentos de Ensino transportados entre o local de residência e o estabelecimento de ensino em carreira pública (serviço regular), cujas condições de utilização e respetivos preços são suportados diretamente pelos referidos estabelecimentos – através da Secretaria Regional da Educação e Cultura - sendo o valor do passe mensal calculado com base em duas viagens por cada dia escolar, de acordo com a formula de cálculo apresentada no número seguinte.

2 – O preço dos passes a que se refere as alíneas h), i), j) e k) do número anterior são calculados de acordo com as formulas seguintes:

Passe 30 dias (P30 Dias)

$TS \times 44 \text{ viagens} \times (1-d)$

Sendo que:

TS – Tarifa Simples

d – percentagem de desconto constante da tabela a que se refere o Anexo IV

Passe Mensal de 3.ª Idade, Pensionista e Invalidez (P 3IP) e

Passe Desempregado (P DES):

$0,75 \times (TS \times 44 \text{ viagens} \times (1-d))$

Sendo que:



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

TS – Tarifa Simples

d – percentagem de desconto constante da tabela a que se refere o Anexo IV

Passe Escolar (P EST)

(Dias escolares/mês x 2 viagens/dia x Bilhete Simples*)

*Para efeitos de calculo do P EST aos alunos com menos de 12 anos o bilhete simples é substituído pelo Meio Bilhete de Bordo (MB).

3 – O Fundo Regional de Ação Social Escolar (Secretaria Regional da Educação e Cultura) suportará o custo da emissão do primeiro cartão-passe de cada aluno.

4- O cocontratante poderá ainda disponibilizar “Cartões Valor”, ou seja, cartões reutilizáveis, pré-carregados com determinada importância, sendo o valor de cada viagem realizada deduzida ao saldo do valor em cartão.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

ANEXO III TARIFÁRIO

Linha 1 - Stª Cruz - P. Delgada

Santa Cruz

0,82	Fazenda Stª Cruz		Euros
1,11	0,82	Cedros	
1,90	1,24	0,82	
2,52	2,21	1,57	Ponta Ruiva
			P. Delgada

Crianças dos 4 aos 11 anos inclusive pagam 1/2 passagem
Mínimo de Cobrança = 0,70 €

2 - Stª Cruz - Lajes

3 - Lajes/ Fajã Grande

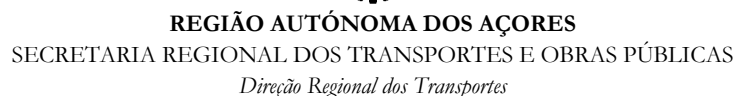
4 - Fajã Grande Stª Cruz (Via Mato)

5 - Stª Cruz – Fajã Grande (Verão)

Fajã Grande

0,94	Fajãzinha									
1,24	0,94	Mosteiro								
1,90	1,44	0,94	Lajedo							
2,12	1,75	1,05	0,82	Costa						
2,38	2,12	1,44	1,11	1,44	Lajes					
2,52	2,38	1,90	1,57	1,90	0,82	Fazenda das Lajes				
2,78	2,52	2,21	2,07	2,21	0,94	0,82	Lomba			
3,25	2,95	2,56	2,52	2,56	1,57	1,11	0,94	Caveira		
1,05	0,82	0,82	0,94	1,05	1,44				Cruzamento Fajãs	
2,52	2,12	2,12	2,21	2,38	2,12	1,75	1,24	0,82	1,57	Cruzamento Mato
2,56	2,52	2,52	2,52	2,56	2,38	2,12	1,75	0,94	2,07	0,82 Sta. Cruz

Crianças dos 4 aos 11 anos inclusive pagam 1/2 passagem
Mínimo de Cobrança = 0,70 €



Linha 1 - Stª Cruz - P. Delgada

Santa Cruz					Km
3,30	Fazenda Stª Cruz				
8,30	5,00	Cedros			
13,20	9,90	4,90	Ponta Ruiva		
20,00	16,70	11,70	11,20	P. Delgada	

5 - Sta. Cruz – Fajã Grande (Verão)

Fajã Grande											Km
5,50	Fajazinha										
9,40	6,30	Mosteiro									
13,70	10,60	5,70	Lajedo								
15,90	12,80	7,90	2,20	Costa							
19,00	15,90	11,00	8,30	10,50	Lajes						
21,80	18,70	13,80	11,10	13,30	2,80	Fazenda Lajes					
24,90	21,80	16,90	14,20	16,40	5,90	3,10	Lomba				
30,80	27,70	22,80	20,10	22,30	11,80	9,00	5,90	Caveira			
7,70	4,60	3,70	5,40	7,60	10,70	---	---	---	Cruza. Fajãs		
19,10	16,00	15,10	16,80	19,00	15,30	12,50	9,40	3,50	11,40	Cruza. Mato	
22,40	19,30	19,60	20,10	22,30	18,60	15,80	12,70	6,80	14,70	3,30	S. Cruz

Em caso de ausência de elementos de identificação da zona e desde que devidamente justificado o adjudicatário utilizar-se-á a tabela de tarifário por km abaixo apresentada, referente ao ano de 2019.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

TARIFÁRIO 2019

Km	Tarifário	Km	Tarifário
1	0,82	44	4,09
2	0,82	45	4,09
3	0,82	46	4,19
4	0,82	47	4,19
5	0,82	48	4,19
6	0,94	49	4,36
7	0,94	50	4,36
8	1,05	51	4,36
9	1,11	52	4,39
10	1,24	53	4,39
11	1,44	54	4,39
12	1,57	55	4,45
13	1,75	56	4,45
14	1,90	57	4,45
15	2,07	58	4,45
16	2,12	59	4,45
17	2,21	60	4,58
18	2,26	61	4,58
19	2,38	62	4,58
20	2,52	63	4,58
21	2,52	64	4,69
22	2,52	65	4,69
23	2,56	66	4,69
24	2,66	67	4,69
25	2,78	68	4,79
26	2,78	69	4,79
27	2,95	70	4,79
28	2,95	71	4,79
29	3,14	72	4,99
30	3,14	73	4,99
31	3,25	74	4,99
32	3,25	75	4,99
33	3,41	76	5,04
34	3,41	77	5,04
35	3,56	78	5,04
36	3,56	79	5,04
37	3,73	80	5,04
38	3,73	81	5,04
39	3,82	82	5,04
40	3,90	83	5,04
41	3,99	84	5,04
42	3,99	85	5,04
43	4,09		

Crianças dos 4 aos 11 anos pagam Meio Bilhete
(Mínimo de cobrança é de 0,70 Euros)



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

ANEXO IV
TABELA DESCONTOS PASSES MENSAIS

Intervalo de Distância (km)			%	PASSES	
De	Até	T Simples		P30 Dias	P 3IP e PDES
0	5	0,82 €	30,000%	25,26€	18,95€
5,00001	7	0,94 €	30,000%	28,95€	21,71€
7,00001	8	1,05 €	30,000%	32,34€	24,26€
8,00001	9	1,11 €	30,000%	34,19€	25,64€
9,00001	10	1,24 €	30,000%	38,19€	28,64€
10,00001	11	1,44 €	34,461%	41,53€	31,15€
11,00001	12	1,57 €	38,636%	42,39€	31,79€
12,00001	13	1,75 €	43,182%	43,75€	32,81€
13,00001	14	1,90 €	45,939%	45,20€	33,90€
14,00001	15	2,07 €	49,140%	46,32€	34,74€
15,00001	16	2,12 €	49,761%	46,86€	35,15€
16,00001	17	2,21 €	51,102%	47,55€	35,66€
17,00001	18	2,26 €	51,395%	48,33€	36,25€
18,00001	19	2,38 €	52,625%	49,61€	37,21€
19,00001	22	2,52 €	54,747%	50,18€	37,64€
22,00001	23	2,56 €	54,941%	50,75€	38,06€
23,00001	24	2,66 €	55,882%	51,64€	38,73€
24,00001	26	2,78 €	57,284%	52,25€	39,19€
26,00001	28	2,95 €	59,194%	52,97€	39,73€
28,00001	30	3,14 €	60,692%	54,31€	40,73€
30,00001	32	3,25 €	61,706%	54,76€	41,07€
32,00001	34	3,41 €	62,444%	56,35€	42,26€
34,00001	36	3,56 €	63,778%	56,74€	42,56€
36,00001	38	3,73 €	64,480%	58,30€	43,73€
38,00001	39	3,82 €	64,912%	58,98€	44,24€
39,00001	40	3,90 €	65,225%	59,67€	44,75€
40,00001	42	3,99 €	65,623%	60,35€	45,26€
42,00001	45	4,09 €	66,188%	60,85€	45,64€
45,00001	48	4,19 €	66,545%	61,68€	46,26€
48,00001	51	4,36 €	67,132%	63,05€	47,29€
51,00001	54	4,39 €	67,235%	63,29€	47,47€
54,00001	59	4,45 €	67,532%	63,57€	47,68€
59,00001	63	4,58 €	68,149%	64,19€	48,14€
63,00001	67	4,69 €	68,431%	65,15€	48,86€
67,00001	71	4,79 €	68,849%	65,65€	49,24€
71,00001	75	4,99 €	69,720%	66,48€	49,86€
75,00001	80	5,04 €	69,765%	67,05€	50,29€

P30 Dias = Passe 30 Dias

P3IP = Passe 3ª Idade, Pensionista e Invalidez

P DES = Passe Desempregado



ANEXO V TIPOLOGIA DE VEÍCULOS

Tipologia e Total de Veículos a afetar à exploração

Abrev	Tipologia	Lotação*
Mini	Minibus (Só Lugares Sentados)	Lotação \geq 15 lugares e \leq 28 lugares
Med	Autocarro interurbano médio	\geq 29 lugares e \leq 49 lugares sentados
Inter.	Autocarro Standard interurbano	\geq 50 Lugares sentados

*Não inclui motorista

Tipologia de Veículos a afetar à exploração	Quant.
Autocarro Standard interurbano	2
Autocarro Médio interurbano	1
Minibus (Lugares sentados)	3
Autocarros de Reserva:	
Autocarro Standard interurbano	1
Minibus (Lugares sentados)	1
Total Veículos a afetar à exploração	8

Informação Adicional:

1. É obrigatório a apresentação de dois autocarros de reserva, conforme definido no quadro supra.
2. Não são admitidos à exploração veículos da Classe III, veículos articulados ou de 2 pisos.
3. Os veículos standard interurbanos a afetar à exploração deverão apresentar dimensões entre os 12,0 a 12,5 m de comprimento e deverão ser afetos às linhas com maior afluência de alunos (circuitos escolares pendulares), com especial relevância para as linhas 1 e 2.
4. Devido à exiguidade de algumas vias, nomeadamente o itinerário da Linha 1A (zona da Lomba da Fazenda de St^a Cruz), Linha 3 (zona do Lajedo, Costa do Lajedo, Mosteiro) as carreiras devem ser efetuadas com veículos da tipologia minibus ou veículo médio, consoante o itinerário.
5. A afetação dos veículos à exploração será da responsabilidade do cocontratante e deverá adequar-se à procura prevista.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

ANEXO VI QUILOMETRAGEM ANUAL

1- Ponta Delgada/ Santa Cruz (ida/Volta)	N.º Carreiras		Km por Carreira	250 dias uteis anuais	180 dias uteis P. escolar	70 dias uteis não escolares
	Dia	Semana				
Carreiras anuais diárias (2ª a 6ª feira)	4	20	30,20	30 200,00	-	-
Carreiras anuais realizadas às 2ª, 4ª e 6ª Feira	1,2	6	30,20	9 060,00	-	-
1A P. Delgada/Fazenda (2ª a 6ª feira P. Escolar)	4	20	7,50	-	5 400,00	-
Linha 1	-	46	67,90	39 260,00	5 400,00	-

1- Ponta Delgada/ Santa Cruz	44 660,00
-------------------------------------	------------------

2 - Santa Cruz/Lajes (Ida/Volta)	N.º Carreiras		Km por Carreira	250 dias uteis anuais	180 dias uteis P. escolar	70 dias uteis não escolares
	Dia	Semana				
Carreiras anuais diárias (2ª a 6ª feira)	3	15	24,00	18 000,00	-	-
Carreiras anuais realizadas às 3ª e 5ª Feira	0,8	4	24,00	4 800,00	-	-
Carreira em Período Escolar (2ª a 6ª feira)	3	15	24,00	-	12 960,00	-
2A - Lajes/Lomba (2ª a 6ª feira Período Escolar)	4	20	2,80	-	2 016,00	-
Carreira em Período Não Escolar	1	5	24,00	-	4 320,00	1 680,00
Linha 2	-	59	98,80	22 800,00	19 296,00	1 680,00

2 - Santa Cruz/Lajes	43 776,00
-----------------------------	------------------

3 - Lajes/Fajã Grande (Ida/Volta)	N.º Carreiras		Km por Carreira	250 dias uteis anuais	180 dias uteis P. escolar	70 dias uteis não escolares
	Dia	Semana				
Carreiras anuais diárias (2ª a 6ª feira)	2	10	29,50	14 750,00	-	-
Carreira em Período Não Escolar	2	10	29,50	-	-	4 130,00
Carreira em Período Escolar (2ª a 6ª feira)	2	10	29,50	-	10 620,00	-
3A- Lajes/Costa Lajedo (desdob.) Período Escolar	2	10	11,20	-	4 032,00	-
Linha 3	-	40	99,70	14 750,00	14 652,00	4 130,00

3 - Lajes/Fajã Grande	33 532,00
------------------------------	------------------

4- F. Grande/S. Cruz (via Costa) (Ida/Volta)	N.º Carreiras		Km por Carreira	250 dias uteis anuais	180 dias uteis P. escolar	70 dias uteis não escolares
	Dia	Semana				
Carreira em Período Escolar (2ª a 6ª feira)	2	10	45,00	-	16 200,00	-
Linha 4	-	10	45,00	0,00	16 200,00	0,00

4 - Lajes/Fajã Grande	16 200,00
------------------------------	------------------

5- F. Grande/S. Cruz (Verão) (Ida/Volta)	N.º Carreiras		Km por Carreira			44 dias uteis Hor. Verão
	Dia	Semana				
Verão entre Julho e Agosto (2ª a 6ª feira)	4	20	22,40	-	-	3 942,40
Linha 5	-	20	22,40	0,00	0,00	3 942,40

5 - Lajes/Fajã Grande (verão)	3 942,40
--------------------------------------	-----------------

6- Atlânticoline (Verão) (Ida/Volta)*	N.º Carreiras		Km por Carreira			Total Quilómetros
	Ida	Volta				
2- Lajes/Santa Cruz	10	8	24,00			432,00
3- Lajes Fajã Grande	10	8	29,50			531,00
Linha 6	20	16	53,50	0,00	0,00	963,00

6 - Atlânticoline (Verão)	963,00
----------------------------------	---------------

* Tendo por base os Horários da operação Sazonal da Atlânticoline 2019

Total Quilómetros Anuais Previstos	143 073,40
---	-------------------



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

ANEXO VII
QUADRO DE PESSOAL DA EMPRESA
UNIÃO DE TRANSPORTES DOS CARVALHOS, LDA. (UTC – Açores)

Origem	Profissão	Depart	Data Nascimento	Admissão	Tipo Contrato	Regime Horário	BASE (anual)	BASE (mensal)	DIUT	AU	SF+SN+F	Sub Total	TSU	TSU	Subsidios	Abono Falhas	Seg AT	HSST	Seg Saúde	Sub Total	CUSTO ANUAL
UTC	Motorista	Flores	24/12/1962	01/08/2010	Sem termo	Total	9 780,00 €	815,00 €	299,04 €		1 679,84 €	11 758,88 €	23,75%	2 792,73 €	801,36 €		394,13 €	95,20 €	0,00 €	1 290,69 €	15 842,30 €
UTC	Motorista	Flores	20/02/1969	01/08/2010	Sem termo	Total	9 780,00 €	815,00 €	299,04 €		1 679,84 €	11 758,88 €	23,75%	2 792,73 €	786,24 €		394,13 €	95,20 €	0,00 €	1 275,57 €	15 827,18 €
UTC	Motorista	Flores	28/09/1977	01/08/2010	Sem termo	Total	9 780,00 €	815,00 €	299,04 €		1 679,84 €	11 758,88 €	23,75%	2 792,73 €	839,16 €		394,13 €	95,20 €	0,00 €	1 328,49 €	15 880,10 €
UTC	Motorista	Flores	26/02/1960	01/08/2010	Sem termo	Total	9 780,00 €	815,00 €	299,04 €		1 679,84 €	11 758,88 €	23,75%	2 792,73 €	858,06 €		394,13 €	95,20 €	0,00 €	1 347,39 €	15 899,00 €
UTC	Motorista	Flores	09/10/1974	18/10/2010	Sem termo	Total	9 780,00 €	815,00 €	299,04 €		1 679,84 €	11 758,88 €	23,75%	2 792,73 €	850,50 €		394,13 €	95,20 €	0,00 €	1 339,83 €	15 891,44 €
UTC	Motorista	Flores	09/02/1982	29/11/2010	Sem termo	Total	9 780,00 €	815,00 €	299,04 €		1 679,84 €	11 758,88 €	23,75%	2 792,73 €	725,76 €		394,13 €	95,20 €	0,00 €	1 215,09 €	15 766,70 €
UTC	Chefe Dep	ADM/FLORES	25/04/1971	01/06/2006	Sem termo	Total	12 474,12 €	1 039,51 €	323,28 €		2 132,90 €	14 930,30 €	23,75%	3 545,95 €	861,84 €	1 012,20 €	394,13 €	95,20 €	0,00 €	2 363,37 €	20 839,62 €
																					115 946,36 €



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS
Direção Regional dos Transportes

ANEXO VIII
BILHÉTICA
(ficheiro separado)



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

ANEXO VIII

ANEXO TÉCNICO

DO SISTEMA DE BILHÉTICA, DE APOIO À EXPLORAÇÃO E DE INFORMAÇÃO AO PÚBLICO



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Índice

1	objecto.....	5
2	enquadramento	6
3	faseamento.....	9
3.1	FASE 0 - Proprietário	10
3.1.1	Contexto Institucional	10
3.1.2	Arquitectura de referência para o SBO	11
3.1.2.1	Títulos de transporte versus tecnologia de suporte.....	11
3.1.2.2	Integração com a AT (Autoridade Tributária)	12
3.1.2.3	Regulamento Geral sobre a Proteção de Dados	12
3.1.3	Elementos de Interoperabilidade	12
3.1.4	SAE.....	13
3.1.5	SINFO.....	13
3.2	FASE 1 - Interoperável	14
3.2.1	Contexto Institucional	14
3.2.2	Arquitectura de referência para o SBO	14
3.2.2.1	Títulos de transporte versus tecnologia de suporte	15
3.2.3	Elementos de Interoperabilidade	15
3.2.3.1	Calypso Portable Object: Arquitectura de Segurança	16
3.2.3.2	Calypso Portable Object: Cartão Calypso	18
3.2.3.3	Modelo de Dados.....	19
3.2.3.3.1	Modelo de Dados: Mapeamento dos Títulos de Transporte	20
3.2.3.3.2	Modelo de Dados: Interface de Programação (API)	20
3.2.4	SAE.....	22
3.2.5	SINFO.....	25
3.3	FASE 2 - Intermodal	28
3.3.1	Contexto Institucional	29
3.3.2	Arquitectura de referência para o SBO	29
3.3.2.1	Títulos de transporte versus tecnologia de suporte.....	30
3.3.3	Elementos de Interoperabilidade	30
3.3.4	SAE.....	30
3.3.5	SINFO.....	30
4	ANEXOS.....	32
	ANEXO A. “Cartões, Bilhetes e SAM: tipologias e preços de referência”	
	ANEXO B. “Entidades e Responsabilidades”	
	ANEXO C. “Modelos de Dados ANDANTE”	
	ANEXO D “API Andante”	
	ANEXO E “Interface de Dados entre SBO e SCiiM”	



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Índice de Figuras

Figura 1 – Arquitectura Institucional e Elementos de Interoperabilidade.....	6
Figura 2 – FASE 0: Diagrama de Contexto Institucional	11
Figura 3 – FASE 1: Diagrama de Contexto Institucional	14
Figura 4 – Segurança num Sistema de Bilhética Calypso	17
Figura 5 – FASE 2: Diagrama de Contexto Institucional	28



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Índice de Tabelas

Tabela 1 – Faseamento	10
Tabela 2 – Fase 0: Níveis IFMS (<i>Interoperable Fare Management System</i>)	11
Tabela 3 – FASE 0/1: Tipologia de Títulos Transporte versus Tecnologia	12
Tabela 4 – Fase 1: Níveis IFMS (<i>Interoperable Fare Management System</i>)	15
Tabela 5 – Fase 2: Níveis IFMS (<i>Interoperable Fare Management System</i>)	30
Tabela 6 – FASE 2: Tipologia de Títulos Transporte versus Tecnologia	30



1 objecto

O presente anexo descreve os requisitos mínimos do Sistema de Bilhética (SBO) bem como dos subsistemas associados, designadamente Sistema de Apoio à Exploração (SAE) e Sistema de Informação ao Público (SINFO), a utilizar pelo concessionário no suporte comercial à operação da concessão de transportes coletivos de passageiros.

Este documento não pretende ser uma especificação completa dos referidos subsistemas (SBO, SAE e SINFO), mas tão só definir as características mínimas que os mesmos deverão cumprir, para funcionamento no âmbito da concessão e na relação com o Concedente. As outras funcionalidades ou características dos subsistemas serão de acordo com os interesses do Concessionário desde que permitam respeitar as boas práticas de gestão e operação de sistemas deste tipo.

O concessionário deverá, pois, incluir na sua proposta, um sistema de bilhética e subsistemas associados conforme com estes requisitos mínimos, assim como todas as atividades associadas à exploração e manutenção dos mesmos para a duração do contrato de concessão, de acordo com o faseamento que adiante se explicitará.



2 enquadramento

A SRTOP – Secretaria Regional de Transportes e Obras Públicas, tem em mente vir a implementar um Sistema Integrado, Intermodal e Multisserviços (SiIM), que venha a incluir a totalidade dos transportes públicos de passageiros, cobrindo todo o território da Região Autónoma dos Açores.

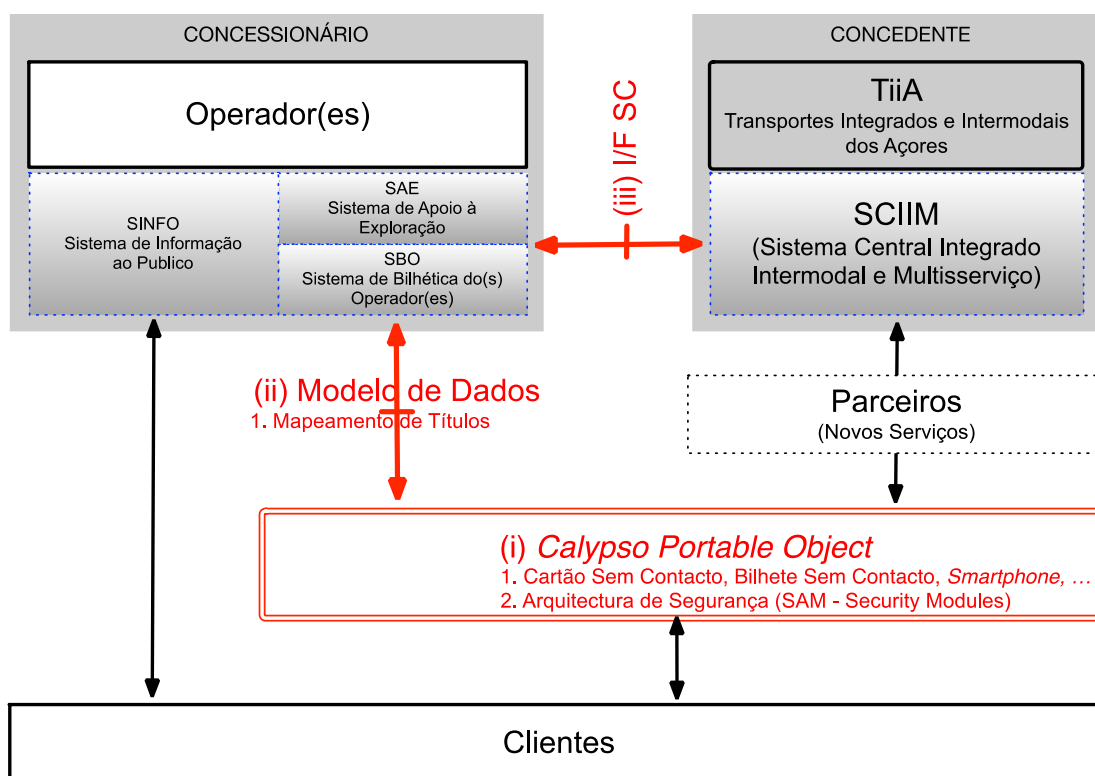


Figura 1 – Arquitectura Institucional e Elementos de Interoperabilidade

Este desígnio visa:

- A criação de um verdadeiro conceito de serviços de mobilidade, generalizável a todos os operadores e à região;
- Um Cartão “único” para toda a RAA, com especificação (modelo de dados) e imagem comum, para suporte dos títulos de transporte, sejam próprios, combinados ou intermodais;
- A promoção de um “Passe Turístico” válido em toda a região;
- A promoção de títulos combinados e intermodais;



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

- A promoção de Rede comum de venda/carregamento;
- Um Portal comum de informação ao público de apoio à mobilidade;
- E, no futuro, um sistema desmaterializado de Bilhética Móvel (*Mobile Ticketing*) e a desenvolvimento de novos serviços de mobilidade (para além do transporte de passageiros).

Tal deverá levar à criação de um contexto institucional que permita o desenvolvimento de uma oferta intermodal, sem prejuízo da independência na operação com títulos próprios de cada um dos operadores envolvidos, cuja arquitectura se documenta na Figura 1:

- Concedente - Governo Regional dos Açores (GRA), através da SRTOP;
- Concessionários - Operadores de Transporte Público de passageiros;
- *Calypso Portable Object* (PO) – a adoção de uma tecnologia não proprietária para o suporte aos títulos de transporte (Cartão sem contacto, Bilhete sem contacto, smartphone, ...) e que se configura como a interface entre os clientes e o sistema de transporte público e de mobilidade em geral;
- Subsistemas - de Bilhética (SBO), Apoio à exploração (SAE) e de Informação ao Público (SINFO) de cada um dos operadores, com os requisitos necessários a garantir a compatibilidade e interoperabilidade;
- Entidade de Gestão - TiiA (Transportes Integrados e Intermodais dos Açores), por agora assim designada, responsável pelo SCiiM (Sistema Central Integrado Intermodal e Multisserviço), atuando como aglutinadora e “guardiã” dos PO (*Calypso Portable Object*) e da intermodalidade, bem como responsável por assumir as funções transversais a todo o sistema;
- Clientes, utilizadores dos transportes públicos de passageiros da RAA.

Caberá ao Concedente definir quando, como e em que formato será criada este contexto institucional e operacionalizada a TiiA (e respetivo SCiiM), bem como o respetivo Regulamento de Funcionamento.

Este contexto institucional deverá vir a ser conseguido através da adoção, por todos os operadores, dos mesmos requisitos base nos seus subsistemas de SAE e SINFO, e da mesma tecnologia de suporte à bilhética no subsistema SBO, cumprindo os Elementos de



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Interoperabilidade ((i) Calypso Portable Objects, (ii) Modelo e Dados e (iii) I/F SC - Interface com o Sistema Central) evidenciados na Figura 1.

Esta adoção deverá, contudo, ocorrer de forma gradual e faseada, acompanhando os prazos necessários à tomada de decisão, por parte do concedente, respeitantes à definição da futura arquitectura institucional e correspondentes trabalhos de preparação da infraestrutura central.



3 faseamento

Conforme atrás se referiu, os subsistemas (SBO, SAE e SINFO) poderão evoluir, do ponto de vista tecnológico e funcional, de forma gradual. A Tabela 1 documenta o Faseamento previsto para essa evolução, durante o período da concessão.

	- Início da Concessão	- Início da Concessão+4 anos	- Início da Concessão+?
Subsistema	Fase 0 - Proprietário	Fase 1 - Interoperável	Fase 2 - Intermodal
Bilhética (SBO)	<ul style="list-style-type: none"> ▪ Sistema de bilhética eletrónica proprietário (tecnologia fechada) [Secção 3.1]. ▪ Arquitectura/Requisitos de acordo com os Níveis 0-3 do IFMS [Secção 3.1.2] ▪ Geração automática ou Edição de Relatórios (Auditáveis), a partir dos dados recolhidos pelo sistema de bilhética, para envio ao concedente [Secção 3.1.3]. ▪ Integração com a AT (Autoridade Tributária) [Secção 3.1.2.2]. ▪ Garantia de cumprimento do RGPD [Secção 3.1.2.3]. 	<ul style="list-style-type: none"> ▪ Sistema de Bilhética eletrónica preparado para ser interoperável (tecnologia aberta, não proprietária) [Secção 3.2]. ▪ Arquitectura/Requisitos de acordo com os Níveis 0-3 do IFMS [Secção 3.2.2] ▪ Tratamento de títulos Calypso [Secção 3.2.3] ▪ Adoção do Modelo de Dados definido pelo concedente [Secção 3.2.3.3] ▪ Definição de layout próprio, a acertar com o Concedente, para os novos cartões sem contacto ▪ Mapeamento dos títulos de transporte no Modelo de dados [Secção 3.2.3.3.1] ▪ Aquisição de cartões sem contacto com o layout definido, nas quantidades necessárias à operação da concessão para suporte, no mínimo, de todos os passes sociais [Secção 3.2.3.2]. ▪ Aquisição de módulos de segurança (SAM), nas quantidades necessárias à operação da concessão [Secção 3.2.3.1] ▪ Geração automática de relatórios dos dados recolhidos pelo sistema de bilhética, de acordo com o formato e procedimento a definir pelo concedente [Secção 3.2.3]. 	<ul style="list-style-type: none"> ▪ Sistema de Bilhética Interoperável (Intermodal) [Secção 3.3] ▪ Arquitectura/Requisitos de acordo com os Níveis 0-4 do IFMS [Secção 3.3.2], cumprindo na integra o actual Anexo Técnico
Apoio à Exploração (SAE)	<ul style="list-style-type: none"> ▪ Geração automática ou Edição de Relatórios (Auditáveis), referentes à oferta, horários e atrasos [Secção 3.1.4] 	<ul style="list-style-type: none"> ▪ SAE mínimo, com subsistema central (BackOffice) e subsistema embarcado, para suporte à bilhética e 	<ul style="list-style-type: none"> ▪ SAE [Secção 3.3.4]



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

		<p>monitorização de serviços [Secção 3.2.4].</p> <ul style="list-style-type: none"> ▪ Geração automática de Relatórios de monitorização dos serviços prestados, horários e atrasos, de acordo com o formato e procedimento a definir pelo concedente. ▪ Disponibilização de um acesso remoto, de forma a permitir ao Concedente realizar consultas e produção de relatórios nas ocasiões e na forma que entender. 	
Informação ao Público (SINFO)	<ul style="list-style-type: none"> ▪ Portal com informação básica sobre, no mínimo, o concessionário (incluindo rede de vendas), a oferta, os horários e as tarifas [Secção 3.1.5]. 	<ul style="list-style-type: none"> ▪ Alimentadores (<i>feeds</i>) GTFS para as mais conhecidas plataformas de âmbito global que tratam informação de transportes, como seja o Google Maps e iMaps (Apple) [Secção 3.2.5] 	<ul style="list-style-type: none"> ▪ INFO [Secção 3.3.5]

Tabela 1 – Faseamento

3.1 **FASE 0 - Proprietário**

A FASE 0 foi pensada de forma a permitir que o Concessionário possa utilizar os subsistemas SBO, SAE e SINFO de acordo com a sua opção/preferência, de modo a minimizar a complexidade inerente ao arranque da Concessão.

Em relação a subsistemas comuns do mercado, a FASE 0 não deverá obrigar a alterações significativas, sendo a mais relevante a obrigatoriedade de produção das estatísticas relativas ao controlo da prestação do Serviço Público para envio ao Concedente.

As restantes fases, a implementar segundo calendário a definir pelo Concedente, já pressupõem requisitos mais específicos de garantia de interoperabilidade em sintonia com a política pretendida para a mobilidade na RAA. Estes requisitos estão em linha com as boas praticas adotadas na construção de sistemas intermodais, comprovadas a nível Nacional (Lisboa, Porto, Coimbra, Funchal, ...) e Internacional.

3.1.1 **Contexto Institucional**

A Figura 2, cuja legenda se encontra no **ANEXO B. “Entidades e Responsabilidades”**, documenta o contexto institucional para a Fase 0.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

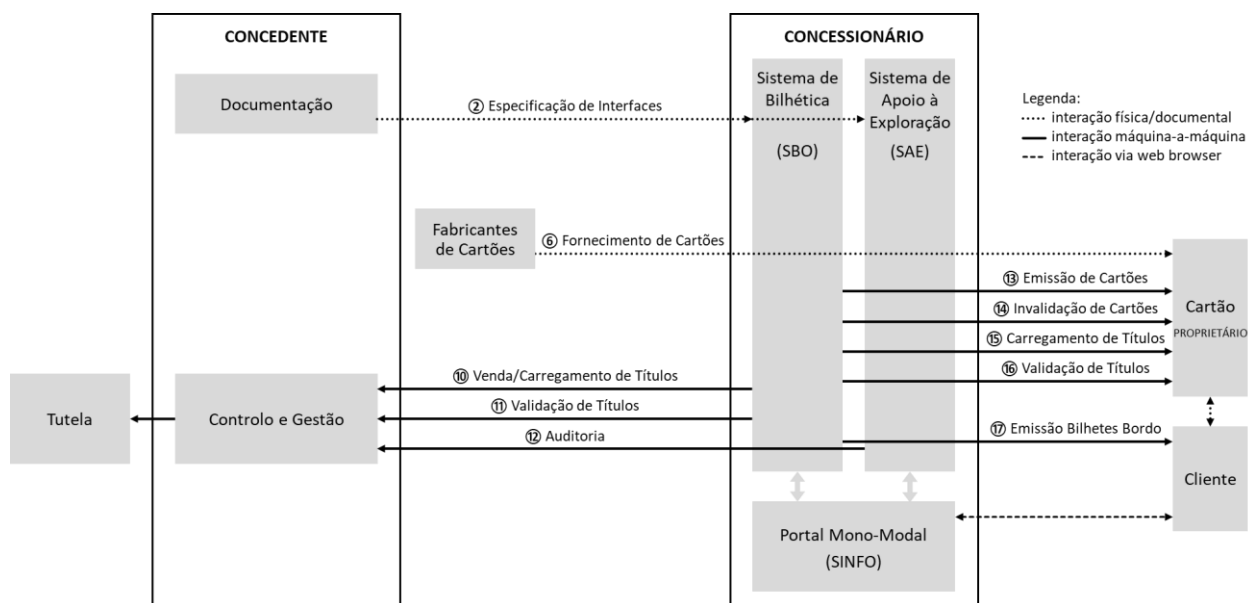


Figura 2 – FASE 0: Diagrama de Contexto Institucional

3.1.2 Arquitectura de referência para o SBO

Com base na estrutura definida abaixo, habitual na conceção de sistemas interoperáveis, resumem-se os principais requisitos a suportar pelo SBO durante a Fase 0 da concessão.

Nível IFM	Item	Requisitos
0	Media	<ul style="list-style-type: none">Bilhete em Papel (Térmico)Cartão sem contacto de tecnologia ISO 14443/A (1,2) – Mifare – ou ISO 14443/B (1,2,3,4) - Calypso
1	Terminais	<ul style="list-style-type: none">Consola de Venda a Bordo para venda e impressão de bilhetes em Papel (bilhetes de bordo) e validação de cartões sem contacto.Posto de Personalização de cartões sem contacto.Ponto de Venda para carregamento de cartões sem contacto.Terminal para Fiscalização de cartões sem contacto.
2	Sistema de Recolha de Dados e Supervisão	<ul style="list-style-type: none">Parametrização e Recolha de Dados manual, módulos de recolha de dados, ou automática, baseada em comunicações móveis (3G/4G) e WiFi.
3	Sistema Central de Operador	<ul style="list-style-type: none">Relatórios de procura para envio ao Concedente.

Tabela 2 – Fase 0: Níveis IFMS (*Interoperable Fare Management System*)

3.1.2.1 Títulos de transporte versus tecnologia de suporte

O SBO do concessionário deverá estar preparado para tratar toda a tipologia de perfis de cliente, de títulos de transporte e tarifários definidos para a concessão, tal como explicitado no ANEXO II ao Caderno de Encargos e reproduzido na Tabela 3.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Modo	Tipo de Título	Suporte (PO)
Próprios	Bilhete Simples de Bordo (BS)	Papel (emissão a bordo)
	½ Bilhete de Bordo (MB)	Papel (emissão a bordo)
	Pré-Comprados 10 viagens 3ª idade e pensionista por invalidez (PC 10V 3IP)	Cartão sem contacto
	Pré-Comprados 10 viagens (PC 10V)	Cartão sem contacto ou Papel
	Passes Rede Geral 1 dia (P1 Dia)	Cartão sem contacto
	Passes Rede Geral 3 dia (P3 Dia)	Cartão sem contacto
	Passes Rede Geral 5 dia (P5 Dia)	Cartão sem contacto
	Passe 30 Dias (P30 Dias)	Cartão sem contacto
	Passe Desempregado (P DES)	Cartão sem contacto
	Passe 3ª idade, pensionista e invalidez (P 3IP)	Cartão sem contacto
	Passe Escolar (P EST)	Cartão sem contacto

Tabela 3 – FASE 0/1: Tipologia de Títulos Transporte versus Tecnologia

Todos os títulos de transporte deverão, obrigatoriamente, ter emissão eletrónica, estando reservado a possibilidade de venda de títulos “manuais” apenas para situações de exceção (modo degradado) de avaria nos equipamentos de bordo.

Todos os títulos de transporte do tipo Passe deverão, obrigatoriamente, ser suportados no Cartão sem contacto.

3.1.2.2 Integração com a AT (Autoridade Tributária)

O SBO deverá assegurar o cumprimento do requisito legal de possuir um sistema de faturação certificado pela AT (Autoridade Tributaria), com a correspondente emissão de ficheiros SAFT-PT. A exata configuração deste mecanismo é opção do Concessionário. Este requisito é exigível seja qual for a fase em causa.

3.1.2.3 Regulamento Geral sobre a Proteção de Dados

É da responsabilidade do Concessionário propor e implementar todas as medidas necessárias ao pleno cumprimento do RGPD - Regulamento Geral sobre a Proteção de Dados (Regulamento UE 2016/679), mediante proposta prévia e respetiva aprovação por parte do Concedente. Este requisito é exigível seja qual for a fase em causa.

3.1.3 Elementos de Interoperabilidade

Os elementos de interoperabilidade que o SBO do concessionário deverá garantir na Fase 0 resumem-se a:

- iii. Interface de dados com o Sistema Central (I/F SC)



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Produção de estatísticas relativas ao controlo da prestação do Serviço Público previstas no CE, para envio ao concedente, designadamente:

- Geração automática ou Edição de Relatórios (Auditáveis) respeitantes à procura, a partir dos dados recolhidos pelo sistema de bilhética.

O formato exato dos relatórios e modo de envio deverão ser objeto de proposta do Concessionário e merecer o acordo do Concedente.

3.1.4 SAE

Na Fase 0 Concessionário poderá não dispor de um SAE propriamente dito, desde que tenha capacidade de produzir as estatísticas relativas ao controlo da prestação do Serviço Público previstas no CE, para envio ao concedente, designadamente:

- Geração automática ou Edição de Relatórios (Auditáveis), referentes à oferta executada, contendo, no mínimo, informação respeitante aos Serviços Programados/Desvios de Oferta.

O formato exato dos relatórios e modo de envio deverão ser objeto de proposta do Concessionário e merecer o acordo do Concedente.

3.1.5 SINFO

O concessionário obriga-se a dispor, no arranque da operação, do seu próprio sistema de informação ao público constituído, no mínimo, por um portal com conteúdo e funcionalidades básicas, tal como a seguir se exemplifica (lista não exaustiva):

1. Informação Geral:

- a) Informação institucional, Contactos e Horário das Estruturas de Apoio ao Cliente;
- b) Locais de aquisição e regras comuns de utilização dos títulos de transporte;
- c) Tipologia de títulos de transporte e tarifários;
- d) Mapas de Rede / Linhas / Carreiras / Horários;
- e) Páginas específicas, direccionadas a clientes estrangeiros (pelo menos em duas diferentes línguas internacionais incluindo o inglês);

2. Ligações



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

- a) Ligações (Links) para outros sites, designadamente dos restantes operadores de transporte publico de passageiros ou outras entidades relacionadas com a mobilidade.

3.2 FASE 1 - Interoperável

A FASE 1 obrigará à evolução tecnológica do SBO de modo a incluir tecnologia aberta para suporte dos títulos de transporte, de forma a vir a permitir a interoperabilidade entre diversos operadores de transporte, quando tal cenário vier a ser colocado pelo Concedente.

3.2.1 Contexto Institucional

A Figura 3, cuja legenda se encontra no **ANEXO B. “Entidades e Responsabilidades”**, documenta o contexto institucional para a Fase 1.

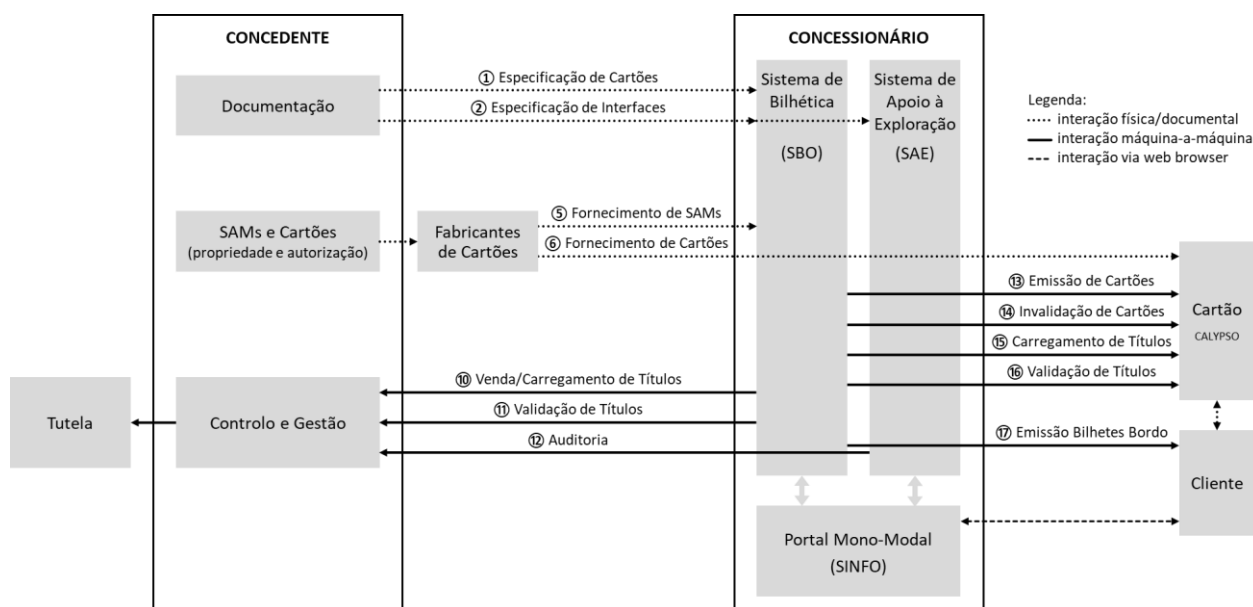


Figura 3 – FASE 1: Diagrama de Contexto Institucional

3.2.2 Arquitectura de referência para o SBO

Com base na estrutura definida abaixo, habitual na conceção de sistemas interoperáveis, resumem-se os principais requisitos a suportar pelo SOB durante a Fase 1 da concessão.

Nível IFM	Item	Requisitos
0	Media	<ul style="list-style-type: none">Bilhete em Papel (Térmico)



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Nível IFM	Item	Requisitos
		<ul style="list-style-type: none">• Cartões sem contacto de tecnologia ISO 14443/B (1,2,3,4) compatível com a arquitetura de segurança baseada em SAM (Security Modules) Calypso (ISO 1545/IOPTA – estrutura de comandos CSC), como o CD97BX ou CDLight. (A decisão final poderá apontar para um cartão com configuração multiserviços, como o CD21-Rev2)• Possibilidade de vir a utilizar bilhetes sem contacto do tipo ISO 14443/B (1,2) ou Mifare Ultralight. Na eventualidade de se decidir utilizar este tipo de bilhetes, a escolha deverá recair na família C-Ticket (ASK), SRIX (STM), Mifare Ultralight Ev 1
1	Terminais	<ul style="list-style-type: none">• Consola de Venda a Bordo para venda e impressão de bilhetes em Papel (bilhetes de bordo) e validação de cartões sem contacto.• Posto de Personalização de cartões sem contacto.• Ponto de Venda para carregamento de cartões sem contacto.• Terminal para Fiscalização de cartões sem contacto.
2	Sistema de Recolha de Dados e Supervisão	<ul style="list-style-type: none">• (Opcional) Parametrização e Recolha de Dados automática, baseada em comunicações móveis (3G/4G) e/ou WiFi.
3	Sistema Central de Operador	<ul style="list-style-type: none">• Relatórios de procura para envio ao Concedente.

Tabela 4 – Fase 1: Níveis IFMS (*Interoperable Fare Management System*)

3.2.2.1 Títulos de transporte versus tecnologia de suporte

No que respeita ao suporte de títulos de transporte, aplica-se o que foi referido para a FASE 0 (secção 3.1.2.1) com a diferença que, na FASE 1 o Cartão sem contacto será o novo Cartão Calypso.

3.2.3 Elementos de Interoperabilidade

Os elementos de interoperabilidade que o SBO do concessionário deverá garantir na Fase 1 serão:

i. *PO / Calypso Portable Object*

Adoção da tecnologia Calypso (CNA) para suporte dos títulos de transporte.

1. Cartão sem contacto ISO 14443/B (1,2,3,4) - Calypso;
2. Arquitectura de Segurança baseada em SAM (*Secure Application Module*).

ii. Modelo de Dados;

Adoção de um modelo de dados não proprietário.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

1. Modelo de Dados, ou seja, definição da codificação, posição e regras de manipulação dos dados existentes no Cartão sem contacto a definir pelo Concedente;
 2. Mapeamento dos diversos Títulos de Transporte no Modelo de Dados a propor pelo Concessionário e sujeito à aprovação do Concedente.
- iii. Interface de dados (I/F SC) entre o SBO e o SCIIIM.

O Erro! A origem da referência não foi encontrada. detalha em pormenor o mecanismo a utilizar para a comunicação automática entre o Operador e o Concedente, bem como o formato da informação a ser veiculada.

Trata-se de uma interface de dados genérica, desenvolvida para ambientes intermodais. Esta especificação virá a ser afinada durante o decurso da FASE 0, em colaboração com o Concessionário. No entanto considera-se que este documento é suficiente para avaliar a complexidade e o esforço a envolver pelo Concessionário na respetiva implementação.

Por se tratar de um mecanismo de transferência de ficheiros, não é exigível uma comunicação em tempo-real de alto débito mas apenas uma ligação segura SFTP.

3.2.3.1 Calypso Portable Object: Arquitectura de Segurança

O sistema de bilhética deverá basear-se na arquitectura de segurança Calypso, suportada em SAM (*Secure Application Modules*), de modo a garantir uma transação segura entre os *Calypso Portable Objects* - PO (cartões, etc.) e os terminais (equipamentos) que os manipulam (leitura/escrita).

Estes módulos de segurança, SAM, que manipulam um conjunto de chaves e algoritmos criptográficos (para autenticar o PO, o terminal e toda a troca de informação entre ambos), permitem uma segurança acrescida nas operações com os cartões e títulos de transporte, no sentido de permitir criar um sistema intermodal, multioperador, com repartição dinâmica de receitas, robusto à fraude e confiável.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

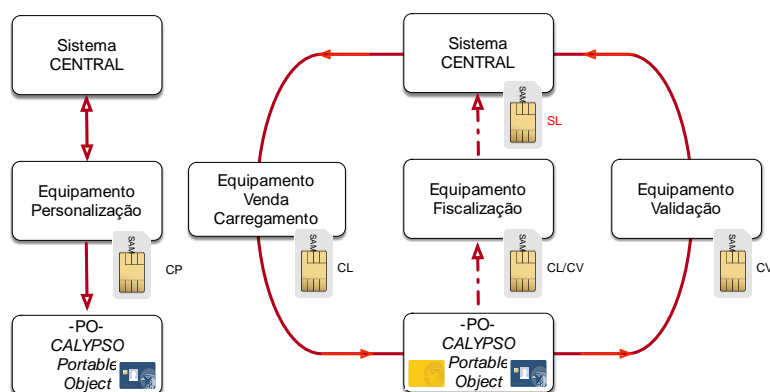


Figura 4 – Segurança num Sistema de Bilhética Calypso

O concessionário deverá garantir que no seu SBO, todos os equipamentos que manipulam (leitura e/ou escrita) os PO (cartão/bilhete sem contacto, etc.) devem estar equipados com um dispositivo de cartões (*contactless card reader*) que disponha de uma interface física (*SAM-slot*) para pelo menos dois Módulos de Segurança, SAM.

O Concedente prevê a utilização de SAM do tipo SAM-S1 E1, desenhados e fornecidos pela empresa Spirtech (<http://www.innovatron.fr/CalypsoProducts-Cards.pdf>), com chaves de encriptação DESX e 3DES.

Em relação aos SAM's, e de acordo com as boas práticas de segurança, estão previstas as seguintes tipologias, típicas dos sistemas Calypso, designadamente:

1. **SAM-SP:** *Personalização de SAM: Não Aplicável (da responsabilidade exclusiva do Concedente).*
2. **SAM-CPP:** *Pré-Personalização e Carregamento de Chaves nos Cartões: Não Aplicável (da responsabilidade exclusiva do Concedente).*
3. **SAM-SL:** *Gestão do limite de transacções (ceiling) dos SAM: Não Aplicável (não previsto nesta implementação).*
4. **SAM-CP:** Personalização de Cartões (Issuer): Necessário em todos os equipamentos de personalização física/lógica do cartão.
5. **SAM-CV:** Validação (Validation): Necessário em todos os equipamentos de validação e fiscalização.
6. **SAM-CL:** Carregamento (Reload): Necessário em todos os equipamentos de venda/carregamento, como sejam bilheteiras e consolas integradas (venda e validação) de bordo.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Por razões de segurança caberá ao Concedente a contratação de todo o processo de produção de chaves de autenticação e prestação da autorização de fabrico dos cartões (carregamento das chaves) e módulos de segurança (SAM).

Caberá ao Concessionário, devidamente autorizado pelo concedente, a aquisição de SAM na quantidade adequada à respetiva Concessão, ao seu SBO e respetivo modelo de operação. Sem prejuízo de melhor definição, o **ANEXO A. “Cartões, Bilhetes e SAM: tipologias e preços de referência”** contém preços indicativos para referência do Concessionário.

3.2.3.2 Calypso Portable Object: Cartão Calypso

O sistema de bilhética do Concessionário deve ser desenhado de acordo com a tecnologia Calypso e recomendações da CNA (*Calypso Network Association* - <https://www.calypsonet-asso.org>).

Os títulos sazonais (passes, assinaturas) ou multisserviços serão suportados em cartões de microprocessador, duais, com interface com contacto ISO 7816 (1,2,3) e interface sem contacto ISO 14443B (1,2,3,4), com mapeamento de ficheiros de acordo com a norma ISO 7816 (4), estrutura de dados de acordo com a norma EN 1545 e compatível com a arquitetura de segurança baseada em SAM (*Secure Application Modules*).

Trata-se de cartões de plástico (ABS), personalizáveis elétrica e graficamente (através de impressora própria para impressão de cartões).

O Concedente prevê a utilização de um cartão com configuração multisserviços, CD21 e CD21 Rev2 (STMicroelectronics) que estão de acordo com as normas:

- CalypsoRev1
CalypsoRev2
- ISO14443B (1 a 4)
- ISO 7816 (1 a 4) T=0
- ENV1545
- Security access module (SAM S1)

No entanto o concessionário deve garantir compatibilidade, quer com o referido cartão, quer com qualquer outro cartão Calypso (<https://www.calypsonet-asso.org/content/calypso-certified-products> e <http://www.innovatron.fr/CalypsoProducts-Cards.pdf>), que possa vir a ser



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

utilizado no futuro e que respeite as mesmas normas e estruturas de dados, numa situação de eventual coexistência de mais do que um tipo de cartão.

Caberá ao Concessionário a definição do *layout* gráfico do cartão, o qual deve merecer o acordo do Concedente.

Caberá ao Concessionário, devidamente autorizado pelo Concedente, a aquisição de cartões na quantidade adequada à respetiva Concessão, ao seu SBO e respetivo modelo de operação. Sem prejuízo de melhor definição, o **ANEXO A. “Cartões, Bilhetes e SAM: tipologias e preços de referência”** contém preços indicativos para referência do concessionário.

3.2.3.3 Modelo de Dados

Entende-se como Modelo de Dados do cartão:

- a) A forma (codificação) como a informação dos diferentes “contratos (títulos) de transporte” de todos os tarifários envolvidos, está arrumada/mapeada (posição) na “memória” do cartão;
- b) O conjunto de “regras de negócio” básicas de acesso a essa informação bem como um conjunto de dados de referência (a que chamaremos de “regras do modelo de dados”).

Ao contrário do que acontecia no passado em que os modelos de dados dos cartões afetos aos sistemas de bilhética eram proprietários, desconhecidos dos operadores de transporte e completamente dependentes do integrador, nos modernos sistemas integrados multioperador e intermodais, os modelos de dados são públicos, controlados pelos operadores ou entidades afetas à operação dos transportes e independentes do integrador.

A especificação do Modelo de Dados a utilizar será disponibilizada pelo Concedente, e será obrigatória a sua utilização por parte do Concessionário.

A escolha do Concedente recaiu no “Andante”, propriedade dos TIP (Transportes Intermodais do Porto), modelo este em utilização quer na Área Metropolitana do Porto quer nos HF (Horários do Funchal) e futuramente nos restantes transportes públicos da Madeira.

A especificação deste Modelo de Dados é apresentada no **Erro! Autorreferência de m arcador inválida.**, na sua versão actual. É natural que esta especificação venha a sofrer



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

algumas alterações. No entanto considera-se que este documento é suficiente para avaliar a complexidade e o esforço a envolver pelo Concessionário na respetiva implementação.

3.2.3.3.1 Modelo de Dados: Mapeamento dos Títulos de Transporte

Será tarefa do Concessionário apresentar uma proposta de mapeamento no modelo de dados, dos tarifários, perfis de clientes, tipos de títulos, validades temporais e espaciais e restante informação relevante, em relação à rede afeta à respetiva concessão.

Este mapeamento deverá ser objeto de aprovação por parte do Concedente, antes de entrar em funcionamento.

O concessionário deverá produzir e fornecer o respetivo documento de referência que ficará propriedade do Concedente. A utilização rigorosa deste mapeamento, no decurso de toda a concessão, será mandatória, devendo ser mantido atualizado, não podendo ser utilizado na implementação e operação do SBO nenhum detalhe que não tenha sido previamente documentado e aprovado.

3.2.3.3.2 Modelo de Dados: Interface de Programação (API)

Entende-se por API (*Application Programming Interface*) o modulo de software que faz a interface entre as aplicações de bilhética e a tecnologia (cartões sem contacto, leitores/cartões, módulos de segurança, Sistema Operativo), visando garantir:

- a) o acesso disciplinado e coerente aos dispositivos de leitura e cartões, de acordo com o “modelo de dados” e “regras do modelo de dados” definidas;
- b) o tratamento rigoroso dos aspetos de segurança, seguindo o “modelo de segurança” definido;
- c) o parametrização e tratamento uniforme de um conjunto essencial de “regras de negócio”;
- d) modularidade e independência em relação aos cartões, leitores, segurança e sistema operativo dos terminais.

A arquitetura do software do SBO deverá, de acordo com as boas práticas, ser modular, e em particular o módulo de acesso ao cartão deverá ser autónomo e fornecer uma API para os restantes módulos. A função principal deste módulo é assegurar um funcionamento homogéneo de toda a bilhética.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

O Concessionário obriga-se a integrar uma API no software de todos os equipamentos a fornecer que interajam com os PO, com as seguintes características mínimas:

- A API deverá disponibilizar um interface único para a aplicação, independente da marca ou modelo de leitor/gravador de cartões, do tipo de PO utilizado, do módulo de segurança (SAM – *Secure Application Module*) instalado ou do sistema operativo do equipamento.
- A substituição de um leitor/gravador de cartões num equipamento, a instalação de um novo módulo de segurança ou o suporte a um novo tipo de cartão no sistema de bilhética, deverão implicar exclusivamente uma atualização da API, sem qualquer modificação da aplicação (aceita-se que sejam eventualmente necessárias atualizações de baixo nível, não-aplicacionais, tais como *device drivers* ou actualização de algum *firmware*).
- Relativamente aos PO, a API deverá virtualizar o modelo de dados, exibindo para a aplicação a mesma estrutura de dados, independentemente da instanciação física da informação no PO. *(A integração, improvável, de um novo modelo de dados na bilhética poderá implicar modificações na estrutura de dados virtual do cartão, sendo neste caso admissível a necessidade de alterar a aplicação.)*
- O interface a disponibilizar pela API para acesso ao leitor de cartões e ao módulo de segurança, deverá ser o que o concessionário considerar necessário para um correto funcionamento da aplicação. O Concessionário deverá garantir, no entanto, que qualquer operação realizada com estes dispositivos será feita unicamente através da API.
- A API deverá oferecer à aplicação as funções básicas da bilhética: personalizar, carregar, validar, fiscalizar e ler/consultar um cartão. As funções que modificarem o conteúdo do cartão, deverão retornar um registo de transação adicionado de uma assinatura que garanta a integridade dos dados. A assinatura das transações deverá ser verificável através de uma outra função da própria API.
- As transações retornadas pela API terão que conter obrigatoriamente um número de sequência que permita verificar, a posteriori, faltas ou repetições de transações. A verificação da sequencialidade das transações será também disponibilizada pela API. O número de sequência deve ser obtido do próprio módulo de segurança (SAM) instalado no equipamento.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

- A API deverá ser obrigatoriamente *thread safe* para permitir uma utilização concorrente nos equipamentos com um sistema operativo que o permita, de forma a evitar bloqueios desnecessários nas aplicações, perdas de desempenho ou até mesmo corrupção dos dados. (Nota: Um programa/código é considerado *thread safe* se manipula estruturas de dados críticas através dos adequados mecanismos de sincronização, de forma a garantir uma execução concorrente e segura de várias *threads*.)

O Concessionário obriga-se a fornecer a especificação técnica detalhada da API que vier a integrar no seu sistema de bilhética (SBO).

Alternativamente o Concedente pode, se assim o entender, vir a fornecer a documentação, código e apoio à integração de uma API própria, num cenário de minimizar o risco e tornar mais eficiente e rápida a compatibilização entre os diferentes sistemas de bilhética da RAA. Neste caso será mandatória a utilização da API fornecida pelo Concedente.

A especificação da API que poderá vir a ser disponibilizada pelo Concedente, é apresentada no **Erro! A origem da referência não foi encontrada.** na sua versão actual. Esta especificação pode vir a sofrer alguma alteração. No entanto, considera-se que este documento é suficiente para avaliar a complexidade e o esforço a envolver pelo Concessionário na respetiva integração.

3.2.4 **SAE**

Na Fase 1 o Concessionário deverá dispor de um SAE mínimo, com componente de localização embarcada (poderá estar incluída no equipamento de bilhética) cobrindo toda a frota de veículos (próprios e subcontratados) utilizada na exploração do Serviço Público, dotado de uma central de comando de tráfego que acompanhe o estado dos serviços prestados em tempo real, e que permita introduzir as adaptações que se revelarem necessárias face a novas exigências de operação.

O SAE deve disponibilizar as funcionalidades básicas normais em sistemas desta natureza, e deverá normalmente ser constituído pelos seguintes módulos funcionais:

- 1) **Equipamento Embarcado**, cuja principal função é a recolha, em tempo real, da informação referente à operação, designadamente a relacionada com a localização geográfica dos veículos e demais informações a eles associados, como as distâncias percorridas, as velocidades, abertura de portas, eventos, alarmes;



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

2) **Central de Comando de Tráfego**, englobando as funcionalidades de apoio à exploração propriamente dita, normalmente suportada num sistema de informação geográfica (GIS), permitindo:

a) Configuração/Parametrização/Gestão da topologia da rede, caracterização das entidades constituintes da rede de transportes, nomeadamente Paragens, Linhas/Carreiras, Troços e Variantes:

i) **Paragens**, através da caracterização dos seus atributos (tipo de instalação, etc) e das respetivas localizações geográficas (lista georreferenciada), distrito, concelho e freguesia. Deve ser ainda possível:

(1) Definir paragens “satélite” (ou outro tipo de paragens “opcionais”) para situações de transporte a pedido/flexível;

(2) Associar informação relevante para a bilhética, como sejam as zonas tarifárias.

ii) **Carreiras/Linhas**, através da definição de espinhas (conjunto sequencial de paragens para cada sentido da linha) e para cada espinha/sentido, um conjunto de circulações.

b) Configuração/Parametrização/Gestão dos Recursos:

i) **Viaturas**, através da informação básica das viaturas (tipo de viatura, ano, matrícula, validade do seguro, data da inspeção, afetação), devendo ser possível configurar um conjunto de outras características, como sejam a existência ou não de tacógrafo e se é habilitada para transporte de crianças.

ii) **Motoristas**, através da informação básica do motorista (identificação, validade da carta de condução, vencimento, local de afetação e de refeição, dias de descanso, ausências/férias, escalas).

iii) **Calendário**

iv) **Horários**

c) Configuração/Parametrização/Gestão dos Serviços:

Serviços regulares, através da definição de percursos, horários e frequências;

(1) O conjunto das várias circulações define um horário.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

(2) Deverá ser possível configurar atrasos e adiantamentos para cada ponto de controlo definido (paragens da espinha);

(3) Deverá ser permitido adicionar elementos referentes a circulações a-pedido aos horários e circulações do serviço regular. Nestes casos os horários serão recalculados, cabendo ao operador a decisão de sobrepor, ou não, estas circulações a-pedido às circulações regulares para um determinado dia.

d) Planeamento da Operação:

- i) Alocação de recursos aos serviços (afetação de viaturas e Motoristas);
- ii) Para as funcionalidades de transporte a pedido/flexível o sistema deverá permitir também a alteração de rotas existentes ou geração de novas rotas.

e) Monitorização/Gestão Operação:

- i) Controlo global e em tempo real da localização das viaturas e respetivos dados de atividade (serviço em execução, motorista atribuído, velocidade comercial, etc);
- ii) Monitorização em tempo real e controlo de cumprimento de serviços e horários face aos serviços regulares ou ocasionais (por exemplo complementada por uma interface gráfica apelativa do tipo “termómetro” com código de cores);
- iii) Gestão de Eventos/Alarmes, tais como atrasos, avanços, excesso de velocidade (por exemplo através de uma interface gráfica apelativa do tipo “termómetro” com código de cores);
- iv) Integração, em tempo real, com o Sistema de Informação ao público.

3) Módulo de **Relatórios/Estatísticas**, permitindo a produção de um conjunto extensivo de relatórios de operação, comparando os registos previstos com reais, incluindo, apresentando dados específicos e agregados, tais como:

- Veículos.Km comerciais/vazio produzidos
- Veículos.Hora
- Velocidade Comercial por carreira
- Serviços programados/Desvios de Oferta



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

- Hora de partida de cada circulação
- Hora de chegada de cada circulação
- Atrasos de cada circulação

Deve ser possível produzir um conjunto de relatórios de alertas/exceções (parâmetros fora dos limites definidos).

Finalmente, deve ser possível produzir estatísticas relativas ao controlo da prestação do Serviço Público de forma automática, fiável, inviolável e verificável. O SAE deverá disponibilizar um acesso remoto permanente, de preferência através de um *front-end web*, de forma a permitir ao Concedente realizar consultas e produção de relatórios nas ocasiões e na forma que entender.

3.2.5 SINFO

Na Fase 1 o concessionário obriga-se a melhorar o seu sistema de informação ao público através de:

- Inclusão no portal de novos conteúdo e mecanismos mais interativos, tais como a seguir se exemplifica (lista não exaustiva):

3. Clientes e Cartões

a) Alteração de dados pessoais do cliente, disponibilizando mecanismo de (single-sign-on) *seguro, sendo que o registo inicial do cliente (1º registo) será sempre forçosamente feito num posto do Concessionário, para permitir a validação dos documentos de identificação;*

b) *Pedido de emissão de cartões (cartões adicionais, 2ª via):*

- *Recolha dos dados necessários ao cartão, no mínimo:*
 - *Nome a figurar no cartão;*
 - *Operador emissor e local de recolha.*
- *Emissão de Talão de recolha;*
- *Pagamento e respetiva emissão de Fatura/Recibo certificada pela AT;*



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

- *Consulta do estado do pedido.*
- c) Acesso aos dados pessoais do cliente, incluindo visualização do(s) cartão(ões) atribuído(s) e respetivo(s) estado(s). No caso da informação respeitante ao cartão, deverá ser possível visualizar:
 - Número de série do cartão;
 - Data de emissão do cartão;
 - Data de Validade do cartão;
 - Para cada um dos Contratos de Transporte 1 a 4:
 - Título carregado;
 - Data Expiração Título, para os títulos validade temporal;
 - Saldo existente (Quantidade/Valor), para os títulos onde se carregam unidades ou valores.
- d) *Mecanismo de notificação automática:*
 - *Cartão emitido e pronto a recolher;*
 - *Títulos (sazonais – validade temporal) prestes a expirar;*
 - *Validade do cartão prestes a expirar;*
- Disponibilização de alimentadores (feeds) para informação ao público e planeamento de viagens, no formato GTFS (General Transit Feed Specification), que serão utilizados tanto em plataformas generalistas (ex: Google Maps e iMaps), como para a disponibilização a terceiros, em regime de “open data”.

No que se refere aos *feeds* acima mencionados, distinguem-se dois tipos de informação, que diferem no seu conteúdo e forma de disponibilização: informação estática e informação em tempo real.

A **informação estática** deverá respeitar integralmente a especificação *GTFS static*, que se encontra disponível em developers.google.com/transit/gtfs. A disponibilização desta informação deve ser necessariamente efetuada antes da respetiva entrada em vigor, com um



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

prazo de antecedência mínimo de cinco dias úteis, salvo situações em que um prazo mais curto decorra de imposições alheias ao Concessionário.

A informação estática é constituída obrigatoriamente pela seguinte informação, mesmo quando considerada opcional na especificação GTFS:

1. feed_info.txt – informação sobre o *feed* – todos os campos obrigatórios;
2. agency.txt – informação sobre o operador ou conjunto de operadores – campos obrigatórios conforme especificação GTFS;
3. stops.txt – informação sobre as localizações e características das paragens – campos obrigatórios conforme especificação GTFS;
4. routes.txt – informação sobre as designações e características das carreiras – campos obrigatórios conforme especificação GTFS;
5. trips.txt – informação sobre os serviços (carreira + sentido) – campos obrigatórios conforme especificação GTFS;
6. stop_times.txt – informação sobre os horários e características das circulações (carreira + sentido + horário), para diversos calendários – campos obrigatórios conforme especificação GTFS;
7. calendar.txt – informação sobre a execução ou não das circulações segundo os dias da semana, para um dado período – campos obrigatórios conforme especificação GTFS;
8. fare_attributes.txt – informação sobre preços e transbordos da tarifa base (tarifa de referência a ser apresentada na informação ao público) – campos obrigatórios conforme especificação GTFS;
9. fare_rules.txt – informação sobre validades espaciais da tarifa base (tarifa de referência a ser apresentada na informação ao público) – campos obrigatórios conforme especificação GTFS;
10. shapes.txt – descrição geo-referenciada dos percursos dos serviços – campos obrigatórios conforme especificação GTFS.

Os restantes ficheiros previstos na especificação *GTFS static* são opcionais, mas todos eles serão devidamente considerados, quando presentes, pelos sistemas externos. Inclui-se no



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

âmbito destes ficheiros opcionais a extensão *translations.txt*, destinada ao suporte multi-idioma da informação a disponibilizar pelo concessionário (extensão à especificação *GTFS static*, descrita em developers.google.com/transit/gtfs/reference/gtfs-extensions).

A informação estática deve começar a ser disponibilizada pelo concessionário até três meses após notificação do Concedente.

3.3 FASE 2 - Intermodal

A FASE 2 constituirá um estágio mais avançado da interoperabilidade (ver desígnio do sistema na secção 2), através da criação de uma entidade de gestão intermodal, TiiA, de um cartão Único para toda a RAA e de títulos combinados entre operadores.

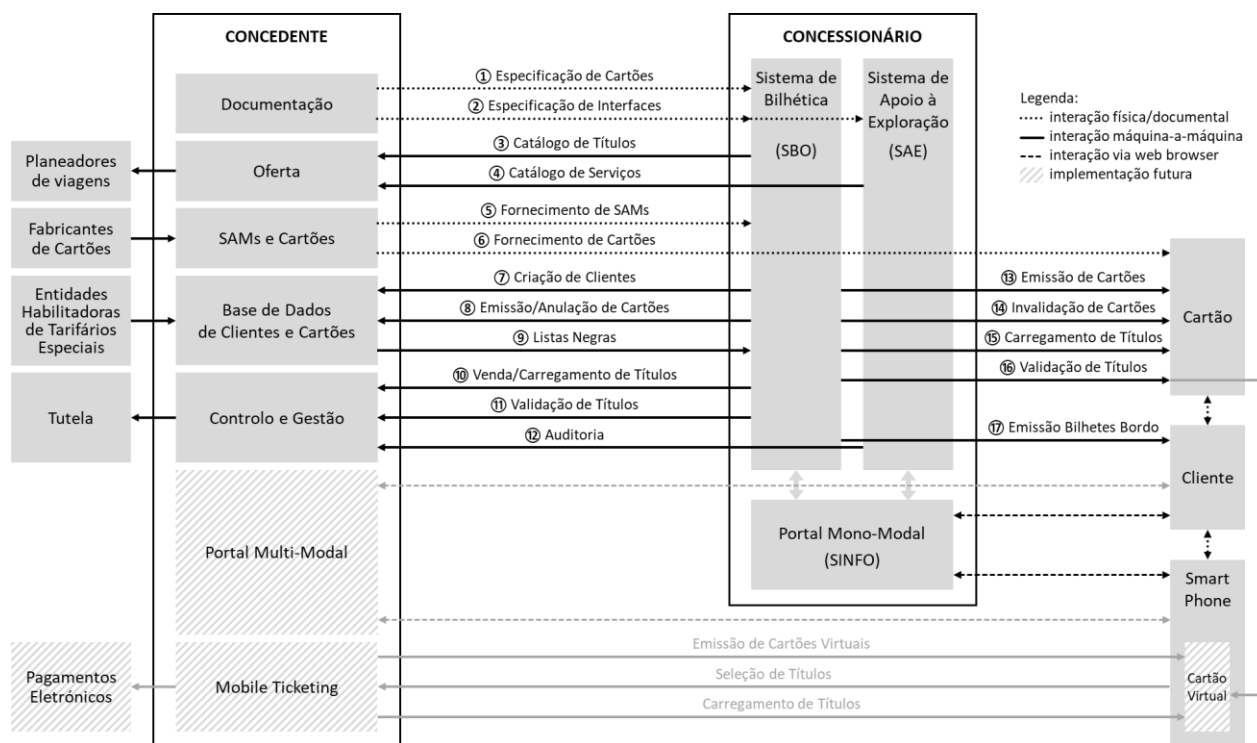


Figura 5 – FASE 2: Diagrama de Contexto Institucional

A FASE 2 poderá vir a adicionar a desmaterialização de títulos de transporte com base em smartphones ou mesmo a integração com outros serviços de mobilidade. A eventual disponibilização, operação e gestão deste sistema será da responsabilidade da TiiA, devendo o Concessionário garantir apenas a compatibilidade do seu SBO ao nível da validação dos PO.



3.3.1 Contexto Institucional

A Figura 5, cuja legenda se encontra no ANEXO B. “Entidades e Responsabilidades”, documenta o contexto institucional para a Fase 2.

3.3.2 Arquitectura de referência para o SBO

Com base na estrutura definida abaixo, habitual na conceção de sistemas interoperáveis, resumem-se os principais requisitos a suportar pelo SBO durante a Fase 2 da concessão.

Nível IFM	Item	Requisitos
0	Media	<ul style="list-style-type: none">Bilhete em Papel (Térmico)Cartões sem contacto de tecnologia ISO 14443/B (1,2,3,4) compatível com a arquitetura de segurança baseada em SAM (Security Modules) Calypso (ISO 1545/IOPTA – estrutura de comandos CSC), como o CD97BX ou CDLight. (A decisão final poderá apontar para um cartão com configuração multiserviços, como o CD21-Rev2)<i>Possibilidade de vir a utilizar bilhetes sem contacto do tipo ISO 14443/B (1,2) ou Mifare Ultralight. Na eventualidade de se decidir utilizar este tipo de bilhetes, a escolha deverá recair na família C-Ticket (ASK), SRIX (STM), Mifare Ultralight Ev 1</i><i>Possibilidade de vir a utilizar Smartphones com sistemas operativos Android (e IOS) para emulação (HCE) do cartão Calypso através de NFC.</i>
1	Terminais	<ul style="list-style-type: none">Consola de Venda a Bordo para venda e impressão de bilhetes em Papel (bilhetes de bordo) e validação de cartões sem contacto.Posto de Personalização de cartões sem contacto.Ponto de Venda para carregamento de cartões sem contacto.Terminal para Fiscalização de cartões sem contacto.
2	Sistema de Recolha de Dados e Supervisão	<ul style="list-style-type: none">(Opcional) Parametrização e Recolha de Dados automática, baseada em comunicações móveis (3G/4G) e/ou WiFi.
3	Sistema Central de Operador	<ul style="list-style-type: none">Relatórios de procura para envio ao Concedente.
4	Sistema Central Integrado Intermodal e Multiserviço (SCIIM)	<ul style="list-style-type: none">Emissão e Gestão de Cartões sem contacto.Gestão da Segurança e controlo antifraude.Repartição de Receitas para os títulos combinados e intermodais.Personalização e Carregamento de Cartões para a globalidade da rede de transportes (ex: Aeroporto, Lojas RIAC).Portal global de Informação ao público.<i>(Futuro) Bilhética Móvel (Mobile Ticketing).</i><i>(Futuro) Gestão de novos Serviços.</i>



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Tabela 5 – Fase 2: Níveis IFMS (*Interoperable Fare Management System*)

3.3.2.1 Títulos de transporte versus tecnologia de suporte

Na Fase 2 aplica-se o que foi referido para a FASE 1 (secção 3.2.2.1). Adicionalmente, e no que respeita aos títulos (Multioperador) combinados/intermodais que o Concedente venha a definir para a FASE 2, eles devem obrigatoriamente ser suportados no Cartão sem contacto.

Modo	Tipo de Título	Suporte (PO)
Multioperador	Combinados/Intermodais	Cartão sem contacto

Tabela 6 – FASE 2: Tipologia de Títulos Transporte versus Tecnologia

3.3.3 Elementos de Interoperabilidade

Deste ponto de vista, a Fase 2 é em tudo idêntica à Fase 1, sendo a principal alteração motivada pela mudança do contexto institucional, com a criação de uma entidade centralizadora, TiiA e a tomada para si de algumas funções (Nível 4 IFMS) transversais aos diversos operadores, obrigando à sofisticação da interface entre o SBO e o SCiiM, para, designadamente:

- Gestão Centralizada de Cartões e Clientes;
- Gestão de Lista Negra;
- Gestão Centralizada de Títulos intermodais/combinados e respetiva repartição de receita.

3.3.4 SAE

Deste ponto de vista, a Fase 2 é em tudo idêntica à Fase 1, não sendo necessária nenhuma alteração a este subsistema.

3.3.5 SINFO

Na Fase 2 o concessionário obriga-se a vir a alimentar o portal e APP que vier a ser criado pelo TiiA, pretendendo-se que, para além da informação estática, seja disponibilizada informação em tempo real.

Relativamente à **informação em tempo real**, a sua disponibilização deverá obedecer à especificação *GTFS Realtime*, que se encontra disponível em developers.google.com/transit/gtfs-realtime. Esta informação deverá ser disponibilizada pelo concessionário à TiiA, através da evocação de um *web service* disponibilizado pelo TiiA



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

para o efeito. A disponibilização desta informação deve ser efetuada com a antecedência adequada, de modo a poder ser utilizada em tempo útil pelos passageiros.

A informação em tempo real baseia-se nas definições efetuadas na informação estática e compreende a seguinte informação, considerando-se o regime de obrigatoriedade definido nas respetivas especificações:

1. *TripUpdate* – veículos atribuídos a circulações, supressão / desdobramento de circulações e avanços / atrasos em relação aos horários planeados;
2. *VehiclePosition* – posição atual dos veículos, geo-referenciada e em relação às paragens (a chegar, presente, em trânsito), e grau de congestionamento;
3. *Alert* – âmbito (operador / carreira / circulação / paragem), causas e efeitos de incidentes (quando esta informação é fornecida, deve sê-lo, no mínimo, em português e inglês);

A informação em tempo real deve começar a ser disponibilizada pelo concessionário até três meses após notificação da TiiA de que o respetivo *web service* de importação da informação se encontra disponível.



4 ANEXOS

ANEXO A. “Cartões, Bilhetes e SAM: tipologias e preços de referência”

- Cartões Calypso: (quantidade ~150K)

- Oberthur: 0.93€ (CD21)
- ASK: 0.97€ (TanGo)
- Gemalto: 1€ (CD21)

Para quantidades menores, 20K, o preço deverá rondar os 1.5€/1.7€.

- Bilhetes sem contacto: (quantidade ~1M)

- Mifare UL Ev1
 - Confidex: 0.0945€
 - Paragon: 0.101€
- Calypso CTS512
 - ASK: ~0,13€

Para quantidades menores, 50K, o preço deverá rondar os 0.15€/0.16€.

- SAM (fornecedor Spirtech – Preços tabelados pelo CNA)

- 36€ para quantidades < 10 SAM
- 30€ para quantidades >= 10 e < 50
- 28€ para quantidades >= 50 e < 250
- 26€ para quantidades >= 250

Preços válidos para qualquer das tipologias de SAM.

Estes valores são meramente indicativos apenas para referência do Concessionário e não têm qualquer carácter vinculativo.



ANEXO B. “Entidades e Responsabilidades”

DESCRIÇÃO DOS FLUXOS DE INTERAÇÃO COM OS SISTEMAS DO CONCESSIONÁRIO

Fluxo		Descrição
01	Especificação de Cartões	Requisitos tecnológicos a que os cartões sem contacto do sistema de bilhética devem obedecer, organização / utilização dos respetivos conteúdos (modelo de dados) e regras de codificação dos mesmos.
02	Especificação de Interfaces	Descrição detalhada das interfaces de comunicação máquina-a-máquina entre os sistemas informáticos do Concedente e do Concessionário, indicando os parâmetros de entrada e de saída dos web services que implementam essas comunicações (fluxos 3, 4, 7, 8, 9, 10, 11 e 12).
03	Catálogo de Títulos	Listagem dos títulos de transporte em vigor, incluindo as respetivas datas de início e fim de validade, preços e codificações (segundo as especificações definidas no ponto 01 da presente tabela).
04	Catálogo de Serviços	Conjunto de ficheiros descritores dos serviços, paragens e horários, segundo a norma Static GTFS (General Transit Feed Specification).
05	Fornecimento de SAMs	Cabe ao Concedente o aprovisionamento e o fornecimento, em condições por ele definidas, dos SAMs (Secure Application Modules) necessários ao funcionamento dos equipamentos do sistema de bilhética do Concessionário, segundo os requisitos tecnológicos definidos no ponto 01 da presente tabela.
06	Fornecimento de Cartões	Cabe ao Concedente o aprovisionamento e o fornecimento, em condições por ele definidas, dos cartões sem contacto a utilizar pelos clientes finais no sistema de bilhética do Concessionário, segundo os requisitos tecnológicos definidos no ponto 01 da presente tabela.
07	Criação de Clientes	Registo de novos clientes finais na BDCC (Base de Dados de Clientes e Cartões) do Concedente, que será comum e partilhada entre os vários prestadores de serviços de transporte público a operar na RAM.
08	Emissão/Anulação de Cartões	<p>Registo, na BDCC, de novos cartões emitidos pelo Concessionário e de cartões por este indicados como sendo para invalidar.</p> <p>Os cartões a invalidar passarão a figurar numa lista negra (ver ponto 09 da presente tabela), sendo obrigatória a sua invalidação numa próxima passagem pelo sistema de bilhética do Concessionário (ver ponto 14 da presente tabela), quer a iniciativa da emissão do cartão em causa tenha sido sua ou não.</p> <p>A invalidação de um cartão deve ser reportada através deste mesmo fluxo, para que possa ser retirado da próxima versão da lista negra.</p>



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Fluxo		Descrição
09	Listas Negras	<p>Lista de cartões a invalidar e de cartões que, tendo sido invalidados, podem ser reabilitados.</p> <p>Todos os títulos de transporte contidos num cartão invalidado ficam automaticamente inibidos. Um cartão invalidado, que tenha sido entretanto readmitido, pode ser reabilitado num posto assistido do sistema de bilhética do Concessionário. Considera-se que as listas de readmissão fazem parte integrante das listas negras.</p>
10	Venda/Carregamento de Títulos	<p>Listagem das transações de venda e carregamento de títulos de transporte realizadas pelo Concessionário, num determinado período.</p> <p>Esta listagem deverá conter cada transação de forma individualizada, contendo todos os elementos necessários à sua caracterização, tais como (entre outros) o cliente, data/hora, local, título e preço.</p> <p>Esta informação poderá ser utilizada pelo Concedente para efetuar a repartição de receitas entre operadores, no âmbito de títulos de transporte intermodais.</p>
11	Validação de Títulos	<p>Listagem das transações de validação de títulos de transporte realizadas pelo Concessionário, num determinado período.</p> <p>Esta listagem deverá conter cada transação de forma individualizada, contendo todos os elementos necessários à sua caracterização, tais como (entre outros) o cliente, data/hora, paragem/zona, título e consumo efetuado.</p> <p>Esta informação poderá ser utilizada pelo Concedente para efetuar a repartição de receitas entre operadores, no âmbito de títulos de transporte intermodais.</p>
12	Auditoria	<p>Relatório referentes à qualidade do serviço prestado pelo Concessionário num determinado período, fornecendo elementos confiáveis e auditáveis tais como (entre outros) serviços suprimidos, atrasos médios por carreira e número de reclamações.</p> <p>Os sistemas do Concessionário deverão estar preparados para incluir a disponibilização, no futuro, de informação em tempo real sobre a localização dos veículos e a previsão do seu tempo de passagem nas paragens.</p>
13	Emissão de Cartões	<p>Personalização gráfica e elétrica dos cartões sem contacto a serem posteriormente utilizados pelos clientes finais no sistema de bilhética do Concessionário.</p> <p>Estas operações deverão estar em conformidades com as especificações definidas no ponto 01 e os cartões fornecidos conforme o ponto 06 (ambos os pontos referentes à presente tabela).</p>
14	Invalidação de Cartões	<p>Inibição ou reabilitação de cartões contidos na lista negra (ver pontos 08 e 09 da presente tabela), conforme especificação definida no ponto 01 da presente tabela.</p>
15	Carregamento de Títulos	<p>Carregamento de títulos de transporte em cartões sem contacto, conforme especificação definida no ponto 01 da presente tabela.</p>



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

Fluxo		Descrição
16	Validação de Títulos	Validação de títulos de transporte em cartões sem contacto, conforme especificação definida no ponto 01 da presente tabela. Os equipamentos de bilhética do Concessionário deverão ser compatíveis com a tecnologia NFC utilizada para a implementação de cartões virtuais em smartphones.
17	Emissão de Bilhetes de Bordo	Impressão de bilhetes em papel térmico, comprados a bordo e com validades ao nível da viagem.



REGIÃO AUTÓNOMA DOS AÇORES
SECRETARIA REGIONAL DOS TRANSPORTES E OBRAS PÚBLICAS

ANEXO C. “Modelos de Dados ANDANTE”

Card Data Model

Contactless Ticketing in Porto

AUTHOR

Frederic LEVY
frederic.levy@spirtech.com
Tel.: +33 1 40 46 36 21

SPIRTECH
29 rue du Louvre – 75002 Paris – France

This document is classified « restricted ».

The information contained in this document is not public and must be kept confidential.

Its re-transmission by an original recipient is under its full responsibility. This transmission may only concern people implied in the project, on a need-to-know basis.

REVISION LIST

Version	Date	Modifications	Author
1.7	181126	Main modifications: - Updated and improved the card description sections, adding Calypso Rev.3 cards. - Added possible management of up to 255 zones. - Added TDES keys, removed simple DES keys. Other noteworthy modifications: - The version of all data structures is coded on 3 bits. - Dates are not significant when equal to 0. - Added a section about RFU management. - EnvCurrency replaced by HolderDataCommercialID and RFU bits. - Season tickets use no counter. - Some of the RFU bits at the end of the event data structure are now defined. - Padding bits added when necessary. - References updated. - Minor improvements and corrections.	SDI
1.6	090701	Definition of SpatialParkingData in the Parking Spatial Validity record for contracts. Added Special Event for parking management. Added rules for parking management. Updated the diffusion list. Minor corrections and rewritings.	SDI
1.5	031103	EventContractUsed field reintegrated in the event data structure. Minor corrections and rewritings.	SDI
1.4	031030	Company type data field size increased to 8 bits with Andante Pro codes. Some RFU fields are reduced or removed. ContractPeriodJourneys size reduced to 6 bits. Added padding in the contract data structure. Minor change in the ContractAuthenticator computation. EventContractUsed and EventRFU removed. Added Zones references in the event structure. Added to contracts in the best contracts list. Minor corrections and rewritings.	SDI
1.3	031008	Modification of the Contract structure to put the restrictions parameters under the bitmap. Modification of the ContractRestrictDuration (size and values). Removal of ContractValidityTripDuration. Correction of some field sizes and removal of some unused data types.	FLV
1.2	031007	Definition of the "Priority" data type Corrections of sizes and field status in the Contract data structure Modifications of the Spatial Coding records	FLV
1.1	030513	Added operators USGP, USGL and UVIR (§5.2) Added precision that "Resende" is a bus operator (§5.2)Modification of some of the data structures following the Porto meeting on 030411	FLV
1.0	021024	First released version.	FLV

TABLE OF CONTENTS

1 INTRODUCTION	5
1.1 References	5
1.2 Glossary	6
2 CONTACTLESS CARDS AND TICKETS	7
2.1 Products Overview	7
2.2 Porto Cards	7
2.2.1 Overview	7
2.2.2 Revision 2 Cards	8
2.2.3 Revision 3 Cards	8
2.3 File Access	9
3 PORTO FARE SYSTEM	10
3.1 Porto Fare Zones	10
3.2 Single and Multiple Trips Tickets.....	10
3.3 Season Tickets	11
3.4 Park & Ride.....	11
3.5 Control	12
3.6 Multimodal and Specific Contracts	12
4 SECURITY	13
4.1 Calypso Transaction	13
4.2 Secret Keys	13
4.3 Access Rights	14
4.4 Surveillance System and Ticket Blacklist.....	14
5 DATA TYPES	15
5.1 Notation, Coding and Format	15
5.2 Data Types.....	15
6 DATA MODEL	19
6.1 Notation	19
6.1.1 ENV 1545	19
6.1.2 Data Structures and Bitmaps	19
6.1.3 Data Structures and Files	19
6.1.4 RFU Management	19
6.1.5 Field Status.....	20
6.2 Field Coding Rule	20
6.3 Ticketing Application Identification Data	21
6.4 Environment and Holder Data Structure	21
6.5 Contract Data Structure	25
6.5.1 Spatial Coding	30
6.6 Validation Event.....	35
6.7 Special Event: Parking.....	40

<p>Card Data Model Contactless Ticketing in Porto Ref.: 011201FLV-FS-DataModel</p>	<p>Page 4 / 45 Version 1.7 Date 26 Novembre 2018</p>
---	--

7 MANDATORY USAGE RULES	43
7.1 Best Contract List Management.....	43
7.2 Available Files.....	43
7.3 Park & Ride Management	44
 8 RECOMMENDED USAGE RULES	 45
8.1 Validator Transaction.....	45
8.2 Best Contract List Management.....	45

1 INTRODUCTION

This document contains the precise description of the Data Model for Calypso contactless cards used in the transport applications for the Porto region.

The purpose of this document is to establish an interoperable card and ticket management (common data model and rules) for the Porto region, allowing any terminal to work with any contactless card or ticket.

The low cost contactless tickets are described in another document (*Extended Ticket Data Model, 030618SDI-NT-ExtdTikData*).

1.1 References

030618SDI-NT-ExtdTikData	Ticket Data Model for Porto
010608-NT-CalypsoGenSpecs	Calypso Functional Specification for Ticketing - Card Application
CalypsoTN001-StartupInfo	Calypso Technical Note #001: Startup Information
EN 1545-1	Identification card systems - Surface transport applications - Part 1: Elementary data types, general code lists and general data elements
EN 1545-2	Identification card systems - Surface transport applications - Part 2: Transport and travel payment related data elements and code lists
ISO/IEC 7816-1	Identification cards - Integrated circuit(s) cards with contacts - Part 1: Physical characteristics
ISO/IEC 7816-2	Identification cards - Integrated circuit(s) cards with contacts - Part 2: Dimensions and location of contacts
ISO/IEC 7816-3	Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic interface and transmission protocols
ISO/IEC 7816-4	Identification cards - Integrated circuit(s) cards with contacts - Part 4, Organization, security and commands for interchange
ISO/IEC 3166-1	Codes for the representation of names of countries and their subdivisions - Part 1: Country codes
ISO/IEC 8859-1	Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1
ISO/IEC 4217	Codes for the representation of currencies
ISO/IEC 18033-3	Information technology - Security techniques - Encryption algorithms - Part 3: Block ciphers
ISO/IEC 9797-1	Message authentication techniques - Part 1 : Mechanisms using a block cipher
ISO/IEC 14443-1	Identification cards - Contactless IC cards - Proximity cards - Part 1: Physical characteristics

ISO/IEC 14443-2	Identification cards - Contactless IC cards - Proximity cards - Part 2: Radio frequency power and signal interface
ISO/IEC 14443-3	Identification cards - Contactless IC cards - Proximity cards - Part 3: Initialization and anticollision
ISO/IEC 14443-4	Identification cards - Contactless IC cards - Proximity cards - Part 4: Transmission protocol
ISBN 3-540-61512-1	DESX algorithm definition. Kilian & Rogaway, "How to Protect DES Against Exhaustive Key Search", from <i>CRYPTO'96. Advances in cryptology 16th annual international conference</i> , Springer Verlag

1.2 Glossary

AES	Ciphering algorithm producing 16 bytes of data from 16 input bytes, using a 128, 192 or 256 bits key. Calypso specifies only 128 bits AES keys.
Card	Within this document, a card is any kind of portable object complying with Calypso specifications (Calypso Revision 3, Calypso HCE and Calypso Light (CLAP)), of any form factor (contactless card, mobile phone, etc.).
CNA	Calypso Networks Association (www.calypsonet-asso.org).
DES	Ciphering algorithm producing 8 bytes of data from 8 input bytes, using a 56 bits key.
DESX	Ciphering algorithm producing 8 bytes of data from 8 input bytes, using a 120 bits key. Extension of DES.
Dedicated File	(DF) Equivalent of a directory. A DF contains other files. DF See Dedicated File.
EF	See Elementary File.
Elementary File	(EF) File containing data. There are three types of EF defined by Calypso: linear, cyclic and counter files.
Long File Identifier	(LID) External unique number identifying a file (0000h to FFFEh, without 3FFFh). All files have an LID. The MF has an LID of 3F00h.
Short File Identifier	(SFI) External unique number identifying a file (1 to 30). Some files have no SFI.
KVC	Key Version and Category. Used as a master key index. LID See Long File Identifier.
Record	The data in the files are organized in records of at least 29 data bytes.
Record Number	A file may contain more than one record. The record number identifies one record in the file. Files have record number from 1 to the maximum number of records of the file. For cyclic files, record number 1 is the most recent record added to the file.
RFU	Reserved for Future Use. SAM Security Application Module.
SFI	See Short File Identifier.
TDES	Ciphering algorithm producing 8 bytes of data from 8 input bytes, using a 112 bits key. Extension of DES.

2 CONTACTLESS CARDS AND TICKETS

2.1 Products Overview

A range of products compatible with the Calypso specification exist to manage ticketing capabilities, for example:

Type	Characteristics	Format	Product Examples
Ticket (contactless memory)	Low unitary cost Low security small memory	Paper (disposable) or plastic, contactless only	CTS512 (ASK) SRT512, SRI2K, SRI4K, ST25TB (ST Microelectronics) Mifare UL (NXP)
Smart card (contactless, SIM)	Medium unitary cost High security Transport and other applications	Plastic (possibly with contacts) Mobile device	CD21 (STMicroelectronics), Calypso G1 (Gemalto), Tango Multiapp (Paragon ID), TimeCOS Calypso (Watchdata), SOMA Atlas (Arjo)

The data model for tickets is described in another document (*Extended Ticket Data Model, 030618SDI-NT-ExtdTikData*).

The contactless cards follow the Calypso specification (see *Calypso Functional Specification for Ticketing - Card Application*).

The main characteristics of the contactless microprocessor cards are:

- Variable amount of memory (at least 2 kbytes, organized in files).
- High security:
 - AES, TDES or DESX cryptography¹;
 - set of three secret keys with different access rights to different data.
- Chip accessible through standard communication interfaces: contactless link (ISO 14443) or contacts (ISO 7816-3, when available).

2.2 Porto Cards

2.2.1 Overview

There are two generations of Calypso cards for Porto:

- Calypso Revision 2: cards issued initially, with a minimum transport application.
- Calypso Revision 3: cards issued from 2017, based on previous cards but with 8 contracts (instead of 4) and 64 bytes for each contract (instead of 29 bytes).

The terminal determines which card is present by analyzing the *Application Type* byte in the *Calypso Startup Information* (returned by *Select Application* command).

¹ Simple DES cryptography is still allowed by Calypso specification, but it is obsolete and strongly not recommended for new cards.

Notes:

- In following file structure descriptions, long file identifier (LIDs) are not indicated. They are chosen by the card manufacturer. They must not be used by terminals for normal ticketing operations.
- Cards may have features not described in this document. If present, they must not be used by terminals for the normal processing of ticketing transactions. They may be, for example:
 - Additional files in the Transport application of Revision 2 cards.
 - Master File, or any other kind of application, complying or not with Calypso.
 - PIN code.
 - ATR (ISO 7816-3 T=0) or ATS (ISO 14443 type A).

2.2.2 Revision 2 Cards

The Calypso cards initially issued for Porto comply with Calypso Revision 2 standard, and contain the following files in the transport application:

The Calypso cards initially issued for Porto comply with Calypso Revision 2 standard, and contain the following files in the transport application:

File	Type	SFI	Rec Num	Rec Size	Group 0	Group 1	Group 2	Group 3
					Read Rehabilitate	Update Invalidate	Write Decrease	Append Increase
Transport Application	DF	–	–	–	Session 1	Session 3	–	–
Environment	Linear	07h	1	29	Always	Session 1	Never	–
Events Log	Cyclic	08h	3	29	Always	Session 3	Session 3	Session 3
Special Events	Linear	1Dh	1	29	Always	Session 3	Never	–
Contract List	Linear	1Eh	1	29	Always	Session 3	Never	–
Contracts	Linear	09h	4	29	Always	Session 2	Session 3	–
Counters	Counters	19h	1	29	Always	Session 2	Session 3	Session 2

Its application name (AID) is the legacy Calypso Revision 2 AID:

315449432E494341h ("1TIC.ICA" in ASCII)

The *Startup Information* returned when the application is selected depend on the product (see *Calypso Technical Note #001*, ref. *CalypsoTN001-StartupInfo*).

2.2.3 Revision 3 Cards

The Calypso cards which may be issued for Porto from 2018 comply with Calypso Revision 3 (or Calypso HCE) standard.

Application Name (AID)

The application name (AID) begins with the legacy Calypso Revision 2 AID. Its complete value is:

315449432E494341 D620 0002 9101h.

To manage upward compatibility with possible future versions of the AID, it is recommended that when sending the Select Application command terminals omit the last byte of this AID.

The full AID is present in the response to Select Application. Terminals must process cards regardless of the full AID present in Select Application response.

Startup Information

The *Startup Information* returned when the application is selected is:

Offset	Byte Name	Value	Remarks
00h	Session Modifications	>= 08h	Session modification buffer of at least 304 bytes
01h	Platform	Var.	According to <i>Calypso Technical Note #001</i>
02h	Application Type	20h...7Fh	According to Calypso Revision 3 specifications
03h	Application Subtype	C2h	Current Calypso Revision 3 file structure for Porto
04h	Software Issuer	Var.	According to <i>Calypso Technical Note #001</i>
05h	Software Version	Var.	Depending on the product
06h	Software Revision	Var.	Depending on the product

Terminals must process all applications as Calypso Revision 2 cards, regardless of their Startup Information.

Serial Number

Since the AID of the Revision 3 Porto application is longer than the AID of the Revision 2 application, in the answer to Select Application response the 8-byte Calypso serial number starts at a bigger offset.

Terminals must retrieve the Calypso serial number for any length of the AID, as defined in Calypso Revision 2 and Revision 3 specifications.

File Structure

The file structure will be defined in a later version of this document.

Until then terminals must process applications having:

- Files not defined in section 2.2.2, ignoring them.
- Files with more records than in section 2.2.2, ignoring any additional record.
- Files with records of more than 29 bytes, ignoring bytes beyond the 29th.

2.3 File Access

Each file may be read freely, but its modification is subject to a mutual authentication and an authentication of the data exchanged (see the *Security* chapter and the Calypso specification).

3 PORTO FARE SYSTEM

3.1 Porto Fare Zones

The Porto fare area is divided into a number of fare zones.

A trip is spatially characterized by its starting zone and its ending zone. A given trip starts from a zone and may end in another, possibly crossing several other zones.

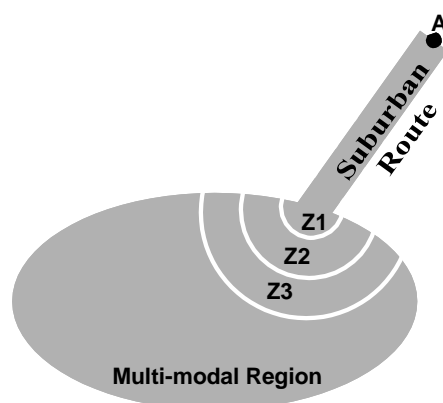
The “range of zones” of the trip is the minimum number of zone borders to cross to go from the starting zone to the ending zone. For example, if the zones are adjacent, the range of zones of the trip is “one”.

The minimum “range of zones” number is “1”, which allows travelling from the starting zone to the same zone or to any adjacent zones.

Suburban Area

A trip may start or end in the suburbs, outside of the multi-modal area.

The illustration of the integration of suburban routes with multi-modal regions, from the ticketing point of view, is presented in the picture below.



In this example, a single ticket (and a single fare) is valid in the Z1+Z2+Z3 multi-modal region and in the suburban route between Z1 and point (or zone) A.

From the point of view of the revenue splitting, the global fare is divided into two components:

- Suburban route component: this share is due only to the suburban provider.
- Multi-modal component: this share is integrated in the global clearance procedures, as any other Z3 ticket in the multi-modal system (the value of this component may be exactly the same of a conventional Z3-only ticket or a lower one, in order to stimulate the offer of suburban combined tickets).

3.2 Single and Multiple Trips Tickets

A single trip ticket allows traveling for a pre-defined range of zones, according to the ticket type (2 zone ticket, 3 zone ticket, etc.). The maximum range of zones is 16 (a “16 zones ticket” allows traveling a range of 16 zones or more).

<p align="center">Card Data Model Contactless Ticketing in Porto Ref.: 011201FLV-FS-DataModel</p>	<p>Page 11 / 45 Version 1.7 Date 26 Novembre 2018</p>
--	---

A multiple trip ticket allows several trips, each trip having the same pre-defined range of zones. The maximum number of trips is 31.

The starting point of a trip may be any Porto fare zone.

A ticket is valid for a fixed amount time depending upon its number of range of zones (e.g. one hour for a 2 zone ticket).

3.3 Season Tickets

A season ticket allows traveling:

- During a time period, from one hour to more than one year (e.g. the current day, all the month of January).
- For any number of trips.
- In a selection of Metropolitan Porto area fare zones.
- Possibly also from or to a defined suburban point or zone, following a defined suburban route.
- From a fix date, or from the date of first use, for a fix duration.

3.4 Park & Ride

The Park & Ride (P&R) season ticket authorizes parking in specific car parks. The following possibilities are forecasted:

a) *Prepaid P&R contactless ticket or card*

The driver enters the parking and presents its *contactless* Park & Ride ticket or card at the entrance. The entrance machine processes the ticket or card, and internally records the date and time of entrance.

It is then possible to use the public transports. When coming back to get the car, the exit validator will let the car go out when presenting the contactless Park & Ride ticket or card.

b) *Parking contactless ticket, or transport card without P&R contract*

If the driver has a transport card, he or she uses it at the entrance machine. Else, the driver gets a *contactless* ticket at the entrance. In both cases, the entrance machine issues a "Parking only" contract, including the date and time of entrance.

It is then possible to go to the transport selling machine and update the ticket or card with a Park & Ride contract, and use the public transports. When coming back to get the car, the exit validator will let the car go out when presenting the contactless Park & Ride ticket or card. (The payment for the transport must occur before being authorized to travel).

If the public transport is not used, the user will pay its parking fee at the park vending machine with the contactless ticket issued or with its card.

c) *Use of magnetic tickets*

The driver enters the parking and gets a *magnetic* ticket at the entrance.

He, or she, then goes to the transport counter to request a "Park & Ride" season ticket for the day. The clerk sells the transport ticket, and updates the parking magnetic ticket to allow exiting from the parking later in the day.

3.5 Control

It is necessary to ensure that the users validate at every part of the trip.

For this, the card must record the time and location of the last validation, to allow a control during the trip, and a possible payment if no validation was done (even if the user has the right to travel at this point).

3.6 Multimodal and Specific Contracts

It is forecasted that in the future only multimodal contracts may exist, which would be accepted by all operators.

However this is not the case today, and the card must have a provision for contracts specific to one or a few operators.

4 SECURITY

4.1 Calypso Transaction

When using a Calypso smart card, the users may wish:

- To ensure that the card data is genuine. It must not be possible for defrauder to forge the data, or to modify it in the card.
- To ensure the integrity of the data written in the card, even if the card power supply is unexpectedly shut down during a single write operation, or during the synchronous update of related files in the card (for example, recording a new event in the card may be linked with a counter decrement).

Calypso solves these needs by a single mechanism called the Secure Session.

A secure session begins by a specific command sent to the card to open the session (*Open Secure Session*), and ends by a specific command to close the session (*Close Secure Session*).

During the session, it is possible to read and write data into the card.

When the session closes, all the data exchanged is certified by the card, and by the SAM included in the terminal. This certification simultaneously:

- proves the authenticity of the terminal to the card (authenticating the terminal),
- proves the authenticity of the card to the terminal (authenticating the card),
- certifies that the data exchanged is genuine and has not been tampered with by a defrauder.

Finally, this command also proves to the terminal that the card has been correctly updated.

The details of the secure session mechanism, as well as a special feature called the *Ratification*, are explained in detail in the *Calypso Functional Specification For Ticketing - Card Application*.

4.2 Secret Keys

The main secret keys to manage the security of a Calypso application are:

Key 1 Issuer key	Issuer Key. It allows changing the other keys, and may be used to authorize the modification, or to verify the value, of some files data. This key is typically used to authorize modifying the data global to the application.
Key 2 Load key	It may be used to authorize the modification, or to verify the value, of some files data. This key is typically used as a reloading key.
Key 3 Debit key	It may be used to authorize the modification, or to verify the value, of some files data. This key is typically used as a debit key.

Currently cards are only produced with DESX keys or with TDES keys². Anyway, terminals must be able to process cards containing keys of any algorithm allowed by Calypso Revision 2 or Revision 3 specifications.

² Even if Revision 2 cards are no more produced with simple DES keys, some of them may still be in use.

To increase the security of the system, the keys written in a card are *diversified*, so that every card has different key values. Thus, if the keys of one card become known, the keys of any other card are still secret.

To simplify this diversification, the keys of each card are computed from *master keys*. Generally, the card key is computed by a cryptographic operation on the card serial number, using the corresponding master key.

The different keys are stored in the terminal in a SAM (Security Application Module). The SAM is a smart card that can authenticate a card and the data received from it, and prove to the card the authenticity of the terminal.

A SAM contains the master keys corresponding to the terminal functions (personalization, loading or debiting).

4.3 Access Rights

The different data types in the card may be modified with one of the DES keys:

- Environment and holder data Issuer Key (Key 1)
- Contract and counter increase Load Key (Key 2)
- Event log and counter decrease Debit Key (Key 3)

It is also possible to *overwrite* a contract during the validator transaction (bits of the contract may be set to “1” but not to “0”).

This allows to invalidate a contract, or to write the date of first use for a contract starting at first use. In order to protect against a fraudulent modification of the contract, the following rules are followed:

- The starting date is coded in “reverse” format, so that setting bits to “1” in a date already written will shorten the contract validity (hence, there is no advantage to modify this date).
- The rest of the contract data is protected by a checksum (called “Authenticator” in the ENV1545 standard) that reflects the number of bits to “0” in the contract (it is therefore impossible to make a correct authenticator by changing bits of the contract or of the authenticator from “0” to “1”).

4.4 Surveillance System and Ticket Blacklist

As for any security system, frauds are possible. Therefore, a surveillance system is recommended to detect such frauds.

The system can keep trace of all events occurring to the cards issued. The transactions are reported to the surveillance system.

The surveillance system manages a copy of the data of the cards. These copies are updated continuously.

It is then possible to detect a fraud and to put the cards that have been defrauded or stolen in a card blacklist. The validators will not accept a card present in the blacklist.

5 DATA TYPES

5.1 Notation, Coding and Format

Notations

Characters	"X...X"	X = ASCII character
Binary	'X...X'b	X = 0 or 1
Hexadecimal	'X...X'h	X = 0 to F

5.2 Data Types

The following data types are largely based upon the ENV 1545 data types. The standard should be read prior to this document.

However, they are coded here in a compact way, suited for the small memory of the ticketing cards.

The fields should be processed in the terminals and central system in the standard ENV 1545 data types format.

Whenever a type is not an ENV 1545 data type, this is indicated.

Name	Bits	Description
Amount	16	Transaction amount in PayUnit units (this amount includes taxes).
Authenticator	V	Element of data allowing authentication of a set of data. May be implemented as a cryptographic signature of the data, or as a simple checksum depending upon its use.
Bitmap	V	Indicates the presence (1) or absence (0) of a sub-field in the card data structure. The lowest significant bit represents the field 0.
Bits1, Bits2 ... Bits8	1 to 8	Bit string of 1 to 8 bits.
BitsN	Var.	Bit string of variable size (N bits).
Boolean	1	Boolean value (0=False, 1=True).
Charv	V	Character string according to ISO 8859-1 of 1 to .255 characters. When the field size is not fixed, a first byte indicates the number of the following characters (0 to 255).
Company	8	Unique identifier for a company within a NetworkId: Operator identifier within the network: 0 (Forbidden, used as marker) 1 ACE 2 Metro do Porto (metro) 3 STCP (bus) 4 CP (train) 5 Resende (bus) 6 USGP (train) 7 USGL (train) 8 UVIR (train) 9 to 31 RFU 32 to 199 RFU (for companies other than transport operators) 200 to 254 Reserved for Andante Pro 255 Andante Pro: Transdev
CountryAlpha	24	Porto reference according to ISO 3166-1. Three ISO 8859-1 characters padded with space characters on the right: 'PRT'.

Name	Bits	Description
CustomerProfile	6	Social profile allowing special fares: 0= Unspecified 1= Adult 2= Child 3= Elderly 4= Pensioner 5= Student (<13 years) 6= Student (>13 years) 7= Student (University) 8= Military 9= Authority 10= Entities 11= Youth Card 12= Industrial Owned Haulage 13= Bus Transport Company 14= Far distance transport 15= Local transport 16= Commuter 17..31= RFU 32..63= RFU specific per operator
Datav	V	Various data of 1 to 256 characters. The first byte indicates the number of the following characters (0 to 255).
DateCompact	14	Number of days since January 1 st , 1997 (date '01'h is January 2 nd , 1997). Last complete year is 2040. Not significant when equal to 0.
DateReverse	14	Same format as DateCompact, but with every bit inverted. DateReverse = DateCompact xor '111111111111'b When using this coding, if a '0' bit is set to '1', the date may only be moved back in time. Not significant when all bits are equal to 1 (coded date equal to 0). <i>(This data type is not defined by ENV1545)</i>
Datef	32	Date given as 8 BCD digits: 'yyyymmdd': • yyyy : year (e.g. '1968') • mm: month (1..12) • dd: day of the month (1..31) Not significant when equal to '00000000'.
Int1, Int2...	8 to 32	Positive number (octet string of 1, 2, 3 or 4 bytes)
NetworkId	24	Unique number identifying a public transport network. In the Porto Region interoperable cards, this number contains the region identification. Given as 6 BCD digits : • 3 for the country (ISO 3166-1): ('620' for Portugal) • 3 for the region ('002' for Porto)
NetworkIdCompact	4	Unique number identifying a public transport network region within Portugal. <i>(This data type is not defined by ENV1545)</i>
PayEventCode	8	Code for the payment event 0=RFU 1=Entrance 2=Entrance detection 3=Re-entrance 4=Exit 5=Exit Detection others : RFU

Name	Bits	Description
PayLocationType	5	Payment location: '00'=bus stop/station '01'=metro station '02'=train station '03'=tram stop/station '04'=ferry gate '05'=toll '06'=parking access '07'=tax '08'=EP reloading point '09'=travel agency '0A'=bank/post office '0B'=reseller '0C'=tunnel '0D'=garage '0E'=gas station '0F'=customs '10'h..'1F'=RUF CEN
PayMethodCompact	3	Payment method. This field is reserved for future use.
PayUnit	16	Unique identification of a currency, as defined in ISO 4217, or a token value specific to a service provider: '0xxx'h main unit of the currency '1xxx'h tenth unit of the currency (1/10) '2xxx'h hundredth of the currency (1/100) '3xxx'h thousandth of the currency (1/1000) '4xxx'h multiple of 10 of the currency '5xxx'h multiple of 100 of the currency '6xxx'h multiple of 1000 of the currency '7xxx'h multiple of 10000 of the currency '8xxx'h multiple of 100000 of the currency where 'xxx' is the BCD currency according to ISO 4217 '9xxx'h, tokens according to the service provider where "xxx" is a specific coding
Priority	4	Priority of a contract: '0'h : Highest level of priority (e.g. Multimodal season contract) '1'h : One below highest level (e.g. Monomodal season contract) '2'h : Two below highest level (e.g. Monomodal season contract) '3'h : Three below highest level (e.g. Multimodal multitrip contract) '4'h : Four below highest level (e.g. Monomodal multitrip contract) ... '9'h : Lowest level of priority (e.g. Contract not for transport) 'A'h : RFU 'B'h : RFU 'C'h : RFU 'D'h : Invalid 'E'h : Erasable (or empty) 'F'h : Absent (record cannot exist in the card, e.g. contract #6 in a 4 contracts card)
RestrictDuration	5	Describes the duration restriction of a journey: 0 RFU 1 to 29 validity duration of the journey in hours 30 end of exploitation 31 end of service (This field is not defined by ENV1545)
RestrictTime	5	Describes period in the day when the contract is not valid. In the present version, this field management is not defined. (The values of this field are not defined by ENV1545)
RestrictCode	5	Describes specific conditions where the contract is not valid. In the present version, this field management is not defined. (The values of this field are not defined by ENV1545)

Name	Bits	Description										
Result	8	<p>Transaction result indicating success or the cause of failure: 'xy'h</p> <p>Where 'x' may be: 0= OK 1= Not OK, no further information 2= Not OK, event anomaly 3= Not OK, contract denied 4= Not OK, EP or account denied</p> <p>And where 'y' may be: 0= No further information 1= Balance close to 0 2= Negative balance 3= Balance overflow 4= Provider denied 5= Authentication incident 6= Incorrect vehicle class</p>										
Tariff	16	<p>Type of contract:. Its coding is not defined in the ENV1545 standard. The following coding is defined for Porto: 'm pp tt dd cc xx nnnnn'b.</p> <p>Where m: '0' = The Tariff is specific to the provider '1' = The Tariff is multimodal</p> <p>Where pp: '00' = RFU '01' = transport access only '10' = parking access only '11' = parking and transport access</p> <p>And where tt: '00' = multi trip tickets '01' = season ticket '10' = RFU '11' = RFU</p> <p>And where dd: '00' = duration/period in months '01' = duration/period in weeks '10' = duration/period in days '11' = duration/period in hours</p> <p>And where cc: '00' = counter as date and number of journeys '01' = counter as number of tokens '10' = counter as cent of euros '11' = counter RFU</p> <p>For season tickets (bits tt equal to '01'b) no counter is used and bits cc are not significant. And where xx is '00' (other values are RFU). And where nnnnnn is the contract name reference (defined according to Network Id):</p> <table><tr><td>0</td><td>RFU</td></tr><tr><td>1</td><td>Multiple trip ticket</td></tr><tr><td>2</td><td>Season ticket</td></tr><tr><td>3</td><td>Park & Ride season ticket.</td></tr><tr><td>3 to 31</td><td>RFU (may be used in the future for new versions of the tickets defined above, or for completely new types of tickets)</td></tr></table>	0	RFU	1	Multiple trip ticket	2	Season ticket	3	Park & Ride season ticket.	3 to 31	RFU (may be used in the future for new versions of the tickets defined above, or for completely new types of tickets)
0	RFU											
1	Multiple trip ticket											
2	Season ticket											
3	Park & Ride season ticket.											
3 to 31	RFU (may be used in the future for new versions of the tickets defined above, or for completely new types of tickets)											
TimeCompact	11	Number of minutes since 0:00, from 0 to 1439.										
TimeReal	30	<p>Date and time</p> <p>Number of seconds elapsed since 1/1/1997 at 0:00 GMT, (allows coding all instants until year 2031)</p> <p>Not significant when equal to 0.</p>										
VersionNumber	3	Data structure version number										

6 DATA MODEL

6.1 Notation

6.1.1 ENV 1545

The data structures are based upon the EN 1545 standard.

However, the card memory being limited, the structures have been adapted to fit in the card memory.

6.1.2 Data Structures and Bitmaps

A data structure is a fixed sequence of data elements defined by:

- The structure name and definition.
- The ordered list of data elements of the structure.

Every data element may be a simple or complex data type (e.g. another data structure).

Most data elements are optional: a *bitmap* introduces the list and contains one bit per element. The value of this bit indicates if the element is present (1) or absent (0) from the data structure.

Some elements may be repeated a number of times. In this case, the element is introduced by a repeat counter containing the number of data elements that follows.

The elements are defined on a bit size basis (not aligned on byte boundaries). Therefore, the terminal must decode the data structure on a bit size basis to be able to interpret its content.

6.1.3 Data Structures and Files

A card file may contain one or more data structures.

6.1.4 RFU Management

RFU Values

For a given data field, the present specification may indicate a list of possible values. All other values are reserved for future use (*RFU values*). Terminals shall not set a field to an RFU value.

RFU Fields

For a given set of data fields, the present specification may indicate some fields as reserved for future use (*RFU fields*).

To avoid compatibility issues in futures releases of the specification:

- When writing in cards, terminals shall set RFU fields to zero (unless another value is indicated).
- When reading from cards, RFU fields are shall be indifferent for terminals.

6.1.5 Field Status

In the data structures definition, when there are possible optional fields each field may be marked as:

	Type	Description
M	Mandatory	The field must be present in the data structure. Its value must be correctly set to allow interoperability.
O	Optional	The field may be present but is not required. If present, its value must follow the rules indicated in this document for this field. Its presence or absence must not prevent the correct processing of the data structure for interoperability.

6.2 Field Coding Rule

The data structures are described in this document as a sequential list of fields. The field sizes are a number of bits, not aligned on bytes boundaries in the ticket.

The first field (top of list) is coded in the first byte of the memory (lower address). The most significant bit of the first field is to the most significant bit of the first byte.

Bytes are ordered from 0 to N, and, in a byte, bits are ordered from b7 (most significant) to b0 (least significant).

For example the following fields:

Name	Type	Size [bits]	Value
Field 1	(any)	2	00h = '00'b
Field 2	(any)	4	03h = '0011'b
Field 3	(any)	5	12h = '10010'b
Field 4	(any)	6	0Fh = '001111'b

Are coded as:

Field 1		Field 2				Field 3					Field 4						...		
b1	b0	b3	b2	b1	b0	b4	b3	b2	b1	b0	b5	b4	b3	b2	b1	b0	...		
0	0	0	0	1	1	1	0	0	1	0	0	0	1	1	1	1	...		
Byte 0								Byte 1								Byte 2 ...			
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	...
0	0	0	0	1	1	1	0	0	1	0	0	0	1	1	1	1	1	...	

Here is another example with the following fields:

Name	Type	Size [bits]	Value
Field 1	(any)	8	35h = '00110101'b
Field 2	(any)	16	0C5Fh = '0000110001011111'b

Which are coded as:

Field 1								Field 2																...					
7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	...								
0	0	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	1	1	...								
Byte 0								Byte 1								Byte 2								...					
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b7	b7	b7	b7	...								
0	0	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	1	1	...								

Note that the coding is therefore Big Endian (MSB first, at lowest address).

6.3 Ticketing Application Identification Data

At the beginning of a card transaction, the ticketing cards will return some identification information when the ticketing application is selected.

Among this information is the card serial number. The serial number is an 8 byte number that uniquely identifies a card ticketing application.

This number is used to identify a card, and allows: fraud detection, managing the black list, and card authentication (thru the key diversification mechanism).

6.4 Environment and Holder Data Structure

Environment And Holder Data Structure	Pos.	Bits	Description	Type
EnvApplicationVersionNumber		3	Environment structure version number (0)	VersionNumber
EnvNetworkIdNetwork		12	Porto Network Identifier within country	Bits12
EnvNetworkIdCountry		12	Country ISO identification	Bits12
EnvApplicationIssuerId		8	Ticketing application Issuer Identifier (Value according to EnvNetworkId)	Company
EnvCardNumber		24	Card number issued by EnvApplicationIssuerId	Bits24
EnvIssuingDate		14	Transport Application Issuing Date	DateCompact
EnvEndDate		14	Expiration date of the ticketing application	DateCompact
EnvPayMethod		3	RFU field to indicate the payment method	PayMethodCompact
HolderDataCommercialID		8	Commercial type of the card	Bits8
HolderRFU		4	RFU = '1111'b	Bits4
HolderBirthDate		32	Birth date	Datef
HolderReductionRate		7	RFU for a reduction rate indication	Bits7
HolderCompany		8	Holder company	Company
HolderIdNumber		20	Holder Identifier	Int20
HolderProf1Code		6	Profile Code	CustomerProfile
HolderProf1Date		14	Profile validity ending date	DateCompact
HolderProf2Code		6	Profile Code	CustomerProfile
HolderProf2Date		14	Profile validity ending date	DateCompact
HolderProf3Code		6	Profile Code	CustomerProfile
HolderProf3Date		14	Profile validity ending date	DateCompact
HolderPadding		3	Padding (0)	Bits3

The size of the environment and holder data structure is fixed: 232 bits (29 bytes).

EnvVersionNumber

Definition	The version of the environment and holder data structure defined in this document is version '0'.
Use	The terminals must reject applications with a greater value of this field.

EnvNetworkIdNetwork

<i>Definition</i>	This field is coded in BCD and contains the network identification number for the Porto region within Portugal: 002h.
<i>Use</i>	The terminals must reject applications with an unknown value of this field.

EnvNetworkIdCountry

<i>Definition</i>	This field is coded in BCD and takes the value 620h (ISO 3166-1 code for Portugal).
<i>Use</i>	The terminals must reject applications with an unknown value of this field.

EnvApplicationIssuerId

<i>Definition</i>	Identifies the issuer who created the transport application.
<i>Use</i>	The terminals should only record this field value during a transaction, but should not reject or modify the card transaction based on this field value.

EnvCardNumber

<i>Definition</i>	Card issuing number (by Issuer ID company)
<i>Use</i>	The terminals should only record this field value during a transaction, but should not reject or modify the card transaction based on this field value.

EnvIssuingDate

<i>Definition</i>	Card issuing date
<i>Use</i>	The terminals should only record this field value during a transaction, but should not reject or modify the card transaction based on this field value.

EnvEndDate

<i>Definition</i>	Expiry date of the transport application.
<i>Use</i>	The terminals must reject cards with an expired end of validity date.

EnvPayMethod

<i>Definition</i>	RFU field
<i>Use</i>	The terminals should only record this field value during a transaction, but should not reject or modify the card transaction based on this field value.

HolderDataCommercialID

<i>Definition</i>	Commercial type of card, coded in ASCII: '47'h Andante GOLD (71: "G" in ASCII) '50'h CP Porto (80: "P" in ASCII) '53'h STCP (83: "S" in ASCII) Others RFU The graphical layout of cards may be different for each commercial type.
	The vending terminals may use this field to define the eligible offer.
<i>Use</i>	The validation and inspection terminals should only record this field value during a transaction, but should not reject or modify the card transaction based on this field value.

HolderRFU

<i>Definition</i>	RFU field
<i>Use</i>	Must be set to '1111'b. The terminals should only record this field value during a transaction, but should not reject or modify the card transaction based on this field value.

HolderBirthDate

<i>Definition</i>	Date of birth of the holder.
<i>Use</i>	If this field is valid, the terminals may base reduction rate upon the age of the holder.

HolderReductionRate

<i>Definition</i>	Reduction rate indication, from 0 to 100%
<i>Use</i>	This field allows the selling equipment to compute a fare reduction.

HolderCompany

<i>Definition</i>	Company of which the holder is employee
<i>Use</i>	This field should not be used during validation, but may be used for physical access control, for example

HolderIdNumber

<i>Definition</i>	Number of the holder (Client ID or employee number of the holder company according to the Holder Profile codes)
<i>Use</i>	This field may allow to identify the holder of the card during control, physical access control, etc.

HolderProfileXCode

<i>Definition</i>	Type of the profile.
<i>Use</i>	The profile may be used to identify social status, operator employee or employee family, etc., and may be used to access special services or reduction rates. Not significant when HolderProfileXDate is not significant (=0).

HolderProfileXDate

Definition	End of validity date of the profile
Use	This date limits the validity of the profile.

HolderPadding

Definition	Padding bits.
Use	Must be set to 0 when written, and when read any value must be allowed.

Example:

Environment Data Structure	Bits	Value
EnvApplicationVersionNumber	3	'000'b
EnvNetworkIdNetwork	12	'002'h
EnvNetworkIdCountry	12	'620'h
EnvApplicationIssuerId	8	'01'h
EnvCardNumber	24	'0123456789AB'h
EnvIssuingDate	14	1 st September 2003: '0982'h
EnvEndDate	14	1 st September 2003: '0982'h
EnvPayMethod	3	'000'b
HolderDataCommercialID	8	"S" ('53'h)
HolderRFU	4	'1111'b
HolderBirthDate	32	11 January 1962 = '19620111'h
HolderReductionRate	7	'0000000'b
HolderCompany	8	'00000001'b
HolderIdNumber	20	'0123456789A'h
HolderProf1Code	6	'000000'b
HolderProf1Date	14	'00000000000000'b
HolderProf2Code	6	'000000'b
HolderProf2Date	14	'00000000000000'b
HolderProf3Code	6	'000000'b
HolderProf3Date	14	'00000000000000'b
HolderRFU	3	'000'b

The bit coding of the beginning of the data structure would be:

'<000> <0000 0000 0010> <0110 0010 0000> <0000 0001> <0001 0010 0011 0100 0101 ...>b

or, organized in bytes and padded on the right with zeroes:

'00000000 00000100 11000100 00000000 00100010 01000110 10001 ... 'b

which would translate to the following bytes values:

'00 04 C4 00 42 46 ... 'h.

6.5 Contract Data Structure

In addition to the contract data structure, the associated counter may contain a number of units (number of trips, of tokens or of currency).

Contract Data Structure	Pos	Bits	Description	Type	S
ContractVersionNumber		3	Contract structure version number (1)	Bits2	M
ContractValidityStartDate		14	Start date of the contract	DateReverse	M
ContractNetworkId		4	Network region within Portugal	NetworkIdCompact	M
ContractProvider		8	Provider identifier	Company	M
ContractTariff		16	Type of contract	Tariff	M
ContractVehicleClassAllowed		2	Class allowed	Bits2	M
ContractSaleDate		14	Sale date of the contract	DateCompact	M
ContractSaleDevice		12	Terminal that sold the contract	Bits12	M
ContractSaleNumberDaily		10	Number of the sale in the day for the device	Bits10	M
ContractJourneyInterchanges		1	Set to 1 if interchange is allowed	Bits1	M
ContractValidityInfo		6	Bitmap	Bitmap	M
ContractRestrictTime	[0]	5	Time period restrictions	RestrictTime	O
ContractRestrictCode	[1]	5	General restrictions	RestrictCode	O
ContractRestrictDuration	[2]	5	Duration restriction of a journey	RestrictDuration	O
ContractValidityEndDate	[3]	14	Last day of validity of the contract	DateCompact	O
ContractValidityDuration	[4]	8	Validity duration for the contract. The time unit (hour, days, month...) is defined according to ContractTariff.	Int1	O
ContractPeriodJourneys	[5]	6	Maximum number of trips authorized for every period of the contract. The period (day, week, month...) is defined according to ContractTariff.	Bits6	O
ContractValiditySpatial1		var	Spatial validity record	See §6.5.1	M
ContractValiditySpatial2		var	Spatial validity record	See §6.5.1	O
ContractValiditySpatial3		var	Spatial validity record	See §6.5.1	O
...	(possibly other spatial validity records)	See §6.5.1	O
ContractValiditySpatialEndList		3	End of list of validity locations	'111'b	M
ContractPadding		var	Padding with zeroes (0 to 132 bits)	BitsVar	M
ContractAuthenticator		8	Security authenticator	Authenticator	M

The size of the contract data structure is fixed: 232 bits (29 bytes). The

counter associated with the contract has the following structure:

Counter Data Structure	Pos	Bits	Description	Type
CounterDate		14	Date of last validation with usage of the counter	DateReverse
CounterValue		10	Number of units (Unit is according to ContractTariff)	Bits10

ContractVersionNumber

<i>Definition</i>	The version of the contract data structure defined in this document is version '001' ^b .
<i>Use</i>	The terminals must ignore a contract with a greater value of this field. For lower values, please see previous versions of this document ³ .

ContractValidityStartDate

<i>Definition</i>	Start date of validity of the contract.
<i>Use</i>	The reloading machine may set it to the start date of validity (reverse coding), or to '0' for a contract with a starting date of validity at first use. The validator may set it to the date of first use if it contains '0'.

ContractNetworkId

<i>Definition</i>	Digit identifying the region network ID in Portugal
<i>Use</i>	The validator uses it to check the validity of the contract at its entrance.

ContractProvider

<i>Definition</i>	Identity of the organization upon which the obligation rests to provide the service specified in the contract. This may be a single operator or a group of operators organized in a network.
<i>Use</i>	The reloading machine sets it to the code of the contract provider.

ContractTariff

<i>Definition</i>	Number that identifies the rules applying to the determination of the price, validation and checking of a contract. These rules and their application are known to the system. This applies both for contracts used as tickets in public transport, as well as for contracts for payment of tolls or other services.
<i>Use</i>	The validator uses this value to determine the main characteristics of the contract (see the definition of Tariff) and to determine the contract type for a possible display of its name.

ContractVehicleClassAllowed

<i>Definition</i>	Code referencing the vehicle classes which may be used together with the contract. A contract may allow use with more than one vehicle class. The vehicle class codes are only meaningful in the context of the contract provider.
<i>Use</i>	The validator or control uses this value to authorize entrance to a specific passenger class.

³ The version changed from 0 to 1 because of spatial coding modifications (see section 6.5.1).

ContractSaleDate

<i>Definition</i>	Sale date of the contract.
<i>Use</i>	<p>The reloading machine will record the date of sale of the contract.</p> <p>The validator may return this information for central management, and may use it to block a contract reloaded with a stolen machine.</p>

ContractSaleDevice

<i>Definition</i>	Terminal that sold the contract.
<i>Use</i>	<p>The reloading machine will record its number.</p> <p>The validator may return this information for central management, and may use it to block a contract reloaded with a stolen machine.</p>

ContractSaleNumberDaily

<i>Definition</i>	Number of the sale within the sale date for that sale device.
<i>Use</i>	<p>The reloading machine will record this number and increment it after each sale.</p> <p>The validator may return this information for central management, and may use it to block a contract reloaded with a stolen machine.</p>

ContractJourneyInterchanges

<i>Definition</i>	Authorize or forbids interchanges
<i>Use</i>	Set to 1 if interchange is allowed

ContractRestrictTime

<i>Definition</i>	<p>Reference to time periods in a day when the contract is not valid. This code is known to the system and it will therefore identify the period start and end times of non-validity.</p> <p>For example:</p> <p>1: no restriction</p> <p>2: not valid on weekends</p> <p>3: not valid during week days</p> <p>4: not valid during peak hours</p> <p>...</p>
<i>Use</i>	The validator or control uses this value to forbid entrance during specific time periods

ContractRestrictCode

<i>Definition</i>	Reference to contract restrictions.
<i>Use</i>	<p>The validator or control uses this value to forbid entrance on specific usages (e.g. entrance to channel for disabled persons).</p> <p>For example:</p> <ul style="list-style-type: none"> 1: Transport only 2: Transport and parking 3: Transport and disabled channels 4: Transport, parking and disabled channels authorized 5: Access to quay only other: rfu

ContractRestrictDuration

<i>Definition</i>	Restriction on the duration of a journey
<i>Use</i>	The control uses this value to check if the contract is still valid during the trip. The validator use this value to check if the entrance is an interchange or a new journey.

ContractValidityEndDate

<i>Definition</i>	End date of validity of the contract including this date. Period ends at 24 hours on the end date.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

ContractValidityDuration

<i>Definition</i>	Length of the period of validity. The period starts at the validity start date of the contract. The unit of duration will be tariff dependent, for example it may be days, hours, minutes, etc.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

ContractValidityTripDuration

<i>Definition</i>	<p>Duration of validity of a trip in hours. This value overrides the default value present in the terminals.</p> <p>For example, for a multitrip ticket contract, each ticket is valid for this number of hours after the first validation.</p>
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

ContractPeriodJourneys

<i>Definition</i>	Number of journeys allowed per period. The period will be specific to each particular contract. It will usually be a day, however the flexibility exists for other time periods, such as week or month, to be used.
<i>Use</i>	The validator uses this value to compute the number of journeys remaining at the beginning of a new period.

ContractValiditySpatial

<i>Definition</i>	Defines the list of locations where the contract is valid. (See coding below.)
<i>Use</i>	The reloading machine initializes this field. The validator use it to check the contract validity at the present location.

ContractValiditySpatialEndList

<i>Definition</i>	Defines the end of the location list.
<i>Use</i>	The reloading machine initializes this field. The validator uses it to find the end of the spatial location list.

ContractPadding

<i>Definition</i>	Number of "0" bits necessary to fill the space between ContractValiditySpatialEndList and ContractAuthenticator.
<i>Use</i>	Inserted by the contract issuing terminals. Ignored by all other terminals (excepted for the contract authenticator computation).

ContractAuthenticator

<i>Definition</i>	Security code allowing to verify the structure's integrity (checksum).
<i>Use</i>	The validator or control uses this value to check the validity of the contract. The authenticator is a checksum of all the other contract data (it is equal to the number of bits to "0" plus 5). <i>Warning:</i> When computing the checksum, all terminals must assume that the contract validity start date is null, whatever its actual value.

CounterDate

<i>Definition</i>	Date of last validation with usage of the counter
<i>Use</i>	The validator may use this field when validating a contract with a number of journeys per day. If the date indicated is the same date and if the CounterValue part is 0, then it is not possible to use that contract for a journey. If the CounterValue is not 0, it may be decremented to allow the journey. If the date is not the present date, it is possible to decrement the counter so that the date becomes the present date and the CounterValue part is reset to the number of authorizes journeys for the day (minus one for the present journey).

CounterValue

<i>Definition</i>	Number of units. The unit meaning depends upon ContractTariff and may be a number of journeys, tokens or euro cent (see Tariff definition).
<i>Use</i>	May be decremented by a validator during the validation. May be incremented (or re written) by a vending machine during a reload.

6.5.1 Spatial Coding

The possible coding for the Spatial Validity record are as follows.

#0.1: Multimodal Zones, All Zones

SpatialType	3	"Zones multimodal"	'000'b
SpatialMultiZonesSize	6	Equal to 0 The contract is valid in all multimodal zones	Bits6

#0.2: Multimodal Zones, List

SpatialType	3	"Zones multimodal"	'000'b
SpatialMultiZonesSize	6	From 1 to 15 (0Fh): number of 8-bit zone numbers in the list, divided by 4 The number of zones must be a multiple of 4 (and <>0) Examples: 1: List of 4 zones (minimum list size = 32 bits) 2: List of 8 zones (list size = 64bits) 15: List of 60 zones (maximum list size = 480 bits)	Bits6
SpatialMultiZonesList	size x 32	List of authorized zones (32 to 480 bits) The actual maximum size depends on the maximum size of a contract, and on the presence and size of the optional	BitsN

#0.3: Multimodal Zones, Long Bitmap

SpatialType	3	"Zones multimodal"	'000'b
SpatialMultiZonesSize	6	From 16 (10h) to 31 (1Fh): size of the zones bitmap, divided by 64, then plus 15 The number of zones must be a multiple of 64 (and <>0) Examples: 16: Bitmap of 64 zones (minimum size) 20: Bitmap of 320 zones 31: Bitmap of 1024 zones (maximum size)	Bitmap
SpatialMultiZonesBitmap	(size - 15) x 64	Bitmap of authorized zones (64 to 1024 bits) The actual maximum size depends on the maximum size of a contract, and on the presence and size of the optional	BitsN

#0.4: Multimodal Zones, Short Bitmap

SpatialType	3	"Zones multimodal"	'000'b
SpatialMultiZonesSize	6	From 32 (20h) to 62 (3Eh): size of the zones bitmap, minus 8 The value 63 (3Fh) is RFU Example: 56: Bitmap of 64 zones	Bits6
SpatialMultiZonesBitmap	size + 8	Bitmap of authorized zones (40 to 70 bits) The actual maximum size depends on the maximum size of a contract, and on the presence and size of the optional	Bitmap

#1: Multimodal Distance

SpatialType	3	"Origin-Destination" multimodal	'001'b
SpatialMultiDistance	5	Maximum number of zone crossings for a journey: 1 (for "Z2") to 30 (for "Z31") 31 for "Z32 or more" Not significant when equal to 0	Bits5

#2: Monomodal Zones

SpatialType	3	"Zones monomodal"	'010'b
SpatialMonoServiceProvider	8	Service Provider for this spatial validity record	Company
SpatialMonoZonesSize	6	Size of the Zones bitmap, minus 8	Bits6
SpatialMonoZonesBitmap	size + 8	Bitmap of authorized zones (8 to 71 bits)	Bitmap

#3: Monomodal Distance

SpatialType	3	"Origin-Destination" monomodal	'011'b
SpatialMonoServiceProvider	8	Service Provider for this spatial validity record	Company
SpatialMonoDistance	22	Distance authorized for the journey (value according to service provider)	Bits22

#4: Monomodal Origin-Destination

SpatialType	3	"Origin-Destination" monomodal	'100'b
SpatialMonoServiceProvider	8	Service Provider for this spatial validity record	Company
SpatialMonoOriginDestination	22	Authorized location references (value according to service provider)	Bits22

#5: Parking

During transport debit or validation, a terminal should ignore the Parking Spatial Validity record (the contract is processed as if there was no Parking Spatial Validity record). The transport selling terminals not managing park & ride contracts should not reload a contract containing a Parking Spatial Validity record.

SpatialType	3	“Parking”			‘101’b
SpatialParkingData	22	Field	Bits	Description	Bits22
		SPDVersion	2	Record version	
		SPDType	2	Type of rights	
		SPDAreald	7	Parking area (or set of areas) identifier	
		SPDParameters	10	According to SPDType (see below)	

#6: RFU

Terminals should ignore contracts containing an RFU Spatial Validity record. They should never reload them, and for debit and validation they should process them as not allowing access.

SpatialType	3	"RFU"	'110'b
SpatialRfuDataSize	6	Size of the RFU data, minus 8	Bits6
SpatialRfuData	size + 8	RFU (8 to 71 bits)	Bits Var

#7: End of List

SpatialType	3	End of List	'111'b
-------------	---	-------------	--------

SpatialType

<i>Definition</i>	Defines the meaning of the remaining structure
<i>Use</i>	Value '110'b is RFU

SpatialMultiZonesSize

<i>Definition</i>	Subtype of the Spatial Validity record, and size of the next field:
	0: Multimodal Zones, All Zones
	1 to 15: Multimodal Zones, List; followed by SpatialMultiZonesList of size = SpatialMultiZonesSize × 32
	16 to 31: Multimodal Zones, Long Bitmap; followed by SpatialMultiZonesBitmap of size = (SpatialMultiZonesSize – 15) × 64
	32 to 62: Multimodal Zones, Short Bitmap; followed by SpatialMultiZonesBitmap of size = SpatialMultiZonesSize + 8
	63: RFU
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SpatialMultiZonesList

<i>Definition</i>	List of 8-bit numbers identifying the multimodal zones authorized for the contract. The zones corresponding to all numbers (except 255) are not defined in the present document. Zone number 255 is RFU.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SpatialMultiZonesBitmap

<i>Definition</i>	Bitmap of multimodal zones authorized for the contract. The zones corresponding to all bits are not defined in the present document. However, in all bitmaps (regardless of their size), the n th bit (left to right, the 1 st bit being the leftmost/most significant bit) always matches the same zone. For example, if in the short bitmap (SpatialMultiZonesSize=16 to 31) the first (leftmost) bit controls access to multimodal zone “A”, then in the long bitmap (SpatialMultiZonesSize=32 to 62) the first bit controls also access to multimodal zone “A”.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SpatialMonoServiceProvider

<i>Definition</i>	Identity of the organization upon which the obligation rests to provide the service specified in the contract, for the spatial area described in this Spatial Validity record. This may be a single operator or a group of operators organized in a network.
<i>Use</i>	The validator or control of this provider uses this value to check the validity of the contract, other providers ignore this spatial record.

SpatialMonoZonesBitmap

<i>Definition</i>	Bitmap of monomodal zones authorized for the contract, according to SpatialMonoServiceProvider.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SpatialMultiDistance

<i>Definition</i>	Distance authorized by the contract: range of zones crossings for a single journey.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SpatialMonoDistance

<i>Definition</i>	Distance authorized by the contract, according to SpatialMonoServiceProvider.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SpatialMonoOriginDestination

<i>Definition</i>	Designated starting point and ending point of the journey under the contract. The point in this context may be a zone, a bus-stop, a station, a toll plaza, etc., according to SpatialMonoServiceProvider.
<i>Use</i>	The validator or control uses this value to check the validity of the contract.

SPDVersion (SpatialParkingData)

<i>Definition</i>	The version of the Parking Spatial Validity record defined in this document is '0'.
<i>Use</i>	The terminals must ignore a Spatial Validity record with an unknown value of this field (excepted for the contract authenticator computation).

SPDType (SpatialParkingData)

<i>Definition</i>	Type of the parking usage rights: 0 RFU. 1 <i>Parking season ticket</i> : access to the parking area is allowed while the transport rights are valid. 2 <i>Parking units</i> : the remaining number of units is indicated by the counter #9. 3 RFU.
<i>Use</i>	The terminals must ignore a Spatial Validity record with an RFU value of this field (excepted for the contract authenticator computation). Note: If several parking contracts with units are present, they share the units counter.

SPDAreald (SpatialParkingData)

Definition	Identifier for the area (or set of areas) within ContractNetworkId where the park & ride contract applies:
	0 RFU.
	1 to 125 Reference of the parking area (or set of areas), <i>to be defined</i> .
	126 Parking areas defined by the other Spatial Validity records ⁴ .
	127 All parking areas.
Use	The terminals use this value to check the validity of the contract.
	The terminals must ignore a Spatial Validity record with an RFU value of this field (excepted for the contract authenticator computation).

SPDParameters (SpatialParkingData)

Definition	This field depends of the type of parking rights (SPDType).
	For a parking season ticket, it defines possible restriction on the parking access:
	0 No restriction.
	1 to 15 Restrictions common to all parking area operators, <i>to be defined</i> .
	16 to 30 Restrictions specific to SPDAreald operator, <i>to be defined</i> .
	Other RFU.
	For parking units, it contains the initial number of units (value of the units counter after the contract issuing), 1 to 1,023 (0 is RFU).
Use	The terminals use this value to limit the entrance allowed conditions.
	The terminals must ignore a Spatial Validity record with an RFU value of this field (excepted for the contract authenticator computation).

⁴ For example, all parking areas in the zones defined by Multimodal Zones, or parking areas at the stations defined by Monomodal Origin-Destination.

6.6 Validation Event

Event Data Structure	Pos	Bits	Description	Type
EventVersionNumber		3	Event structure version number (0)	VersionNumber
EventDataTimeStamp		30	Date and time of event	TimeReal
EventZone		7	LSbits of the multimodal zone of the event	Bits7
EventContractPointer		3	Number of the contract used for this event (0 if irrelevant)	Bits3
EventFirstDataTimeStamp		30	Time of first boarding (in case of interchanges, indicates the first boarding time, if first boarding, this field is identical to EventDataTimeStamp)	TimeReal
EventFirstZone		7	LSbits of the multimodal zone of first boarding (in case of interchanges, indicates the first boarding zone, if first boarding, this field is identical to EventZone)	Bits7
EventJourneyInterchange		1	Set to 0 for a first boarding, set to 1 if interchange	Bits1
EventContractUsed		8	Bitmap of the contracts used for this event (bit b0 for contract #1)	Bits8
EventBestContractPriority1		4	Contract priority level	Priority
EventBestContractPriority2		4	Contract priority level	Priority
EventBestContractPriority3		4	Contract priority level	Priority
EventBestContractPriority4		4	Contract priority level	Priority
EventBestContractPriority5		4	Contract priority level	Priority
EventBestContractPriority6		4	Contract priority level	Priority
EventBestContractPriority7		4	Contract priority level	Priority
EventBestContractPriority8		4	Contract priority level	Priority
EventCode		8	Type of the event	PayEventCode
EventNetworkId		4	Network region within Portugal	NetworkIdCompact
EventServiceProvider		8	Provider identifier (value according to network region)	Company
EventLocationId		16	Location or vehicle identifier (value according to EventServiceProvider)	Int2
EventDevice		16	Terminal identifier (value according to EventServiceProvider)	Int2
EventRouteNumber		8	Route identifier (value according to EventServiceProvider)	Int1
EventJourneyRun		12	Run identifier (value according to EventServiceProvider)	Bits12
EventFirstServiceProvider		8	Provider of first boarding	Company
EventFirstZoneMono		7	LSbits of the monomodal zone of first boarding (value according to EventFirstServiceProvider)	Bits7
EventFirstLineZone		5	Monomodal line of first boarding (value according to EventFirstServiceProvider)	Bits5
EventZoneMSB		1	MSbit of the multimodal zone of the event	Bits1
EventFirstZoneMSB		1	MSbit of the multimodal zone of first boarding	Bits1
EventFirstZoneMonoMSB		1	MSbit of the monomodal zone of first boarding	Bits1
EventProviderRFU		16	RFU	Bits16

The size of the event structure is fixed: 232 bits (29 bytes).

EventVersionNumber

Definition	The version of the event data structure defined in this document is version '0'.
Use	The terminals must ignore an event log with a greater value of this field.

EventDateTimeStamp

Definition	Date and time at which the event took place.
Use	The date and time of the validation may be used for example: during validation to avoid another immediate transaction and to allow interchanges, during control for verification, for passenger information.

EventZone and EventZoneMSB

Definition	Multimodal zone of the event. The zones corresponding to each value (except 255) are not defined in the present document. Zone number 255 is RFU.
Use	The 8-bit zone number is split as follows: <ul style="list-style-type: none">• 7 LSBits in EventZone• 1 MSbit in EventZoneMSB The zone of the validation may be used for example: during validation to avoid another immediate transaction and to allow interchanges, during control for verification, for passenger information.

EventContractPointer

Definition	Code referring to the contract related to the concerned event.
Use	May be used during control during the trip or for interchanges.

EventFirstDataTimeStamp

Definition	Time at when the journey started (when the first entrance took place).
Use	May be used during control during the trip or for interchanges.

EventFirstZone and EventFirstZoneMSB

<i>Definition</i>	Multimodal zone of first boarding (in case of interchanges, indicates the first boarding zone, if first boarding, this field is identical to EventZone). The zones corresponding to each value (except 255) are not defined in the present document. Zone number 255 is RFU.
<i>Use</i>	The 8-bit zone number is split as follows: <ul style="list-style-type: none"> • 7 LSBits in EventFirstZone • 1 MSbit in EventFirstZoneMSB May be used during control during the trip or for interchanges.

EventJourneyInterchange

<i>Definition</i>	Set to '0' for a first boarding, and to '1' for an interchange.
<i>Use</i>	May be used during control during the trip or for interchanges.

EventContractUsed

<i>Definition</i>	Bitmap of the contracts related to the concerned event. The last (rightmost, least significant) to first (leftmost, most significant) bits match respectively contract records #1 to #8.
<i>Use</i>	May be used during control during the trip or for interchanges, in place of EventContractPointer when more than one contract is related to the concerned event.

EventBestContractPriorityX

<i>Definition</i>	Priority of the contract.
<i>Use</i>	This field allows the terminal to only read valid contracts from the card, and to read them in a best priority order.

EventCode

<i>Definition</i>	Type of transport service at the event. The code expresses both the transport mode and the action or activity (e.g. entrance, exit).
<i>Use</i>	May be used for example at validation to check for interchanges. (Note that the control of a card during the trip does not have a code, because it is not recorded as an event.)

EventNetworkId

<i>Definition</i>	Digit identifying the region network ID in Portugal
<i>Use</i>	May be used for example at validation to check for interchanges.

EventServiceProvider

<i>Definition</i>	Identity of the company which provided the service related to the event. Coding according to <i>Company</i> .
<i>Use</i>	May be used for example at validation to check for interchanges.

EventLocationId

<i>Definition</i>	Identification of the point within the system of the service provider where the event concerned took place. For example, this may be a metro station entrance, or a vehicle Id, or a bus stop. The locations corresponding to each value are according to EventServiceProvider.
<i>Use</i>	May be used during control during the trip, for interchanges or to compute the price of a trip in a closed network system.

EventDevice

<i>Definition</i>	Identification of the specific equipment which wrote card data at the event concerned. This is a technical identification number and not to be confused with a location. Since the devices corresponding to each value are according to EventServiceProvider, the identification is only unique within the context of the service provider.
<i>Use</i>	May be used during control or audits.

EventRouteNumber

<i>Definition</i>	Reference to the actual route, including route variants, of the event concerned. This is for example the bus line identification. The routes corresponding to each value are according to EventServiceProvider.
<i>Use</i>	May be used during control during the trip or for interchanges.

EventJourneyRun

<i>Definition</i>	Number which identifies the run on which the event occurred. The runs corresponding to each value are according to EventServiceProvider.
<i>Use</i>	May be used to avoid multiple validations in the same run if the card is presented to a validator of the same bus again by mistake.

EventFirstServiceProvider

<i>Definition</i>	Service provider of first boarding.
<i>Use</i>	May be used during control during the trip or for interchanges.

EventFirstZoneMono and EventFirstZoneMonoMSB

<i>Definition</i>	Monomodal zone of first boarding. The zones corresponding to each value (except 255) are according to EventFirstServiceProvider. Zone number 255 is RFU.
<i>Use</i>	The 8-bit zone number is split as follows: <ul style="list-style-type: none">• 7 LSBits in EventFirstZoneMono• 1 MSbit in EventFirstZoneMonoMSB May be used during control during the trip or for interchanges.

EventFirstLineZone

<i>Definition</i>	Monomodal line of first boarding. The lines corresponding to each value are according to EventFirstServiceProvider.
<i>Use</i>	May be used during control during the trip or for interchanges.

EventProviderRFU

<i>Definition</i>	Reserved for future use.
<i>Use</i>	See section 6.1.4.

6.7 Special Event: Parking

The Special Event Structure describes the usage of a park & ride contracts at the parking operators terminals. It is relevant only if the in the contract referred to by the event is a park & ride contract.

Event Data Structure	Bits	Description	Type
SpecialEventVersionNumber	3	Special Event structure version number (0)	VersionNumber
SpecialEventTimeStamp	30	Parking event date and time	TimeReal
SpecialEventParkAreald	7	Parking area (or set of areas) identifier	Bits7
SpecialEventContractPointer	3	Number of the contract used for this event (cannot be 0)	Bits3
SpecialEventCode	8	Parking event code	Int1
SpecialEventNetworkIdNetwork	12	Porto Network Identifier within country	Bits12
SpecialEventNetworkIdCountry	12	Country ISO identification	Bits12
SpecialEventOperator	8	Operator (value according to SpecialEventNetworkId)	Company
SpecialEventLocationId	8	Location identifier within the parking area	Int1
SpecialEventDevice	8	Terminal identifier within the location	Int1
SpecialEventAmount	10	Amount of parking units for the event	Bits10
SpecialEventRFU	123	RFU	Bits123

The size of the special event structure is fixed: 232 bits (29 bytes).

SpecialEventVersionNumber

<i>Definition</i>	The version of the special event data structure defined in this document is version '0'. Note: In possible future version of the data structure, the version must not be greater than '011'b (3).
<i>Use</i>	The terminals must ignore a Special Event with a greater value of this field.

SpecialEventTimeStamp

<i>Definition</i>	Time at which the event took place.
<i>Use</i>	The date and time of the validation may be used during any parking related transaction, for example to record the entrance or exit time.

SpecialEventParkAreald

<i>Definition</i>	Identifier of the parking area where the event took place, as defined for SPDAreald.
<i>Use</i>	May be used during any parking related transaction.

SpecialEventContractPointer

<i>Definition</i>	Code referring to the contract related to the concerned event.
<i>Use</i>	May be used during any parking related transaction.

SpecialEventCode

<i>Definition</i>	Type of service at the event, according to SpecialEventOperator:
	0 RFU.
	1 Parking entrance.
	2 Parking exit.
	3 Park & ride contract issuing.
	≥4 RFU.
<i>Use</i>	May be used for example at entrance or exit.

SpecialEnvNetworkIdNetwork

<i>Definition</i>	This field is coded in BCD and contains the network identification number for the Porto region within Portugal.
<i>Use</i>	The terminals must reject applications with an unknown value of this field.

SpecialEnvNetworkIdCountry

<i>Definition</i>	This field is coded in BCD and takes the value 620h (ISO 3166-1 code for Portugal).
<i>Use</i>	The terminals must reject applications with an unknown value of this field.

SpecialEventOperator

<i>Definition</i>	Identifies the issuer who created the Special Event.
<i>Use</i>	The terminals should only record this field value during a transaction, and should not reject or modify the card transaction based on this field value.

SpecialEventLocationId

<i>Definition</i>	Identifies a specific location inside the parking area SpecialEventParkAreaId. The value 0 indicates that the location is undefined.
<i>Use</i>	May be used during any parking related transaction.

SpecialEventDevice

<i>Definition</i>	Identifies a specific terminal at the location SpecialEventLocationId (or inside the parking area, e.g. when the location is undefined). The value 0 indicates that the terminal is undefined.
<i>Use</i>	May be used during any parking related transaction.

SpecialEventAmount

<i>Definition</i>	Amount of parking units related to the event. If the event is a contract issuing (SpecialEventCode=4), SpecialEventAmount indicates the amount of parking units added, else it indicates the amount of parking units removed. If no amount is related to the event, SpecialEventAmount is set to 0.
<i>Use</i>	May be used during any parking related transaction.

SpecialEventRFU

<i>Definition</i>	Set to all zeroes.
<i>Use</i>	Ignored when read by terminals.

7 MANDATORY USAGE RULES

This chapter contains the rules that the terminals must follow.

These rules ensure interoperability among the terminals of the region.

7.1 Best Contract List Management

The Best Contract List is a central element of the ticketing card interoperability. Its purpose is:

- To accelerate the validation transaction, by indicating to the validator which contracts are valid, and by giving some information about the contract allowing a better probability to find the correct contract rapidly.
- To indicate to the selling machines how many contracts are possible in the card and which of them are empty or obsolete, and may be replaced by new ones.

The Best Contract List is created during personalization of the card, and is then managed by the reloading machines and the validators.

To make the transaction faster, the Best Contract List is included in the Event structure for every event written in the card.

Personalization:

During personalization, the Best Contract List is created to indicate the number of contracts in the card. All contracts are set to "erasable".

Reloading:

When a reloading machine must write a new contract in a card, it updates the Best Contract List to mark the obsolete contracts as invalid and erasable, and to include the newly loaded contracts. The reloading machine sets the priority of the newly loaded contracts.

Validation:

When a validator processes a card, it may update the Best Contract List to mark the obsolete contracts as invalid. This is the only operation authorized at validation time on the Best Contract List.

7.2 Available Files

Calypso defines another file in the Ticketing Application, the Contract List file. This file may be modified during validation, with the Debit Key (Key 3).

It usually contains complementary information (e.g. the Best Contract List if it does not fit within the Event log).

The present document does not define any use for this file.

7.3 Park & Ride Management

Parking terminals process the cards as follows:

1. Search for the best possible valid park & ride contract.
2. If a valid park & ride contract is found, process the entrance according to the contract found, writing the corresponding Special Event structure in the Special Event file.
3. Else, process the card as defined by the parking operator (for example as if the card was a contactless ticket⁵), using only the Special Event file.

The previous content of the Special Event file is ignored by the parking *entrance* terminals.

5 Note that in this case the contactless ticket data size should be at most 232 bits, and that a SAM is always necessary to write in a card.

8 RECOMMENDED USAGE RULES

This chapter describes recommended usage rules for managing the ticketing cards in the region. These rules are not mandatory, but are offered as guidelines to help understanding the ticketing management and interoperability.

8.1 Validator Transaction

The validator transaction must be as fast as possible to ease the card usage by the passenger.

Reading of all the card data must therefore be avoided during the validation.

The recommended algorithm for the transaction is as follows:

- Reading of the Environment/Holder file during the Open Session.
- Reading of the last event, which includes the Best Contract List.
- Reading of the best contract by using the Best Contract structure.
- Writing of the new event, with possible counter decrement and contract update.

During validation it is *never* useful to read more events than the last one. All necessary information to process the entrance is present in the last event, including the time of first validation during a multi-trip journey. Be careful that some cards may have only one, or three, events.

8.2 Best Contract List Management

To search for the contract allowing entrance:

The general algorithm is as follows:

1. Read the Best Contract List (from the last event).
2. For every valid contract in the Best Contract List, in the order of the priority indicated:
 - 2a. Read the contract.
 - 2b. If the contract has become obsolete, set its priority to the "Invalid" value ('D'h) and loop to the next contract.
 - 2c. If the contract does not allow entrance, loop to the next contract.
3. If a contract was found that allow entrance in the network, write a new event in the card, decrement a counter if necessary, and if a contract read was invalid, write the new Best Contract List.

Contract Deletion

The general algorithm is as follows:

1. Read the Best Contract List.
2. For every contract which is not marked "Erasable", nor marked "Absent" in the Best Contract List:
 - 2a. Read the contract.
 - 2b. If the contract is invalid, set its priority as "Erasable" ('E'h) in the Best Contract List.
3. Find the number (N) of a contract marked as erasable contract in the Best Contract List.
4. Set its priority according to the type of contract in the Best Contract List.
5. Write the contract and a new event with the new Best Contract List in the card.

ANEXO D “API Andante”



Transportes Intermodais do Porto

ANDANTE - Application Program Interface

ESPECIFICAÇÃO

Versão 01.24

outubro 2018

FICHA TÉCNICA

HISTÓRIA

Versão	Autor	Organização	Criação	Atualização
01.00P	Carlos Tomás	Octal		1ª versão. Documento preliminar.
01.00	Carlos Tomás	Octal	2004-01-29	1ª versão distribuída.
01.10	Henrique Parente	Octal	2006-01-12	Inclusão do CTS-512B; Atualização do zonamento (46 zonas).
01.20	Luís Faria	Novabase	2009-12-04	Revisão da definição de constantes; introdução do ClickZ.
01.21	José Barreira	Novabase	2010-08-25	Alterações na estrutura TYPE_TITLE_INFO por causa da introdução de novos cartões Mifare UL
01.22	José Cardoso	Novabase	2013-11-27	Adição dos operadores Pacense e Carvalhos
01.23	Bruno Silva	Novabase	2013-11-27	Adição do acesso aos cartões MIAFRE 1k
01.24	João Leitão	Novabase	2017-07-05	Novo código de erro de cartão invalidado. Operação de invalidação de suporte. Novas callbacks de escrita em zonas OTP para invalidação eletrónica de suportes CTS512B e MIFARE UL.

PROPRIEDADES

Responsável	Processador de texto
TIP	Microsoft Office Word
Gestor	Ficheiro
TIP	ANDANTE-API V1_24.DOCX
Nº. de palavras / Nº. de caracteres	Actualização
13 197 / 85 298	Pedro Castro Lopes
Palavras chave	Distribuição
API; Especificação	TIP

DOCUMENTOS RELACIONADOS

Título	Edição	Precedência	Conteúdo
-	-	-	-

CONVENÇÕES

Notação	Significado
Negrito	Títulos
Sublinhado	Parte de texto a realçar
Itálico	Definição de termos
Courier	Aspectos técnicos, como exemplos de utilização ou de programação
AAAA-MM-DD	Datas

Termo	Significado
ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
API	Application Program Interface
ATR	Answer To Reset
ISO	International Organization for Standardization
SAM	Secure Application Module
OTP	One Time Programmable

ÍNDICE

ÍNDICE GERAL

1. INTRODUÇÃO	11
1.1. FICHEIROS DA API	11
1.2. ARQUITECTURA DA API	11
2. INTERFACE APLICAÇÃO/API	13
2.1. CONSTANTES REFERENTES A CÓDIGOS DE ERRO	14
2.1.1. API ANDANTE	14
2.1.2. FUNÇÕES DE CALLBACK DE ACESSO AO LEITOR	15
2.2. ENUMERADOS	18
2.2.1. ENUM_CONFIG_PARAM	18
2.2.2. ENUM_CALLBACK_TYPE	19
2.2.3. ENUM_SAM_TYPE	21
2.2.4. ENUM_OPERATOR	21
2.2.5. ENUM_DURATION_UNIT	23
2.2.6. ENUM_TITLE	24
2.3. TIPOS PRÉ-DEFINIDOS	26
2.3.1. ANDANTE_HANDLER	26
2.3.2. TYPE_CALLBACK_FUNCTION	26
2.3.3. TYPE_TITLE_INFO	26
2.3.4. TYPE_CARD_ISSUE	30
2.3.5. TYPE_CONTRACT_LOAD	31
2.4. FUNÇÕES A DISPONIBILIZAR PELA APLICAÇÃO	34

2.4.1. "CALLBACKS" DE ACESSO AO SISTEMA OPERATIVO	34
2.4.1.1. "Callback" OS_Malloc()	34
2.4.1.2. "Callback" OS_Free()	35
2.4.2. "CALLBACKS" DE ACESSO AO SAM	36
2.4.2.1. "Callback" SAM_TxRxAPDU()	36
2.4.3. "CALLBACKS" DE ACESSO A UM CARTÃO DE MEMÓRIA	37
2.4.3.1. "Callback" CTS256B_Read()	37
2.4.3.2. "Callback" CTS256B_Update()	38
2.4.3.3. "Callback" CTS512B_Read()	39
2.4.3.4. "Callback" CTS512B_Update()	40
2.4.3.5. "Callback" CTS512B_Write()	41
2.4.3.6. "Callback" MFlight_Read()	41
2.4.3.7. "Callback" MFlight_Update()	42
2.4.3.8. "Callback" MFlight_Write()	43
2.4.4. "CALLBACKS" DE ACESSO A UM CARTÃO COM MICROPROCESSADOR	44
2.4.4.1. "Callback" Card_TxRxAPDU ()	44
2.4.4.2. "Callback" Card OpenSession()	45
2.4.4.3. "Callback" Card_CloseSession()	46
2.4.4.4. "Callback" Card UpdateRecord()	46
2.4.4.5. "Callback" Card AppendRecord()	48
2.4.4.6. "Callback" Card Decrease()	49
2.5. FUNÇÕES DISPONIBILIZADAS PELA API	51
2.5.1. FUNÇÕES DE GESTÃO DA API	51

2.5.1.1. Andante_NewAntenna()	51
2.5.1.2. Andante_DeleteAntenna()	52
2.5.1.3. Andante_NewSAM()	52
2.5.1.4. Andante_DeleteSAM()	53
2.5.1.5. Andante_NewCard()	53
2.5.1.6. Andante_DeleteCard()	54
2.5.1.7. Andante_SetCallback()	54
2.5.2. FUNÇÕES DE CONFIGURAÇÃO	55
2.5.2.1. Andante_SetConfigParam()	55
2.5.2.2. Andante_GetCardInfo()	56
2.5.2.3. Andante_SetSAM()	56
2.5.3. FUNÇÕES DE EXPLORAÇÃO	57
2.5.3.1. Andante_Read()	57
2.5.3.2. Andante_GetTitleInfo()	57
2.5.3.3. Andante_Load()	58
2.5.3.4. Andante_Validate()	59
2.5.3.5. Andante_Control()	60
2.5.3.6. Andante_Invalidate()	61
2.5.3.7. Andante_SaveData()	61
2.5.3.8. Andante_RestoreData()	62
2.5.3.9. Andante_FreeData()	63
2.5.4. EXTENÇÃO DAS FUNÇÕES DE EXPLORAÇÃO	63
2.5.4.1. Andante_Issue() Descrição:	63

2.5.4.2. Andante_ChangeProfiles()	64
3. ANEXO A – API ASK	66
3.1. INTRODUÇÃO	66
3.2. FICHEIROS DA API ASK	66
3.3. FUNÇÕES DE CALLBACK	66
3.3.1. DELAY	66
3.3.2. FLUSHTXRX	67
3.3.3. TX	67
3.3.4. RX	68
3.4. FUNÇÕES DA API	69
3.4.1. READASK_SETCALLBACK	69
3.4.2. READASK_NEW	70
3.4.3. READASK_DELETE	70
3.4.4. READASK_RESETREADER	71
3.4.5. READASK_RESETSAM	71
3.4.6. READASK_ANTENNAOFF	72
3.4.7. READASK_SEARCHCARD	73
3.4.8. ROTINAS DA API (CALLBACKS DA API ANDANTE)	74
3.4.8.1. Acesso ao SAM	74
3.4.8.1.1. READask_SAM_TxRxAPDU	74
3.4.8.2. Acesso a cartões de memória	75
3.4.8.2.1. READask_CTS256B_Read	75
3.4.8.2.2. READask_CTS256B_Update	76

3.4.8.2.3. READask_CTS512B_Read	77
3.4.8.2.4. READask_CTS512B_Update.....	77
3.4.8.2.5. READask_CTS512B_Write	78
3.4.8.2.6. READask_MFlight_Read	79
3.4.8.2.7. READask_MFlight_Update	80
3.4.8.2.8. READask_MFlight_Write	81
3.4.8.2.9. READask_MIFARE_LoadKey	81
3.4.8.2.10. READask_MIFARE_ChangeKey	82
3.4.8.2.11. READask_MIFARE_Authenticate	83
3.4.8.2.12. READask_MIFARE_ReadBlock	84
3.4.8.2.13. READask_MIFARE_WriteBlock	85
3.4.8.2.14. READask_MIFARE_IncrementValue	86
3.4.8.2.15. READask_MIFARE_DecrementValue	86
3.4.8.2.16. READask_MIFARE_CopyValue	87
3.4.8.2.17. READask_MIFARE_SimpleWriteBlock	88
3.4.8.2.18. READask_MIFARE_ReadSectorData	89
3.4.8.2.19. READask_MIFARE_WriteSectorData	89
3.4.8.3. Acesso a cartões com microprocessador	90
3.4.8.3.1. READask_CALYPSO_TxRxAPDU	90
3.4.8.3.2. READask_CALYPSO_OpenSession	91
3.4.8.3.3. READask_CALYPSO_CloseSession	92
3.4.8.3.4. READask_CALYPSO_UpdateRecord	93
3.4.8.3.5. READask_CALYPSO_AppendRecord	94

3.4.8.3.6. READask_CALYPSO_Decrease	95
3.4.8.3.7. READask_CALYPSO_ChangePIN	96
4. ANEXO B - EXEMPLO DE UTILIZAÇÃO	98

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

1. INTRODUÇÃO

Este documento descreve o modo de utilização da API de acesso ao Andante.

A construção desta API teve em conta os seguintes aspetos:

Só a API tem conhecimento do modelo de dados usado no Andante;

Os mecanismos de segurança de acesso ao título são da exclusiva responsabilidade da API;

Ser independente do tipo/marca do leitor usado, desde que este seja compatível com os cartões Andante usados.

Em anexo é descrita a utilização da API de acesso ao leitor da ASK. Esta API está concebida para fornecer as funções de acesso ao cartão/SAM necessárias ao funcionamento da API Andante.

1.1. FICHEIROS DA API

Os ficheiros necessários para a utilização desta API são os seguintes:

Andante.h: Ficheiro que contém os protótipos das funções e estruturas necessários à utilização da API;

Andante.lib: Biblioteca contendo o código objeto correspondente à API que permite o acesso ao modelo de dados do Andante. Esta biblioteca será construída de acordo com o ambiente de desenvolvimento usado no equipamento onde será usada a API. A biblioteca está construída em ANSI C.

1.2. ARQUITECTURA DA API

Para facilitar e simplificar a compreensão de como utilizar a API, designar-se-á Aplicação ao software que usa os serviços da API. É da responsabilidade da aplicação:

Fornecer as funções de acesso de baixo nível ao leitor(es), ao seu(s) módulos de segurança e aos cartões de acordo com as regras estipuladas por esta API;

Detetar a presença de cartão no leitor, passando à API o resultado do ATR;

Guardar quaisquer parâmetros ou outros dados de carácter não volátil necessários ao funcionamento do equipamento;

Fornecer à API os dados necessários à codificação do cartão Andante; É da responsabilidade da API:

Mediante os parâmetros fornecidos pela aplicação, interpretar a informação existente no cartão Andante, tendo em conta o tipo de cartão e os modelos de dados em vigor;

Mediante os parâmetros fornecidos pela aplicação, codificar num Andante um título de transporte válido, tendo em conta os modelos de dados definidos;

Chamar as funções necessárias, e disponibilizadas pela Aplicação, que permitem o acesso de baixo nível ao leitor/leitores, módulo(s) de segurança e cartão sempre que for necessário;

Executar as funções de segurança necessárias de acesso ao cartão;

2. INTERFACE APLICAÇÃO/API

A API Andante contém, essencialmente, três categorias de funções:

Inicialização/Gestão;

Configuração;

Exploração.

Na categoria de Inicialização/Gestão existem funções que permitem a alocação e inicialização em memória dos objetos que são necessários ao funcionamento da API. Servem também o propósito de indicar à API qual o conjunto de funções, disponibilizadas pela Aplicação, que irão permitir o acesso aos seguintes dispositivos, ou funcionalidades:

Funções de acesso aos seguintes dispositivos:

Leitor de cartões sem contacto;

Módulo ou módulos de segurança;

Cartão sem contacto;

Funções dependentes do sistema operativo:

Alocação/libertação de memória;

Formatação de texto;

Estas funções que a API deve chamar para aceder a estes dispositivos ou funcionalidades dependentes do sistema operativo devem ser disponibilizadas pela Aplicação e serão chamadas pela API, quando delas necessite, através do mecanismo de “callback”. A API, através da utilização de uma das suas funções, permite à Aplicação, definir quais as funções onde reside a implementação das mesmas. É obrigatório, por parte da Aplicação, definir todas as funções de “callback”.

De referir que se o leitor de cartões for o da ASK, poderá ser fornecida uma API de acesso ao leitor (READAsk), não sendo necessário ao integrador implementar todas as funções correspondentes às funções de “callback” exigidas pela API Andante. O modo como se integra e usa a READAsk consta de um anexo a este documento.

Na categoria de funções de Configuração, a API permite à Aplicação definir um conjunto de dados que servem para informar a API qual o ambiente onde o equipamento se encontra assim como de informar a API qual o módulo de segurança que será usado aquando do acesso a um cartão Andante. Na terceira categoria de funções, Exploração, reside a implementação das regras que regem a utilização do cartão Andante, por parte dos equipamentos responsáveis pelo seu manuseamento. Segue-se, nos subcapítulos seguintes a descrição das funções que podem ser usadas na API do Andante, assim como de outros objetos necessários à sua utilização.

2.1. CONSTANTES REFERENTES A CÓDIGOS DE ERRO

2.1.1. API ANDANTE

ANDANTE_ERROR_CARD_UNKNOWN	Tipo de cartão desconhecido
ANDANTE_ERROR_NO_RELOAD_CHANGE	Alteração do título não permitida
ANDANTE_ERROR_MAX_RELOAD	Número máximo de carregamentos atingido
ANDANTE_ERROR_SAM	Erro no SAM
ANDANTE_ERROR_CARD_SERIALNUMBER	Número de série do cartão errado
ANDANTE_ERROR_PIN	PIN incorreto
ANDANTE_ERROR_UPDATE	Erro de escrita no cartão
ANDANTE_ERROR_DATAMODEL_UNKNOWN	Modelo de dados do cartão desconhecido
ANDANTE_ERROR_CARD_DATA	Dados do cartão incorretos
ANDANTE_CARD_EMPTY	Cartão não inicializado (virgem)
ANDANTE_ERROR_RESTORABLE	Título recuperável
ANDANTE_ERROR_NO_INFO	Dados inexistentes
ANDANTE_MISSING_OPERATOR	Parâmetro de configuração do operador não definido
ANDANTE_MISSING_MACHINE	Parâmetro de configuração do equipamento não definido
ANDANTE_MISSING_CONTRACT	Contrato inexistente
ANDANTE_MISSING_ZONE	Parâmetro de configuração da zona não definido
ANDANTE_MISSING_LOCATION	Parâmetro de configuração do local não definido
ANDANTE_ERROR_CARD_EXPIRED	Cartão expirado
ANDANTE_MISSING_CONFIG_PARAM	Parâmetro de configuração não definido
ANDANTE_ERROR_CARD_INVALIDATED	Cartão invalidado
ANDANTE_CONTROL_NO_MORE_TRIPS	Viagens esgotadas
ANDANTE_CONTROL_OUT_OF_DATE	Título fora de validade
ANDANTE_CONTROL_INV_DATE	Data inválida
ANDANTE_CONTROL_FUTURE_DATE	Dados no futuro
ANDANTE_CONTROL_INV_PROVIDER	Operador inválido

ANDANTE_CONTROL_INV_TARIFF	Tipo de tarifa inválida
ANDANTE_CONTROL_INV_ZONE	Zona inválida
ANDANTE_CONTROL_NOT_VALID_YET	Ainda não é válido
ANDANTE_CONTROL_BLACK_LIST	Cartão em lista negra
ANDANTE_CONTROL_OUT_OF_HOUR	Horário inválido
ANDANTE_CONTROL_NOT_VALIDATED	Cartão não validado
ANDANTE_CONTROL_EXPIRED_TRIP	Limite de tempo ultrapassado
ANDANTE_CONTROL_WORKER	Cartão de funcionário
ANDANTE_ERROR_NOT_ALLOWED	Função não autorizada
ANDANTE_ERROR_READ	Erro na leitura do cartão
ANDANTE_ERROR_CALLBACK_MISSING	Função de callback não definida
ANDANTE_ERROR_HANDLER	Handler incorreto
ANDANTE_ERROR_API	Erro genérico da API

2.1.2. FUNÇÕES DE CALLBACK DE ACESSO AO LEITOR

READASK_ERROR_RCSC_OpenCOM	Erro na inicialização da comunicação com o coupler
READASK_ERROR_RCSC_NoAnswer	Ausência de resposta do coupler
READASK_ERROR_RCSC_CheckSum	Resposta do coupler com checksum incorreto
READASK_ERROR_RCSC_Fail	Parâmetros de chamada a uma função incorretos
READASK_ERROR_RCSC_Timeout	Timeout na espera de uma resposta do coupler
READASK_ERROR_RCSC_DataWrong	Resposta incorreta do coupler
READASK_ERROR_RCSC_Overflow	Dimensão da resposta demasiado alta
READASK_ERROR_RCSC_ErrorSAM	Falha no acesso ao SAM
READASK_ERROR_RCSC_TXError	Falha no envio de um comando para o coupler
READASK_ERROR_RCSC_UnkClass	Parâmetro CLA incorreto na resposta do coupler
READASK_ERROR_COLLISION	Demasiados cartões detetados pela antena
READASK_ERROR_INVALID_HANDLER	Handler do coupler incorreto
READASK_ERROR_UNEXPECTED_PROT	Tipo de cartão (protocolo) inesperado

READASK_ERROR_TIMEOUT	Nenhum cartão detetado pela antena
READASK_ERROR_CALLBACK	Função de callback indefinida
READASK_ERROR_API	Erro não definido da API do coupler
READASK_ERROR_STA_FILE_STRUCT	Comando incompatível com a estrutura do ficheiro
READASK_ERROR_STA_FILE_NOT_FOUND	Ficheiro não encontrado
READASK_ERROR_STA_REC_NOT_FOUND	Registo não encontrado
READASK_ERROR_STA_SEC_NOT_RESP	Condições de segurança não respeitadas
READASK_ERROR_STA_SEC_INV_SIGN_0	PIN incorreto – 0 tentativas ainda autorizadas
READASK_ERROR_STA_SEC_INV_SIGN_1	PIN incorreto – 1 tentativa ainda autorizadas
READASK_ERROR_STA_SEC_INV_SIGN_2	PIN incorreto – 2 tentativas ainda autorizadas
READASK_ERROR_STA_SEC_REJECTED	Número máximo de erros (3) atingido
READASK_ERROR_STA_SESSION_EEPROM	Capacidade da EEPROM insuficiente para a sessão
READASK_ERROR_STA_SESSION_FORBID	Comando não permitido
READASK_ERROR_STA_SESSION_PROHIB	Comando não autorizado num DF
READASK_ERROR_STA_BADGE_CORRUPT	Dados retornados pelo cartão corrompidos
READASK_ERROR_STA_BADGE_EEPROM	Falha na EEPROM do cartão
READASK_ERROR_STA_APP_INV_VALUE	Valor a subtrair ou adicionar incorreto
READASK_ERROR_STA_APP_INV_KEY	Nível da chave especificada incorreto
READASK_ERROR_STA_APP_INV_LC	Parâmetro Lc incompatível com os parâmetros P1-P2
READASK_ERROR_STA_APP_INV_P3	Parâmetro P3 inválido
READASK_ERROR_STA_APP_INV_P1_P2	Parâmetros P1-P2 não suportados
READASK_ERROR_STA_APP_INV_INS	Parâmetro INS não suportado
READASK_ERROR_STA_APP_INV_CLA	Parâmetro CLA não suportado
READASK_ERROR_STA_CARD_INV	DF superior ou MF invalidado
READASK_ERROR_STA_SAM_NOT_AUTH1	Comando para o SAM não autorizado – 1
READASK_ERROR_STA_SAM_NOT_AUTH2	Comando para o SAM não autorizado – 2
READASK_ERROR_STA_SAM_INV_CRYPTO	Criptograma incorreto
READASK_ERROR_STA_SAM_EEPROM	Falha na EEPROM do SAM

READASK_ERROR_STA_SAM_INV_VALUE	Resposta do SAM com valor incorreto
READASK_ERROR_STA_SAM_EXCEEDED	Capacidade do PME do cartão excedida
READASK_ERROR_STA_SAM_REC_NFOUND	Registo do SAM não encontrado
READASK_ERROR_STA_SAM_INV_P1_P2	Parâmetros P1-P2 incorretos
READASK_ERROR_STA_SAM_INV_LC	Parâmetro Lc incompatível com os parâmetros P1-P2
READASK_ERROR_STA_SAM_EXEC	Falha na execução do comando para o SAM
READASK_ERROR_STA_CSC_UNK	Comando desconhecido
READASK_ERROR_STA_CSC_CONCAT	Concatenação não autorizada para a função
READASK_ERROR_STA_CSC_MAX_CONCAT	Número máximo de funções concatenáveis atingido
READASK_ERROR_STA_CSC_ACC_CONCAT	Modo de acesso incompatível com a concatenação
READASK_ERROR_STA_CSC_ACC_FUNC	Modo de acesso não permitido para a função
READASK_ERROR_STA_CSC_NO_SESSION	Não existe nenhuma sessão aberta
READASK_ERROR_STA_CSC_SESS_OPEN	Sessão já aberta
READASK_ERROR_STA_CSC_INV_RESP	Resposta do cartão incorreta
READASK_ERROR_STA_CSC_PROHIBITED	Comando não permitido na sessão
READASK_ERROR_STA_CSC_OVERFLOW	Pacote de transmissão para o cartão demasiado longo
READASK_ERROR_STA_CSC_INV_KEY	Chave do SAM não encontrada
READASK_ERROR_STA_CSC_KEY_SEL	ChangeKey não permitida antes de um SelectFile
READASK_ERROR_STA_CSC_KEY_VERS	ChangeKey não permitida quando a nova chave é 0
READASK_ERROR_STA_CSC_KEY_FAILED	ChangeKey com uma versão de chave >
READASK_ERROR_STA_API	Erro de status não definido

2.2. ENUMERADOS

2.2.1. ENUM_CONFIG_PARAM

Descrição:

Enumera os parâmetros que podem ser configurados através da chamada à função Andante_SetConfigParam.

Definição:

```
typedef enum {  
  
    CONFIG_PARAM_OPERATOR,  
    CONFIG_PARAM_ZONE,  
  
    CONFIG_PARAM_LOCATION,  
  
    CONFIG_PARAM_MACHINE,  
  
    CONFIG_PARAM_SERIALNUMBER,  
    CONFIG_PARAM_LINE,  
  
    CONFIG_PARAM_SAEB_SENTIDO,  
  
    CONFIG_PARAM_TRIP_START,  
  
    CONFIG_PARAM_VEHICLE,  
  
    CONFIG_PARAM_SW_VERSION  
  
} ENUM_CONFIG_PARAM;
```

Valores possíveis:

CONFIG_PARAM_OPERATOR	Define o código de operador de transporte que está associado ao equipamento.
CONFIG_PARAM_ZONE	Define a zona tarifária onde está situado o equipamento. Sempre que o código da zona mude, a aplicação deverá chamar a função Andante_SetConfigParam(), informando qual o novo código de zona.
CONFIG_PARAM_LOCATION	Define o local onde está situado o equipamento. Sempre que o código do local mude, a aplicação deverá chamar a função Andante_SetConfigParam(), informando qual o novo código de local.
CONFIG_PARAM_MACHINE	Define a identificação do equipamento
CONFIG_PARAM_SERIALNUMBER	Define o número de série associado ao equipamento.
CONFIG_PARAM_LINE	Define linha em transporte rodoviário
CONFIG_PARAM_SAEB_SENTIDO	Define sentido da linha

CONFIG_PARAM_TRIP_START	Define data de início da viagem
CONFIG_PARAM_VEHICLE	Define o número de viatura
CONFIG_PARAM_SW_VERSION	Define a versão de software da aplicação.

Ver também:

`Andante_SetConfigParam()`

2.2.2. ENUM CALLBACK TYPE

Descrição:

Enumera todos os tipos de funções de callback que podem ser chamadas pela API Andante, e que devem ser implementadas pela Aplicação.

Definição:

```
typedef enum {  
CALLBACK_OS_MALLOC,  
  
    CALLBACK_OS_FREE,  
  
    CALLBACK_OS_SPRINTF,  
  
    CALLBACK_SAM_TPDU,  
  
    CALLBACK_CTS256B_READ,  
  
    CALLBACK_CTS256B_UPDATE,  
CALLBACK_CTS512B_READ,  
  
    CALLBACK_CTS512B_UPDATE,  
  
    CALLBACK_CTS512B_WRITE,  
  
    CALLBACK_MFLIGHT_READ,  
  
    CALLBACK_MFLIGHT_UPDATE,  
  
    CALLBACK_MFLIGHT_WRITE,  
  
    CALLBACK_CALYPSO_TPDU,  
  
    CALLBACK_CALYPSO_OPEN,  
  
    CALLBACK_CALYPSO_CLOSE,  
  
    CALLBACK_CALYPSO_UPDATE_REC,  
  
    CALLBACK_CALYPSO_APPEND_REC,  
}
```

```
    CALLBACK_CALYPSO_DECREASE,  
  
    CALLBACK_CALYPSO_CHANGEPIN  
} ENUM_CALLBACK_TYPE;
```

Valores possíveis:

<code>CALLBACK_OS_MALLOC</code>	Função para alocação de memória dinâmica (malloc em ANSI-C).
<code>CALLBACK_OS_FREE</code>	Função para libertação de memória dinâmica (free em ANSI-C).
<code>CALLBACK_OS_SPRINTF</code>	Função para formatar texto (sprintf em ANSI-C).
<code>CALLBACK_SAM_TPDU</code>	Função para comunicação ISO com o SAM.
<code>CALLBACK_CTS256B_READ</code>	Função para ler um cartão CTS256B.
<code>CALLBACK_CTS256B_UPDATE</code>	Função para escrever num cartão CTS256B.
<code>CALLBACK_CTS512B_READ</code>	Função para ler um cartão CTS512B.
<code>CALLBACK_CTS512B_UPDATE</code>	Função para escrever em zonas de memória num cartão CTS512B.
<code>CALLBACK_CTS512B_WRITE</code>	Função para escrever em zonas OTP num cartão CTS512B.
<code>CALLBACK_MFLIGHT_READ</code>	Função para ler um cartão MIFARE Ultralight.
<code>CALLBACK_MFLIGHT_UPDATE</code>	Função para escrever em zonas de memória num cartão MIFARE Ultralight.
<code>CALLBACK_MFLIGHT_WRITE</code>	Função para escrever em zonas OTP num cartão MIFARE Ultralight.
<code>CALLBACK_CALYPSO_TPDU</code>	Função para comunicação ISO com um cartão CALYPSO.
<code>CALLBACK_CALYPSO_OPEN</code>	Função para abrir uma sessão segura com um cartão CALYPSO.
<code>CALLBACK_CALYPSO_CLOSE</code>	Função para fechar uma sessão segura com um cartão CALYPSO.
<code>CALLBACK_CALYPSO_UPDATE_REC</code>	Função para atualização de registos num cartão CALYPSO.
<code>CALLBACK_CALYPSO_APPEND_REC</code>	Função para adição de registos num cartão CALYPSO.
<code>CALLBACK_CALYPSO_DECREASE</code>	Função para decrementar contadores num cartão CALYPSO.
<code>CALLBACK_CALYPSO_CHANGEPIN</code>	Função para mudança do PIN num cartão CALYPSO.

Ver também:

`Andante_SetCallback()`

2.2.3. ENUM SAM TYPE

Descrição:

Enumera os tipos de SAM que podem ser usados nos equipamentos.

Definição:

```
typedef enum {  
    ANDANTE_SAM_CP,  
  
    ANDANTE_SAM_CL,  
  
    ANDANTE_SAM_CV  
} ENUM_SAM_TYPE;
```

Valores possíveis:

ANDANTE_SAM_CP	Indica que o equipamento possui um SAM de personalização (CP).
ANDANTE_SAM_CL	Indica que o equipamento possui um SAM de carregamento (CL).
ANDANTE_SAM_CV	Indica que o equipamento possui um SAM de validação ou fiscalização (CV).

Ver também:

Andante_NewSAM()

2.2.4. ENUM OPERATOR

Descrição:

Enumera os operadores de transporte que se podem associar aos equipamentos.

Definição:

```
typedef enum {  
    PT_OPER_TIP = 1,  
  
    PT_OPER_METRO,  
  
    PT_OPER_STCP,  
  
    PT_OPER_CP,  
  
    PT_OPER_RESENDE,  
  
    PT_OPER_CP_PORTO,
```

```
PT_OPER_CP_LISBOA,  
  
PT_OPER_CP_REGIONAL,  
  
PT_OPER_GUINDAIS,  
  
PT_OPER_VALPI = 11,  
  
PT_OPER_ESPIRITOSANTO,  
  
PT_OPER_GONDOMARENSE,  
  
PT_OPER_TRANSDEV,  
  
PT_OPER_MAIA,  
  
PT_OPER_OFRIBEIRO,  
  
PT_OPER_ARRIVA,  
  
PT_OPER_PACENSE,  
  
PT_OPER_NOGUEIRA_DA_COSTA,  
  
PT_OPER_COSTA_REIS,  
  
PT_OPER_LANDIM,  
  
PT_OPER_RODONORTE,  
  
PT_OPER_CARVALHOS  
  
} ENUM_OPERATOR;
```

Valores possíveis:

PT_OPER_TIP	Transportes Intermodais do Porto, ACE.
PT_OPER_METRO	Metro do Porto, SA.
PT_OPER_STCP	Sociedade de Transportes Colectivos do Porto, SA.
PT_OPER_CP	Caminhos de Ferro Portugueses, EP
PT_OPER_RESENDE	Resende – Actividades Turísticas, SA.
PT_OPER_CP_PORTO	CP – Urbanos / Porto
PT_OPER_CP_LISBOA	CP – Urbanos / Lisboa
PT_OPER_CP_REGIONAL	CP - Regional
PT_OPER_GUINDAIS	Funicular dos Guindais
PT_OPER_VALPI	Valpi Bus – Alberto Pinto & Filhos - Transportes Rodoviários SA.

PT_OPER_ESPIRITOSANTO	J. Espírito Santo & Irmãos, Lda.
PT_OPER_GONDOMARENSE	ETG – Empresa de Transportes Gondomarense, Lda.
PT_OPER_TRANSDEV	Transdev (Portugal) Transportes, Lda.
PT_OPER_MAIA	Maia Transportes
PT_OPER_OFRIBEIRO	Oliveira, Fernandes & Ribeiro, Lda.
PT_OPER_ARRIVA	Arriva Portugal – Transportes, Lda.
PT_OPER_PACENSE	Auto-Viação Pacense, Lda
PT_OPER_NOGUEIRA_DA_COSTA	A. Nogueira da Costa, Lda
PT_OPER_COSTA_REIS	A. Costa Reis & Filhos, Lda
PT_OPER_LANDIM	Auto-Viação Landim, Lda
PT_OPER_RODONORTE	Rodonorte – Transportes Portugueses, SA
PT_OPER_CARVALHOS	União de Transportes dos Carvalhos

Ver também:

`Andante_SetConfigParam()`

2.2.5. ENUM_DURATION_UNIT

Descrição:

Enumera unidades de tempo de duração.

Definição:

```
typedef enum {  
    DURATION_UNITS_YEARS,  
  
    DURATION_UNITS_MONTHS,  
  
    DURATION_UNITS_WEEKS,  
    DURATION_UNITS_DAYS,  
  
    DURATION_UNITS_HOURS,  
  
    DURATION_UNITS_HALF_HOURS,  
  
    DURATION_UNITS_MINUTES,  
  
    DURATION_UNITS_SECONDS  
}
```

```
} ENUM_DURATION_UNIT;
```

Valores possíveis:

DURATION_UNITS_YEARS	Duração em anos.
DURATION_UNITS_MONTH	Duração em meses.
DURATION_UNITS_WEEKS	Duração em semanas.
DURATION_UNITS_DAYS	Duração em dias.
DURATION_UNITS_HOURS	Duração em horas.
DURATION_UNITS_HALF_HOURS	Duração em ½'s horas.
DURATION_UNITS_MINUTES	Duração em minutos.
DURATION_UNITS_SECONDS	Duração em segundos.

Ver também:

```
TYPE_TITLE_INFO, TYPE_CONTRACT_LOAD, Andante_GetTitleInfo(), Andante_Load()
```

2.2.6. ENUM_TITLE

Descrição:

Enumera os tipos de título passíveis de serem carregados no Andante.

Definição:

```
typedef enum {  
  
    ANDANTE_TITLE_MULTITRIP,  
  
    ANDANTE_TITLE_SEASON,  
  
    ANDANTE_TITLE_MULTITRIP_24,  
  
    ANDANTE_TITLE_SEASON_10_16  
  
} ENUM_TITLE;
```

Valores possíveis:

ANDANTE_TITLE_MULTITRIP	Andante multiviagem
-------------------------	---------------------

ANDANTE_TITLE_SEASON	Andante assinatura
ANDANTE_TITLE_MULTITRIP_24	Andante 24 horas
ANDANTE_TITLE_SEASON_10_16	Andante 10-16

Ver também:

Andante_Load()

2.3. TIPOS PRÉ-DEFINIDOS

2.3.1. ANDANTE_HANDLER

Descrição:

Todos os dispositivos com que a API trabalha (antena, SAM e cartão) tem uma identificação interna associada: Handler. A atribuição de um handler a um determinado dispositivo é efetuada pela API, sempre que a aplicação chama a função de inicialização do dispositivo (Andante_NewAntenna(), Andante_NewSAM() e Andante_NewCard()). Este handler será depois usado pela aplicação sempre que chamar funções da API que lidem com os dispositivos.

Definição:

```
typedef void *ANDANTE_HANDLER;
```

2.3.2. TYPE_CALLBACK_FUNCTION

Descrição:

Definição comum (genérica) de todas as funções de callback

Definição:

```
typedef unsigned long (*TYPE_CALLBACK_FUNCTION)(long Param, ...);
```

2.3.3. TYPE_TITLE_INFO

Descrição:

Estrutura que serve de repositório aos dados lidos do cartão Andante.

Definição:

```
typedef struct { /* Flags */
    short      IsEmpty;    short
    IsConsumed;  short
    CanChangeOperator;  short
    CanReload; /* User */  short
    Layout;  short      Profile1;
    short      Profile1Year;
    short      Profile1Month;
    short      Profile1Day;
    short      Profile2;  short
    Profile2Year;  short
    Profile2Month;  short
    Profile2Day;  short
```

```

Profile3;    short
Profile3Year;    short
Profile3Month;    short
Profile3Day;    short
CardIssuer;    unsigned long
CardNumber;    unsigned long
CardExpire;    short
Company;    unsigned long
OwnerNumber;    short
ReductionRate;    short
BirthDateYear;    short
BirthDateMonth;    short
BirthDateDay;    /* */    short
Multimodal;    short
Operator;    short        Title;
unsigned long ContractId[2];    /*
Multitrip */    short
Zones;    long        Journeys;
long        JourneysLeft;
long        MaxJourneys;
unsigned long TripFinal;    /*
Season */    unsigned long
ZoneMask[3];    unsigned long
DateInitial;    unsigned long
Duration;    char
DurationUnits;    /* Validation */
/* First Validation */    unsigned
long FirstTimestamp;    short
FirstOperator;    short
FirstZone;

    short        FirstLocation;
/* Last Validation */
unsigned long LastTimestamp;
short        LastOperator;
short        LastZone;
short        LastLocation;
short        EventType;

    /* */    short
Priority;    short
EmbedderCode;

} TYPE_TITLE_INFO;
    
```

Membros:

Short	IsEmpty	Indica se o registo do contrato está vazio
short	IsConsumed	Indica que não tem saldo e já terminou a viagem

short	CanChangeOperator	Indica se o operador do contrato pode ser alterado
short	CanReload	Indica se o contrato pode ser recarregado
short	Layout	Layout do cartão (utilizado para os cartões Mifare UL)
short	Profile1	Código de perfil 1 do utente.
short	Profile1Year	Ano de expiração do perfil 1 (Profile1)
short	Profile1Month	Mês de expiração do perfil 1 (Profile1)
short	Profile1Day	Dia de expiração do perfil 1 (Profile1)
short	Profile2	Código do perfil 2 do utente
short	Profile2Year	Ano de expiração do perfil 2 (Profile2)
short	Profile2Month	Mês de expiração do perfil 2 (Profile2)
short	Profile2Day	Dia de expiração do perfil 2 (Profile2)
short	Profile3	Código do perfil 3 do utente
short	Profile3Year	Ano de expiração do perfil 3 (Profile3)
short	Profile3Month	Mês de expiração do perfil 3 (Profile3)
short	Profile3Day	Dia de expiração do perfil 3 (Profile3)
short	CardIssuer	Emissor do título
unsigned long	CardNumber	Número do cartão
unsigned long	CardExpire	Data em que expira o cartão (segundos desde 1970/01/01 00:00:00)
short	Company	Código da empresa
unsigned long	OwnerNumber	Identificação numérica do possuidor do cartão
short	ReductionRate	Taxa de desconto
short	BirthDateYear	Ano de nascimento
short	BirthDateMonth	Mês de nascimento
short	BirthDateDay	Dia de nascimento
short	Multimodal	Indica se o título é multimodal ou monomodal
short	Operator	Código do operador (de transportes)
short	Title	Código do título

unsigned long	ContractId[2]	Não usado
short	Zones	Nº de zonas para o caso do um título multiviagem
long	Journeys	Nº total de viagens no caso (título multiviagem)
long	JourneyLeft	Nº de viagens por utilizar (título multiviagem)
long	MaxJourneys	Nº máximo de viagens carregáveis (título multiviagem)
unsigned long	TripFinal	Data/Hora do fim da viagem (formato: segundos desde <1970-01-01 00:00:00>)
unsigned long [3]	ZoneMask	Máscara de zonas (título temporal)
unsigned long	DateInitial	Data de validade inicial (título temporal) (formato: segundos desde <1970-01-01 00:00:00>)
unsigned long	Duration	Duração do contrato (contado a partir de DateInitial). As unidades dependem do campo DurationUnits
char	DurationUnits	Identifica a unidade de tempo usada nos campos FirstTimestamp e LastTimestamp.
unsigned long	FirstTimestamp	Data e hora da 1ª validação
short	FirstOperator	Operador onde foi efetuada a 1ª validação.
short	FirstZone	Zona tarifária onde foi efetuada a 1ª validação
short	FirstLocation	Local onde foi efetuada a 1ª validação.
unsigned long	LastTimestamp	Data e hora da última validação
short	LastOperator	Operador onde foi efetuada a última validação.
short	LastZone	Zona tarifária onde foi efetuada a última validação
short	LastLocation	Local onde foi efetuada a última validação.
short	EventType	Tipo de evento da última validação.
short	Priority	Indica se o contrato é um complemento (Clikz)
short	EmbedderCode	Código de fabricante para os cartões Mifare UL

Ver também:

2.3.4. TYPE CARD ISSUE

Descrição:

Estrutura de dados que a Aplicação deve preencher, antes da chamada à função `Andante_Issue()`, de modo a efetuar a personalização de um cartão.

Definição:

```
typedef struct TAG_STRUCT_CARD_ISSUE {
short EnvApplicationVersionNumber;  short
Layout;  char Name[17 + 17 + 1];  long
CardNumber;  long ExpireDate;  struct
TAG_STRUCT_CARD_ISSUE_BIRTHDATE {  short
Year;  short Month;  short Day;  }
BirthDate;  short ReductionRate;  short
Company;  long IdNumber;  struct
TAG_STRUCT_PROFILE {  short Code;  long
Date;  } Profile[4];  long Residence;
long StudyPlace;  long SaleDevice;
long NIF;

} TYPE_CARD_ISSUE;
```

Membros:

short	EnvApplicationVersionNumber	
short	Layout	Layout do cartão
char[35]	Name	Nome do utente
long	CardNumber	Número do cartão
long	ExpireDate	Data de validade do cartão (segundos desde <1970-01-01 00:00:00>).
short	BirthDate.Year	Ano da data de nascimento
short	BirthDate.Month	Mês da data de nascimento
short	BirthDate.Day	Dia da data de nascimento
short	ReductionRate	Percentagem de desconto (Andante Pro)
short	Company	Código da empresa (Andante Pro)
long	IdNumber	Número da identificação do utente
short	Profile.Code	Código do perfil do cartão
long	Profile.Date	Data de validade do cartão (segundos desde <1970-01-01 00:00:00>).

long	Residence	Morada
long	StudyPlace	Morada do estabelecimento de ensino
long	SaleDevice	Código de equipamento onde foi personalizado o cartão
long	NIF	NIF do utente

2.3.5. TYPE CONTRACT LOAD

Descrição:

Estrutura de dados que a Aplicação deve preencher, antes da chamada à função Andante_Load(), de modo a efetuar o carregamento de um título de transporte num cartão Andante.

Definição:

```
typedef struct {          ENUM_TITLE
Type; short Multimodal; short
Profile; short ReloadCount; /*
Output */ union {        struct {
short Journeys;          short
NumZones;                } Multitrip;
struct {                  short Journeys;
short NumZones;          } Multitrip24;
struct {                  unsigned long Date;
short Duration;          char
DurationUnits;           unsigned long
ZoneMask[3];

        } Season;        struct {
unsigned long Date;      short
Duration;                char
DurationUnits;           unsigned
long ZoneMask[3];

        } Season1016;

    } Data;
} TYPE_CONTRACT_LOAD;
```

Membros:

ENUM_TITLE	Type	Tipo de título de transporte a carregar.
short	Multimodal	0: Título monomodal; 1: Título intermodal
short	Profile	Perfil.

short	ReloadCount	Contador de carregamentos no cartão. É a API que preenche este valor depois de proceder ao carregamento de um título de transporte.
short	Data.Multitrip.Journeys	Número de viagens a carregar (títulos multiviagem).
short	Data.Multitrip.NumZones	Número de zonas a carregar (títulos multiviagem).
short	Data.Multitrip24.Journeys	Número de viagens a carregar (títulos multiviagem 24 horas).
short	Data.Multitrip24.NumZones	Número de zonas a carregar (títulos multiviagem 24 horas).
unsigned long	Data.Season.Date	Início de validade das assinaturas. (nº de segundos desde 1970/01/01).
short	Data.Season.Duration	Duração da validade das assinaturas
char	Data.Season.DurationUnits	Unidades de Data.Season.Duration
unsigned long [3]	Data.Season.ZoneMask	Máscara de zonas onde a assinatura é válida (*). O bit menos significativo corresponde à Zona A.
unsigned long	Data.Season1016.Date	Início de validade das assinaturas 1016. (nº de segundos desde 1970/01/01).
short	Data.Season1016.Duration	Duração da validade das assinaturas 10-16
char	Data.Season1016.DurationUnits	Unidades de Data.Season1016.Duration
unsigned long [3]	Data.Season1016.ZoneMask	Máscara de zonas onde a assinatura 10-16 é válida (*).

(*) Máscara de zonas (correspondência do bit de maior peso para o bit menos significativo, até ao bit com índice 63, inclusive; as zonas identificadas com prefixo 'z' não se encontram atualmente definida mas serão definidas através de ficheiros de configuração nos sistemas aplicáveis):

ZoneMask[0] N16 N15 N14 N13 N12 N11 N10 N9 N8 N7 N6 N5 N4 N3 N2 N1
C16 C15 C14 C13 C12 C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1

ZoneMask[1]	---	Z63	Z62	Z61	Z60	Z59	Z58	Z57	Z56	Z55	Z54	Z53	Z52	Z51	Z50	Z49	
		Z48	N18	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	N17
ZoneMask[2]	---																

Ver também:

[Andante_Load\(\)](#)

2.4. FUNÇÕES A DISPONIBILIZAR PELA APLICAÇÃO

Tal como já foi referido, a API Andante necessita, por parte da Aplicação, a definição e implementação de um conjunto de funções que permitam o acesso aos seguintes recursos:

Funções que dependem do Sistema Operativo;

Funções de acesso ao leitor de cartões;

Funções de acesso ao módulo de segurança (SAM).

De referir que dependendo do leitor usado, estas funções podem ser fornecidas em conjunto com a API Andante. Nomeadamente, caso o leitor seja o da ASK as funções de acesso ao leitor/cartões e ao SAM poderão ser fornecidas através da API de acesso ao leitor da ASK. Esta API será descrita em Anexo a este documento.

Antes de a Aplicação poder usar as funções da API Andante, é necessário que todas as funções de callback sejam definidas. Caso se use a API ASK, a Aplicação deverá indicar à API Andante as funções de callback que estão definidas na API ASK.

Nos subcapítulos seguintes, descrevem-se quais as funções de callback que devem ser definidas e implementadas pela Aplicação.

A API Andante é informada sobre as funções de callback, através da chamada à função `Andante_SetCallback()` da API Andante, pelo que a Aplicação deverá chamar esta função por cada função de callback necessária ao funcionamento da API.

Excepto para as funções de callback de acesso a funcionalidades do sistema operativo, todas elas têm como primeiro parâmetro um long de utilização livre. O valor deste parâmetro é definido pela Aplicação aquando da sua inicialização na API Andante através da função da API Andante `Andante_SetCallback`. O significado deste parâmetro é irrelevante para a API Andante, sendo da responsabilidade da Aplicação dar-lhe significado se assim o desejar. Como hipótese de utilização deste parâmetro pela Aplicação, é o número do porto série onde o leitor de cartões está ligado.

2.4.1. “CALLBACKS” DE ACESSO AO SISTEMA OPERATIVO

Nesta categoria de funções de callback estão indicadas quais as funções a disponibilizar pela Aplicação que têm a ver com funções associadas ao sistema operativo onde a Aplicação é executada.

2.4.1.1. “Callback” OS_Malloc()

Descrição:

Aloca um bloco de memória.

O bloco alocado deve ser posteriormente libertado através de OS_Free().

Esta função está disponível se a aplicação utilizar a “C Run-Time library” e incluir <stdlib.h>.

Protótipo:

```
void * OS_Malloc(unsigned long size);
```

Parâmetros

unsigned long	size	Entrada	Nº de bytes a alocar
---------------	------	---------	----------------------

Retorno:

void *	!= 0	Apontador para o bloco de memória alocado;
--------	------	--

void *	0	Erro na alocação de memória.
--------	---	------------------------------

Ver também:

```
OS_Free(),  
Andante_SetCallback()
```

2.4.1.2. “Callback” OS Free()

Descrição:

Liberta o bloco de memória alocado por OS_Malloc().

Esta função está disponível se a aplicação utilizar a “C Run-Time library” e incluir <stdlib.h>.

Protótipo:

```
void OS_Free(void *memblock);
```

Parâmetros

void *	memblock	Entrada	Apontador para o bloco de memória a libertar (retornado por OS_Malloc)
--------	----------	---------	--

Retorno:

Nenhum

Ver também:

```
OS_Malloc(), Andante_SetCallback()
```

2.4.1.3. “Callback” OS Sprintf() Descrição:

Escreve dados formatados numa string.

Esta função está disponível se a aplicação utilizar a “C Run-Time library” e incluir <stdio.h>.

Protótipo:

```
int OS_Sprintf(char *buffer, const char *format, ...);
```

Parâmetros

<code>char *</code>	<code>buffer</code>	Saída	Apontador para a string onde se vão escrever os dados formatados;
<code>const char *</code>	<code>format</code>		String de formatação;
<code>...</code>			Argumentos opcionais.

Retorno:

<code>int</code>	Nº de bytes escritos na string de output buffer.
------------------	--

Ver também:

`Andante_SetCallback()`

2.4.2. “CALLBACKS” DE ACESSO AO SAM

A função de callback aqui definida deve permitir o acesso ao módulo de segurança (SAM). Caso o leitor usado seja o da ASK, a Aplicação poderá usar a função equivalente existente na API Ask.

2.4.2.1. “Callback” SAM_TxRxAPDU()

Descrição:

Envia um comando ISO e recebe a resposta correspondente do SAM.

Esta função está disponível se a aplicação utilizar o SAM colocado num “coupler” ASK.

Protótipo:

```
unsigned long SAM_TxRxAPDU(
    long Param
    , unsigned char *TxData
    , unsigned long TxDataLen
    , unsigned char *RxData
```



```
, unsigned long *RxDataLen );
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char *</code>	<code>TxData</code>	Entrada	Dados a enviar para o SAM;
<code>unsigned long</code>	<code>TxDataLen</code>	Entrada	Comprimento dos dados a enviar;
<code>unsigned char *</code>	<code>RxData</code>	Saída	Dados recebidos do SAM;
<code>unsigned long *</code>	<code>RxDataLen</code>	Saída	Comprimento dos dados recebidos.

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3. “CALLBACKS” DE ACESSO A UM CARTÃO DE MEMÓRIA

Definição das funções de callback que permitem o acesso aos cartões de memória que podem ser usados. Nomeadamente os bilhetes de baixo custo, CTS.

O conjunto das funções de callback aqui indicadas podem ser as que são definidas na API Ask para acesso a este tipo de cartões, caso o leitor seja o da ASK.

2.4.3.1. “Callback” CTS256B_Read()

Descrição:

Lê dados de um cartão de memória CTS-256B.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long CTS256B_Read(
                                long Param
                                , unsigned char DataAddress
                                , unsigned char DataLen
                                )
```

```
, unsigned char *Data )
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a ler
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a ler
<code>unsigned char *</code>	<code>Data</code>	Saída	Dados lidos

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.2. “Callback” CTS256B_Update()

Descrição:

Escreve dados num cartão de memória CTS-256B.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long CTS256B_Update(          long Param
                                     , unsigned char DataAddress
                                     , unsigned char DataLen
                                     , unsigned char *OldData
                                     , unsigned char *UpdateData
                                     , unsigned char *NewData );
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a escrever;

<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a escrever;
<code>unsigned char *</code>	<code>OldData</code>	Entrada	Dados existentes no cartão;
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados a escrever;
<code>unsigned char *</code>	<code>NewData</code>	Saída	Dados no cartão após a escrita.

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.3. “Callback” CTS512B Read()

Descrição:

Lê dados de um cartão de memória CTS-512B.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long CTS512B_Read(           long Param
                                     , unsigned char DataAddress
                                     , unsigned char DataLen
                                     , unsigned char *Data )
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a ler
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a ler
<code>unsigned char *</code>	<code>Data</code>	Saída	Dados lidos

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
----------------------------	---	-------------------------------

`unsigned long` `!= 0` Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.4. “Callback” CTS512B_Update()

Descrição:

Escreve dados num cartão de memória CTS-512B.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long CTS512B_Update(           long Param
                                     , unsigned char DataAddress
                                     , unsigned char DataLen
                                     , unsigned char *UpdateData
                                     , unsigned char *NewData );
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a escrever;
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a escrever;
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados a escrever;
<code>unsigned char *</code>	<code>NewData</code>	Saída	Dados no cartão após a escrita.

Retorno:

<code>unsigned long</code>	<code>0</code>	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.5. “Callback” CTS512B Write()

Descrição:

Escreve dados em zonas OTP num cartão de memória CTS-512B. Esta

função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long CTS512B_Write(          long Param
                                     , unsigned char DataAddress
                                     , unsigned char DataLen
                                     , unsigned char *UpdateData
                                     , unsigned char *NewData );
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a escrever;
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a escrever;
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados a escrever;
<code>unsigned char *</code>	<code>NewData</code>	Saída	Dados no cartão após a escrita.

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.6. “Callback” MFlight Read()

Descrição:

Lê dados de um cartão de memória MIFARE Ultralight.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long MFlight_Read(                long Param

                                         , unsigned char DataAddress

                                         , unsigned char DataLen

                                         , unsigned char *Data )
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a ler
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a ler
<code>unsigned char *</code>	<code>Data</code>	Saída	Dados lidos

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.7. “C al l bac k” MFI g ht Update()

Descrição:

Escreve dados num cartão de memória MIFARE Ultralight.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long MFlight_Update(                long Param

                                         , unsigned char DataAddress

                                         , unsigned char DataLen

                                         , unsigned char *UpdateData

                                         , unsigned char *NewData );
```

Parâmetros:

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a escrever;
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a escrever;
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados a escrever;
<code>unsigned char *</code>	<code>NewData</code>	Saída	Dados no cartão após a escrita.

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.3.8. “C allback” MFlight_Write()

Descrição:

Escreve dados em zonas OTP num cartão de memória MIFARE Ultralight. Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long MFlight_Write(           long Param
                                   , unsigned char DataAddress
                                   , unsigned char DataLen
                                   , unsigned char *UpdateData
                                   , unsigned char *NewData );
```

Parâmetros:

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço dos dados a escrever;
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Comprimento dos dados a escrever;
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados a escrever;

<code>unsigned char *</code>	<code>NewData</code>	Saída	Dados no cartão após a escrita.
------------------------------	----------------------	-------	---------------------------------

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
----------------------------	---	-------------------------------

<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.
----------------------------	-------------------	-----------------------------

Ver também:

`Andante_SetCallback()`

2.4.4. “CALLBACKS” DE ACESSO A UM CARTÃO COM MICROPROCESSADOR

Definição das funções de callback que permitem o acesso aos cartões com microprocessador que podem ser usados. Nomeadamente os cartões GTML-2.

O conjunto das funções de callback aqui indicadas podem ser as que são definidas na API Ask para acesso a este tipo de cartões, caso o leitor seja o da ASK.

2.4.4.1. “Callback” Card TxRxAPDU ()

Descrição:

Envia um commando ISO e recebe a resposta correspondente de um cartão com microprocessador CALYPSO (ex: GTML-2, CD-LIGHT, TANGO, CD21).

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_TxRxAPDU(
                                long Param
                                , unsigned char *BufIN
                                , unsigned long LnIN
                                , unsigned char *BufOUT
                                , unsigned long *lpLnOUT );
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char *</code>	<code>BufIN</code>	Entrada	Dados a enviar para o cartão;
<code>unsigned long</code>	<code>LnIN</code>	Entrada	Comprimento dos dados a enviar;

<code>unsigned char *</code>	<code>BufOUT</code>	Saída	Dados recebidos do cartão;
<code>unsigned long *</code>	<code>lpLnOUT</code>	Saída	Comprimento dos dados recebidos.

Retorno:

<code>unsigned long</code>	<code>0</code>		Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>		Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.4.2. “Callback” Card OpenSession()

Descrição:

Abre uma sessão segura com o cartão.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_OpenSession(          long Param
                                     , unsigned char Type
                                     , unsigned char SID
                                     , unsigned char NKEY
                                     , unsigned char Nrec
                                     , unsigned char *Record);
```

Parâmetro:

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>Type</code>	Entrada	Tipo de operação: 0x00: Personalização; 0x01: Carregamento; 0x02: Validação.
<code>unsigned char</code>	<code>SID</code>	Entrada	Nº de identificação curto do ficheiro
<code>unsigned char</code>	<code>NKEY</code>	Entrada	Posição, no SAM, da chave a utilizar

<code>unsigned char</code>	<code>Nrec</code>	Entrada	Número do registo;
<code>unsigned char *</code>	<code>Record</code>	Saída	Dados retornados;

Retorno:

<code>unsigned long</code>	0		Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>		Erro na execução da função.

Ver também:

`Card_CloseSession()`, `Andante_SetCallback()`

2.4.4.3. “C allback” Card CloseSession()

Descrição:

Fecha a sessão segura com o cartão (aberta via `Card_OpenSession()`). Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_CloseSession(          long Param
                                     , unsigned char *Result
                                     , unsigned long *cbResult);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre
<code>unsigned char *</code>	<code>Result</code>	Saída	Resultado do comando;
<code>unsigned long *</code>	<code>cbResult</code>	Saída	Comprimento do resultado;

Retorno:

<code>unsigned long</code>	0		Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>		Erro na execução da função.

Ver também:

`Card_OpenSession()`, `Andante_SetCallback()`

2.4.4.4. “C allback” Card UpdateRecord()

Descrição:

Escreve um registo (update) num cartão com microprocessador. Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_UpdateRecord(          long Param
                                     , unsigned char  AccMode
                                     , unsigned char  SID
                                     , unsigned char  LID
                                     , unsigned char  NKEY
                                     , unsigned char  NuRec
                                     , unsigned char  DataLen
                                     , unsigned char *Data);
```

Parâmetros

long	Param	Entrada	Parâmetro de utilização livre;
unsigned char	AccMode	Entrada	Modo de acesso: = Personalização; = Carregamento; = Validação
unsigned char	SID		Nº de identificação curto do ficheiro
unsigned char	LID		Nº de identificação longo do ficheiro
unsigned char	NKEY	Entrada	Posição, no SAM, da chave a utilizar
unsigned char	NuRec	Entrada	Número do registo;
unsigned char	DataLen	Entrada	Comprimento dos dados a escrever;
unsigned char *	Data	Entrada	Dados a escrever;
Retorno:			
unsigned long	0		Função executada com sucesso;

`unsigned long` != 0 Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.4.5. “C allback” Card AppendRecord()

Descrição:

Acrescenta um registo (append) num cartão com microprocessador.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_AppendRecord(          long Param
                                         , unsigned char AccMode
                                         , unsigned char SID
                                         , unsigned char LID
                                         , unsigned char NKEY
                                         , unsigned char *Rec
                                         , unsigned char RecSize);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>AccMode</code>	Entrada	Modo de acesso: = Personalização; = Carregamento; = Validação
<code>unsigned char</code>	<code>SID</code>		Nº de identificação curto do ficheiro
<code>unsigned char</code>	<code>LID</code>		Nº de identificação longo do ficheiro
<code>unsigned char</code>	<code>NKEY</code>	Entrada	Posição, no SAM, da chave a utilizar
<code>unsigned char *</code>	<code>Rec</code>	Entrada	Dados a escrever;

<code>unsigned char</code>	<code>RecSize</code>	Entrada	Comprimento dos dados a escrever;
----------------------------	----------------------	---------	-----------------------------------

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
----------------------------	---	-------------------------------

<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.
----------------------------	-------------------	-----------------------------

Ver também:

`Andante_SetCallback()`

2.4.4.6. “C allback” Card Decrease()

Descrição:

Decrementa um contador num cartão com microprocessador.

Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_Decrease(          long Param
                                   , unsigned char AccMode
                                   , unsigned char SID
                                   , unsigned char LID
                                   , unsigned char NKEY
                                   , unsigned char Icount
                                   , unsigned long Value
                                   , unsigned long *NewValue);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
-------------------	--------------------	---------	--------------------------------

<code>unsigned char</code>	<code>AccMode</code>	Entrada	Modo de acesso:
			= Personalização;
			= Carregamento;
			= Validação

<code>unsigned char</code>	<code>SID</code>		Nº de identificação curto do ficheiro
----------------------------	------------------	--	---------------------------------------

<code>unsigned char</code>	<code>LID</code>		Nº de identificação longo do ficheiro
<code>unsigned char</code>	<code>NKEY</code>	Entrada	Posição, no SAM, da chave a utilizar
<code>unsigned char</code>	<code>Icount</code>	Entrada	Índice do contador;
<code>unsigned long</code>	<code>Value</code>	Entrada	Valor a decrementar;
<code>unsigned long *</code>	<code>NewValue</code>	Saída	Novo valor do contador (fora do modo de sessão);

Retorno:

<code>unsigned long</code>	<code>0</code>	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

`Andante_SetCallback()`

2.4.4.7. “Callback” Card ChangePIN()

Descrição:

Altera o PIN (Personal Identification Number) de um cartão com microprocessador. Esta função está disponível se a aplicação utilizar um “coupler” ASK.

Protótipo:

```
unsigned long Card_ChangePIN(
    long Param
    , unsigned char NKEY
    , unsigned char *OldPIN
    , unsigned char *NewPIN
    , unsigned char P1);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização livre;
<code>unsigned char</code>	<code>NKEY</code>	Entrada	Posição, no SAM, da chave a utilizar;
<code>unsigned char *</code>	<code>OldPIN</code>	Entrada	PIN actual;
<code>unsigned char *</code>	<code>NewPIN</code>	Entrada	Novo PIN;

<code>unsigned char</code>	<code>P1</code>	Entrada	Configuração do tipo de cartão
----------------------------	-----------------	---------	--------------------------------

Retorno:

<code>unsigned long</code>	<code>0</code>		Função executada com sucesso;
----------------------------	----------------	--	-------------------------------

<code>unsigned long</code>	<code>!= 0</code>		Erro na execução da função.
----------------------------	-------------------	--	-----------------------------

Ver também:

`Andante_SetCallback()`

2.5. FUNÇÕES DISPONIBILIZADAS PELA API

Nos subcapítulos seguintes, descrevem-se as funções disponibilizadas pela API Andante.

A maioria das funções desta categoria utiliza um handler que está associado a um determinado recurso usado pela API Andante. Cada um dos seguintes recursos pode ter um handler associado:

Antena (leitor);

Módulo de segurança (SAM);

Cartão.

De referir que na definição das funções de callback que a Aplicação deve definir e informar a API Andante devem ser associadas ao handler correspondente ao leitor (handler devolvido pela função `Andante_NewAntenna`).

2.5.1. FUNÇÕES DE GESTÃO DA API

2.5.1.1. Andante_NewAntenna()

Descrição:

Aloca e inicializa memória necessária para a API poder suportar a utilização de uma antena.

A aplicação deverá chamar esta função para cada uma das antenas disponíveis no equipamento.

Protótipo:

`ANDANTE_HANDLER Andante_NewAntenna(void);`

Parâmetros

Nenhum

Retorno:

```

ANDANTE_HANDLER      != 0      Handler da antenna;

```

ANDANTE_HANDLER	0	Erro.
-----------------	---	-------

Ver também:

```
Andante_DeleteAntenna()
```

2.5.1.2. Andante DeleteAntenna()

Descrição:

Liberta a memória alocada por Andante_NewAntenna().

Protótipo:

```
void Andante_DeleteAntenna(ANDANTE_HANDLER AntennaHandler);
```

Parâmetros

ANDANTE_HANDLER	AntennaHandler	Entrada	Handler da antena.
-----------------	----------------	---------	--------------------

Retorno:

Nenhum

Ver também:

```
Andante_NewAntenna()
```

2.5.1.3. Andante NewSAM()

Descrição:

Aloca e inicializa memória necessária para a API poder suportar a utilização de um SAM.

A Aplicação deverá chamar esta função para cada um dos SAM's disponíveis no equipamento.

Protótipo:

[illegible]

Parâmetros

<code>unsigned char *</code>	<code>Atr</code>	posta ao RESET (Answer To Reset) do SAM;	
<code>unsigned char</code>	<code>AtrLen</code>	Entrada	Comprimento do ATR;
<code>ENUM_SAM_TYPE</code>	<code>SAMtype</code>	Entrada	Tipo de SAM.

Retorno:

<code>ANDANTE_HANDLER</code>	<code>!= 0</code>	Handler do SAM;
<code>ANDANTE_HANDLER</code>	<code>0</code>	Erro.

Ver também:

`Andante_DeleteSAM()`

2.5.1.4. Andante_DeleteSAM()

Descrição:

Liberta a memória alocada por `Andante_NewSAM()`.

Protótipo:

```
void Andante_DeleteSAM(ANDANTE_HANDLER SAMhandler)
```

Parâmetros

<code>ANDANTE_HANDLER</code>	<code>SAMhandler</code>	Entrada	Handler do SAM
------------------------------	-------------------------	---------	----------------

Retorno:

Nenhum

Ver também:

`Andante_NewSAM()`

2.5.1.5. Andante_NewCard()

Descrição:

Aloca e inicializa memória necessária para a API poder realizar as operações sobre um cartão Andante.

Protótipo:

```
ANDANTE_HANDLER Andante_NewCard(ANDANTE_HANDLER AntennaHandler
```

```
, unsigned char *Atr
```

```
, unsigned char AtrLen );
```

Parâmetros

ANDANTE_HANDLER	AntennaHandler	Entrada	Handler da antena que irá interactivar com o cartão Andante
unsigned char *	Atr	Entrada	Resposta ao RESET ao cartão
unsigned char	AtrLen	Entrada	Comprimento do ATR.

Retorno:

ANDANTE_HANDLER	!= 0	Handler do cartão;
ANDANTE_HANDLER	0	Erro.

Ver também:

[Andante_DeleteCard\(\)](#)

2.5.1.6. Andante_DeleteCard()

Descrição:

Liberta a memória alocada por [Andante_NewCard\(\)](#).

Protótipo:

```
void Andante_DeleteCard(ANDANTE_HANDLER CardHandler)
```

Parâmetros:

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão.
-----------------	-------------	---------	--------------------

Retorno:

Nenhum

Ver também:

[Andante_NewCard\(\)](#)

2.5.1.7. Andante_SetCallback()

Descrição:

Define uma função de “callback” para o handler indicado.

Protótipo:

```
unsigned long Andante_SetCallback(          ANDANTE_HANDLER Handler
                                         , ENUM_CALLBACK_TYPE CallbackType
                                         , void *CallbackFunction
                                         , long CallbackParam );
```

Parâmetros

ANDANTE_HANDLER	Handler	Entrada	Handler;
ENUM_CALLBACK_TYPE	CallbackType	Entrada	Tipo de “callback”;
void *	CallbackFunction	Entrada	Função de “callback”
long	CallbackParam	Entrada	Parâmetro de utilização livre (1º argumento das funções de “callback”).

Retorno:

unsigned long	0	Função executada com sucesso;
unsigned long	!= 0	Erro na execução da função.

Ver também:

2.5.2. FUNÇÕES DE CONFIGURAÇÃO

2.5.2.1. Andante_SetConfigParam()

Descrição:

Permite informar a API Andante quais os parâmetros que descrevem algumas das características do equipamento.

Esta função deve ser chamada sempre que algum parâmetro sofra alterações.

Protótipo:

```
void Andante_SetConfigParam(          ENUM_CONFIG_PARAM ConfigParam
                                   , unsigned long Value);
```

Parâmetros

ENUM_CONFIG_PARAM	ConfigParam	Entrada	Tipo de parâmetro a configurar.
-------------------	-------------	---------	---------------------------------

`unsigned long` `Value` Valor a atribuir ao parâmetro escolhido

Retorno:

Nenhum

Ver também:

2.5.2.2. Andante_GetCardInfo()

Descrição:

Permite obter o número de série do cartão, a partir da resposta ao RESET (ATR)

Protótipo:

```
unsigned long Andante_GetCardInfo(           unsigned char *Atr
                                           , unsigned char  AtrLen
                                           , char           *SerialNumber );
```

Parâmetros

<code>unsigned char *</code>	<code>Atr</code>	Entrada	Resposta ao RESET (ATR)
<code>unsigned char</code>	<code>AtrLen</code>	Entrada	Tamanho do ATR
<code>char *</code>	<code>SerialNumber</code>	Saída	Número de série do cartão

Retorno:

<code>unsigned long</code>	0	Função executada com sucesso;
<code>unsigned long</code>	<code>!= 0</code>	Erro na execução da função.

Ver também:

2.5.2.3. Andante_SetSAM()

Descrição:

Faz a associação entre a utilização de um cartão e um SAM.

Protótipo:

```
unsigned long Andante_SetSAM(           ANDANTE_HANDLER CardHandler
```

```
, ANDANTE_HANDLER SAMhandler );
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão;
ANDANTE_HANDLER	SAMhandler	Entrada	Handler do SAM.

Retorno:

unsigned long	0	Função executada com sucesso;
unsigned long	!= 0	Erro na execução da função.

Ver também:

2.5.3. FUNÇÕES DE EXPLORAÇÃO

2.5.3.1. Andante_Read()

Descrição:

Permite ler o conteúdo de um cartão. O resultado da leitura, caso seja bem sucedida, é colocado numa estrutura interna da API. A aplicação poderá consultar este conteúdo através da função da API Andante_GetTitleInfo().

Protótipo:

```
unsigned long Andante_Read(ANDANTE_HANDLER CardHandler);
```

Parâmetros

ANDANTE_HANDLER	AntennaHandler	Entrada	Handler do cartão
-----------------	----------------	---------	-------------------

Retorno:

unsigned long	0	Função executada com sucesso
	!= 0	Erro na execução da função

Ver também:

2.5.3.2. Andante_GetTitleInfo()

Protótipo:

```

unsigned long Andante_GetTitleInfo(
    ANDANTE_HANDLER CardHandler
    , unsigned char ContractNumber
    , TYPE_TITLE_INFO *Info
    , unsigned long Now );

```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
unsigned char	ContractNumber	Entrada	Número do contracto que se quer inpeccionar
TYPE_TITLE_INFO	Info	Saída	Apontador para a estrutura de dados para onde é copiada a informação sobre o título.
unsigned long	Now	Entrada	Data e Hora actuais no formato: número de segundos desde
Retorno:			1970/01/01 00:00:00.
unsigned long	0		Função executada com sucesso
	!= 0		Erro na execução da função

Ver também:

2.5.3.3. Andante Load()

Descrição:

Protótipo:

```

unsigned long Andante_Load(
    ANDANTE_HANDLER CardHandler
    , unsigned long Now
    , unsigned char ContractNumber
    , unsigned long ReloadNumber
    , TYPE_CONTRACT_LOAD *Contract );

```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão;
unsigned long	Now	Entrada	Data e hora actuais (formato: segundos desde <1970-01-01 00:00:00>);
unsigned char	ContractNumber	Entrada	Nº do contrato
unsigned long	ReloadNumber	Contador	de carregamentos efectuados num determinado posto. Este valor deve ser colocado a zero no início de cada dia.
TYPE_CONTRACT_LOAD*	Contract	Contador	para a estrutura de dados contendo a informação sobre o título a carregar.
Retorno:			
unsigned long	0	Função executada com sucesso;	
unsigned long	!= 0	Erro na execução da função.	

Ver também:

2.5.3.4. Andante Validate()

Descrição:

Permite proceder à validação de um título de transporte, a fim de aferir se o título é ou não válido. Esta função será usada em equipamentos de validação.

Protótipo:

```
unsigned long Andante_Validate(
    ANDANTE_HANDLER CardHandler
    , unsigned long Now
    , unsigned char *ContractNumber );
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
unsigned long	Now	Entrada	Data e hora actual no formato: número de segundos desde 1970/01/01 00:00:00

<code>short *</code>	<code>ContractNumber</code>	Saída	Número do contracto usado no processo de validação.
Retorno:			
<code>unsigned long</code>	0		Função executada com sucesso
	<code>!= 0</code>		Erro na execução da função

Ver também:

2.5.3.5. Andante Control()

Descrição:

Esta função deverá ser chamada pelos equipamentos de fiscalização. Verifica se a última validação efectuada corresponde a uma viagem válida.

Protótipo:

```
unsigned long Andante_Control(
                                ANDANTE_HANDLER CardHandler
                                , unsigned long Now
                                , unsigned char *ContractNumber );
```

Parâmetros

<code>ANDANTE_HANDLER</code>	<code>CardHandler</code>	Entrada	Handler do cartão
<code>unsigned long</code>	<code>Now</code>	Entrada	Data e hora e actual no formato: número de segundos desde 1970/01/01 00:00:00
<code>short *</code>	<code>ContractNumber</code>	Saída	Indica qual o número do contracto que foi usado no processo de fiscalização.
Retorno:			
<code>unsigned long</code>	0		Função executada com sucesso
	<code>!= 0</code>		Erro na execução da função

Ver também:

2.5.3.6. Andante Invalidate()

Descrição:

Esta função deverá ser chamada pelos equipamentos sempre que seja necessário invalidar um suporte eletronicamente.

Será utilizado pelos equipamentos de validação sempre que seja detetado um suporte em lista negra.

Protótipo:

```
unsigned long Andante_Invalidate(          ANDANTE_HANDLER CardHandler );
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
-----------------	-------------	---------	-------------------

Retorno:

unsigned long	0	Função executada com sucesso
	!= 0	Erro na execução da função

Ver também:

2.5.3.7. Andante SaveData()

Descrição:

Esta função permite guardar o conteúdo dum cartão em memória. O conteúdo salvaguardado é o referente ao lido do cartão através das funções Andante_Read(), Andante_Issue(), Andante_Load(), Andante_Validate() e Andante_Control().

Quer o conteúdo da memória activa (zona de memória onde é guardado o conteúdo de um cartão depois de lido, através das funções atrás mencionadas) quer o conteúdo salvaguardado pela função Andante_SaveData(), tem associado o número de série do cartão.

Protótipo:

```
ANDANTE_HANDLER Andante_SaveData(ANDANTE_HANDLER CardHandler);
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
-----------------	-------------	---------	-------------------

Retorno:

ANDANTE_HANDLER	0	Não foi efectuada a salvaguarda dos dados do cartão
	!= 0	Devolve um handler que correspondente ao conteúdo dos dados salvaguardados (handler de salvaguarda).

Ver também:

2.5.3.8. Andante RestoreData()

Descrição:

Esta função permite restaurar o conteúdo dos dados, previamente salvaguardado pela função Andante_SaveData(), para a memória activa e, opcionalmente, escrever esse conteúdo no cartão.

Protótipo:

```
unsigned long Andante_RestoreData(          ANDANTE_HANDLER CardHandler
                                           , ANDANTE_HANDLER Data
                                           , char *SerialNumber
                                           , int WriteCard );
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
ANDANTE_HANDLER	Data	Entrada	Handler de salvaguarda, retornado pela função Andante_SaveData()
char *	SerialNumber	Entrada	Caso seja NULL ou string vazia, os números de série associados (do Data e do CardHandler) têm de ser iguais, caso contrário esta função falha. Caso seja preenchido com um número de série, a função só se realiza com sucesso se o número de série colocado neste parâmetro coincidir com o número de série associado ao CardHandler.

<code>Int</code>	<code>WriteCard</code>	Entrada	Permite definir se o restauro dos dados (previamente guardados por <code>Andante_SaveData</code>) é efectuada no cartão colocado na antena.
Retorno:			Se 0, só restaura a memória activa. Nos restantes casos, restaura também no cartão.
<code>unsigned long</code>	0		Função executada com sucesso
	<code>!= 0</code>		Erro na execução da função

Ver também:

2.5.3.9. Andante_FreeData()

Descrição:

Liberta o espaço de memória alocado em `Andante_SaveData()`.

Protótipo:

```
unsigned long Andante_FreeData(ANDANTE_HANDLER Data);
```

Parâmetros

<code>ANDANTE_HANDLER</code>	<code>Data</code>	Entrada	Handler de salvaguarda retornado por uma chamada a <code>Andante_SaveData()</code> .
Retorno:			
<code>unsigned long</code>	0		Função executada com sucesso
	<code>!= 0</code>		Erro na execução da função

Ver também:

2.5.4. EXTENÇÃO DAS FUNÇÕES DE EXPLORAÇÃO

2.5.4.1. Andante_Issue() Descrição:

Permite personalizar um cartão.

Protótipo:

```
unsigned long Andante_Issue(  ANDANTE_HANDLER CardHandler,
unsigned long Now,

                                TYPE_CARD_ISSUE *IssuingData,
unsigned char **UndoData,

short *UndoDataLen,                                unsigned
char **SaveData,                                short
*SaveDataLen);
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
unsigned long	Now	Entrada	Data e hora e actual no formato: número de segundos desde 1970/01/01 00:00:00
TYPE_CARD_ISSUE	IssuingData	Entrada	Informação do cartão a personalizar
unsigned char	UndoData	Saída	
short unsigned	UndoDataLen	Saída	
char short	SaveData	Saída	
	SaveDataLen	Saída	

Retorno:

unsigned long	0	Função executada com sucesso
	!= 0	Erro na execução da função

Ver também:

2.5.4.2. Andante_ChangeProfiles()

Descrição:

Permite alterar a informação de personalização do cartão.

Protótipo:

```
unsigned long Andante_ChangeProfiles( ANDANTE_HANDLER DataHandler,  
                                     TYPE_CARD_ISSUE *IssuingData);
```

Parâmetros

ANDANTE_HANDLER	CardHandler	Entrada	Handler do cartão
TYPE_CARD_ISSUE	IssuingData	Entrada	Informação a alterar na personalização do cartão

Retorno:

unsigned long	0	Função executada com sucesso
	!= 0	Erro na execução da função

Ver também:

3. ANEXO A – API ASK

3.1. INTRODUÇÃO

Quando o leitor de cartões que suportam os títulos de transporte for o leitor da ASK, a Aplicação pode usar as funções disponibilizadas pela API da ASK. Neste caso, a maioria das funções de callback que são necessárias à utilização da API Andante estão definidas nesta API, descrita neste anexo. Esta API permite implementar todos os pormenores de implementação de acesso ao leitor da ASK (leitor, SAM e cartões), sendo responsável pela construção dos comandos necessários ao acesso a esses recursos.

Apesar da construção dos comandos ser efetuada por esta API, o envio/receção dos dados ao leitor deverá ser implementada pela Aplicação, já que a implementação depende do equipamento onde a Aplicação é executada. Assim, existe um conjunto de funções de callback que são necessárias para o funcionamento desta API e que servem essencialmente para implementarem as funções de baixo nível que permitem o envio/receção de dados para o leitor da ASK.

3.2. FICHEIROS DA API ASK

Os ficheiros necessários para a utilização desta API são os seguintes:

READask.h: Ficheiro que contém os protótipos das funções e estruturas necessários à utilização da API;

READask.lib: Livraria contendo o código objeto correspondente à API que permite o acesso ao leitor da ASK. Esta livraria será construída de acordo com o ambiente de desenvolvimento usado no equipamento onde será usada a API. A livraria está construída em ANSI C.

3.3. FUNÇÕES DE CALLBACK

Nos subcapítulos seguintes descrevem-se as funções que necessitam de ser implementadas pela Aplicação.

3.3.1. DELAY

Descrição:

Deverá implementar uma espera de um número determinado de milissegundos.

Protótipo:

```
void Delay(long Param, unsigned long MiliSecs);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização genérica pela Aplicação.
-------------------	--------------------	---------	--

<code>unsigned long</code>	<code>MiliSecs</code>	Entrada	Número de milissegundos de espera
----------------------------	-----------------------	---------	-----------------------------------

Retorno:

Ver também:

`READask_SetCallback()`

3.3.2. FLUSHTXRX

Descrição:

Deverá fazer o flush dos buffers de transmissão e recepção no canal de comunicações estabelecido com o leitor da ASK.

Protótipo:

```
void FlushTxRx(long Param);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização genérica pela Aplicação.
-------------------	--------------------	---------	--

Retorno:

Ver também:

`READask_SetCallback()`

3.3.3. TX

Descrição:

Deverá fazer o envio de dados para o leitor da ASK.

Protótipo:

```
int Tx(long Param, unsigned char *BufIN, unsigned long LnIN);
```

Parâmetros

<code>long</code>	<code>Param</code>	Entrada	Parâmetro de utilização genérica pela Aplicação.
<code>unsigned char *</code>	<code>BufIN</code>	Entrada	Ponteiro para buffer contendo os dados a serem enviados para o leitor da ASK.
<code>unsigned long</code>	<code>LnIN</code>	Entrada	Número de bytes que devem ser transmitidos para o leitor da ASK. Bytes esses que estão no buffer apontado pelo parâmetro <code>BufIN</code>
Retorno:			
<code>int</code>	<code>= 0</code>		Operação efetuada com sucesso
	<code>!= 0</code>		Operação falhou

Ver também:

`READask_SetCallback()`

3.3.4. RX

Descrição:

Deverá fazer a recepção de dados do leitor da ASK.

Protótipo:

```
int Rx(long Param, unsigned long TimeOut, unsigned long Len, unsigned char *BufOUT);
```

Parâmetros

<code>long</code>	<code>Param</code>		Parâmetro de utilização genérica pela Aplicação.
<code>unsigned long</code>	<code>TimeOut</code>	Entrada	Número de milissegundos máximo para que todos os bytes estipulados no parâmetro <code>Len</code> sejam recebidos do leitor da ASK. Passado este tempo, função deve abortar o processo de leitura dos dados e retornar erro.
<code>unsigned long</code>	<code>Len</code>	Entrada	Número de bytes que devem ser lidos do leitor da ASK. Bytes esses que deverão ser colocados no buffer apontado por <code>BufOut</code>

`unsigned char *` `BufOUT` Ponteiro para buffer contendo os dados lidos do leitor da ASK.

Retorno:

`int` `= 0` Operação efetuada com sucesso

`!= 0` Operação falhou

Ver também:

`READask_SetCallback()`

3.4. FUNÇÕES DA API

3.4.1. READASK SETCALLBACK

Descrição:

Permite configurar as funções de callback necessárias ao funcionamento desta API.

Protótipo:

```
unsigned long READask_SetCallback (                      COUPLER_HANDLER CouplerHandler
                                     , char CallbackType
                                     , TYPE_CALLBACK_FUNCTION CallbackFunction
                                     , long CallbackParam );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor da ASK (retornado na chamada à função <code>READask_New</code>).
<code>char</code>	<code>CallbackType</code>	Entrada	Função de callback a definir: ‘D’ : Delay ‘F’ : FlushTxRx ‘T’ : Tx ‘R’ : Rx
<code>TYPE_CALLBACK_FUNCTION</code>	<code>CallbackFunction</code>	Entrada	Ponteiro para a função de callback

<code>long</code>	<code>CallbackParam</code>	Entrada	Parâmetro de utilização genérica pela Aplicação.
-------------------	----------------------------	---------	--

Retorno:

<code>int</code>	<code>= 0</code>	Operação efetuada com sucesso
	<code>!= 0</code>	Operação falhou

Ver também:

`READask_New()`

3.4.2. READASK_NEW

Descrição:

Permite iniciar a comunicação com o leitor da ASK. Antes de se poder proceder à primeira operação sobre o leitor, deve-se chamar esta função.

Protótipo:

```
COUPLER_HANDLER READask_New(char Protocol);
```

Parâmetros

<code>char</code>	<code>Protocol</code>	Entrada	Deve ser preenchido com um valor diferente de 'V'.
-------------------	-----------------------	---------	--

Retorno:

<code>COUPLER_HANDLER</code>	<code>= 0</code>	Operação falhou.
	<code>!= 0</code>	Operação executada com sucesso. Sempre que se proceder a operações sobre o leitor, através desta API, este Handler deverá ser utilizado.

Ver também:

`READask_Delete()`

3.4.3. READASK_DELETE

Descrição:

Permite terminar a comunicação com o leitor da ASK.

Protótipo:

```
unsigned long READask_Delete(COUPLER_HANDLER CouplerHandler);
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	O Handler do leitor devolvido e READask_New().
-----------------	----------------	---------	--

Retorno:

unsigned long	= 0	Operação executada com sucesso.
	!= 0	Operação falhou.

Ver também:

READask_New()

3.4.4. READASK RESETREADER

Descrição:

Faz reset ao leitor de cartões da ASK.

Protótipo:

```
unsigned long READask_ResetReader(COUPLER_HANDLER CouplerHandler);
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Handler do leitor para o qual se quer enviar um comando de reset. Handler este que é retornado pela função READask_New.
-----------------	----------------	---

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

READask_New()

3.4.5. READASK RESETSAM

Descrição:

Permite enviar um comando ao SAM.

Protótipo:

```
unsigned long READask_ResetSAM(          COUPLER_HANDLER CouplerHandler
                                     , unsigned char *Atr
                                     , unsigned long *AtrLen );
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char *	Atr	Saída	Ponteiro para o buffer que irá receber a resposta ao Reset ao SAM (ATR = Answer To Reset)
unsigned long *	AtrLen	Saída	É preenchido neste parâmetro o número de bytes devolvidos na resposta ao Reset.

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

3.4.6. READASK_ANTENNAOFF

Descrição:

Permite desligar a antena do leitor da ASK.

Protótipo:

```
unsigned long READask_AntennaOFF(COUPLER_HANDLER CouplerHandler);
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler associado ao leitor
-----------------	----------------	---------	-----------------------------

Retorno:

unsigned long	= 0	Função executada com sucesso
---------------	-----	------------------------------

$$\neq 0$$

Função com erro.

Ver também:

3.4.7. READASK SEARCHCARD

Descrição:

Permite colocar o leitor em modo de detecção de cartão na zona de influência da antena.

Protótipo:

```

unsigned long READask_SearchCard(
                                COUPLER_HANDLER CouplerHandler
                                , unsigned char inTypeCardToSearch
                                , unsigned char *outTypeCardFound
                                , unsigned char *Atr
                                , unsigned char *AtrLen );

```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char	inTypeCardToSearch	Entrada	Tipo de cartão a procurar. Máscara de bits identificando quais os tipos de cartões que deve tentar encontrar: 0x01: CTS 0x04: ISO-A 0x08: ISO-B
unsigned char *	outTypeCardFound	Saída	Tipo de cartão encontrado. Ver tabela anterior para os valores possíveis de serem retornados.
unsigned char *	Atr	Saída	ATR do cartão encontrado
unsigned char *	AtrLen	Saída	Tamanho em bytes do ATR

Retorno:

```
unsigned long    = 0
```

Função executada com sucesso

$$!= 0$$

Função com erro.

Ver também:

3.4.8. ROTINAS DA API (CALLBACKS DA API ANDANTE)

As rotinas descritas nos subcapítulos seguintes implementam as rotinas de callback necessárias ao funcionamento da API Andante para quando é utilizado o leitor da ASK.

3.4.8.1. Acesso ao SAM

3.4.8.1.1. READask_SAM_TxRxAPDU

Descrição:

Permite enviar um comando ISO ao módulo de segurança existente no leitor ASK.

Protótipo:

```
unsigned long READask_SAM_TxRxAPDU(          COUPLER_HANDLER CouplerHandler  
  
            , unsigned char *BufIN  
  
            , unsigned long LnIN  
  
            , unsigned char *BufOUT  
  
            , unsigned long *lpLnOUT );
```

Parâmetros

COUPLER_HANDLER	CouplerHandler		Handler do leitor
unsigned char *	BufIN	Entrada	Buffer de dados para enviar comandos ISO para o módulo de segurança
unsigned long	LnIN	Entrada	Número de bytes a enviar
unsigned char *	BufOUT	Saída	Buffer que irá receber os dados relativos à resposta ao comando ISO.
unsigned long *	lpLnOUT	Saída	Número de bytes recebidos.

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

3.4.8.2. Acesso a cartões de memória

3.4.8.2.1. READask_CTS256B_Read

Descrição:

Permite realizar a leitura de um cartão de memória de CTS256-B.

Protótipo:

```
unsigned long READask_CTS256B_Read(          COUPLER_HANDLER CouplerHandler
                                           , unsigned char DataAddress
                                           , unsigned char DataLen
                                           , unsigned char *Data );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço de memória no cartão a partir do qual será lido os dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a ler do cartão.
<code>unsigned char *</code>	<code>Data</code>	Saída	Bytes lidos do cartão

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_CTS256B_Update()`

3.4.8.2.2. READask_CTS256B_Update

Descrição:

Permite alterar o conteúdo da memória de um cartão CTS256-B.

Protótipo:

```
unsigned long READask_CTS256B_Update(      COUPLER_HANDLER CouplerHandler
                                           , unsigned char  DataAddress
                                           , unsigned char  DataLen,
                                           , unsigned char *OldData,
                                           , unsigned char *UpdateData,
                                           , unsigned char *NewData );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço a partir do qual é feita a alteração do conteúdo dos dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a alterar.
<code>unsigned char *</code>	<code>OldData</code>	Entrada	Conteúdo do cartão antes de se efetuar a alteração. A aplicação deverá efetuar uma leitura dos dados antes da sua alteração, e passar neste parâmetro esse conteúdo, de modo a que esta função possa otimizar o tempo de escrita.
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados com a alteração a aplicar no cartão
<code>unsigned char *</code>	<code>NewData</code>	Saída	Conteúdo dos dados do cartão após a alteração efetuada.

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_CTS256B_Read()`

3.4.8.2.3. READask_CTS512B_Read

Descrição:

Permite realizar a leitura de um cartão de memória de CTS512-B.

Protótipo:

```

unsigned long READask_CTS512B_Read(          COUPLER_HANDLER CouplerHandler
                                           , unsigned char  DataAddress
                                           , unsigned char  DataLen
                                           , unsigned char *Data );

```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char	DataAddress	Entrada	Endereço de memória no cartão a partir do qual será lido os dados
unsigned char	DataLen	Entrada	Número de bytes a ler do cartão.
unsigned char *	Data	Saída	Bytes lidos do cartão

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

```

READask_CTS512B_Update(), READask_CTS512B_Write()

```

3.4.8.2.4. READask_CTS512B_Update

Descrição:

Permite alterar o conteúdo da memória de um cartão CTS512-B.

Protótipo:

```

unsigned long          , unsigned char  DataAddress
READask_CTS512B_Update(
COUPLER_HANDLER
CouplerHandler

```

```
, unsigned char DataLen

, unsigned char *UpdateData

, unsigned char *NewData );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço a partir do qual é feita a alteração do conteúdo dos dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a alterar.
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados com a alteração a aplicar no cartão
<code>unsigned char *</code>	<code>NewData</code>	Saída	Conteúdo dos dados do cartão após a alteração efetuada.

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_CTS512B_Read()`, `READask_CTS512B_Write()`

3.4.8.2.5. READask_CTS512B_Write

Descrição:

Permite alterar o conteúdo de zonas OTP de um cartão CTS512-B.

Protótipo:

```
unsigned long READask_CTS512B_Write(          COUPLER_HANDLER CouplerHandler

, unsigned char DataAddress

, unsigned char DataLen

, unsigned char *UpdateData

, unsigned char *NewData );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço a partir do qual é feita a alteração do conteúdo dos dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a alterar.
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados com a alteração a aplicar no cartão
<code>unsigned char *</code>	<code>NewData</code>	Saída	Conteúdo dos dados do cartão após a alteração efetuada.

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_CTS512B_Read()`, `READask_CTS512B_Update()`

3.4.8.2.6. READask_MFlight_Read

Descrição:

Permite realizar a leitura de um cartão de memória de MIFARE Ultralight.

Protótipo:

```
unsigned long READask_MFlight_Read(
    COUPLER_HANDLER CouplerHandler, unsigned char DataAddress,
    unsigned char DataLen, unsigned char *Data );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço de memória no cartão a partir do qual será lido os dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a ler do cartão.

<code>unsigned char *</code>	<code>Data</code>	Saída	Bytes lidos do cartão
------------------------------	-------------------	-------	-----------------------

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MFlight_Update()`, `READask_MFlight_Write()`

3.4.8.2.7. READask_MFlight_Update

Descrição:

Permite alterar o conteúdo da memória de um cartão MIFARE Ultralight.

Protótipo:

```
unsigned long READask_MFlight_Update(      COUPLER_HANDLER CouplerHandler
                                         , unsigned char DataAddress
                                         , unsigned char DataLen
                                         , unsigned char *UpdateData
                                         , unsigned char *NewData );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço a partir do qual é feita a alteração do conteúdo dos dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a alterar.
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados com a alteração a aplicar no cartão
<code>unsigned char *</code>	<code>NewData</code>	Saída	Conteúdo dos dados do cartão após a alteração efetuada.

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MFlight_Read()`, `READask_MFlight_Write()`

3.4.8.2.8. READask_MFlight_Write

Descrição:

Permite alterar o conteúdo de zonas OTP de um cartão MIFARE Ultralight.

Protótipo:

```
unsigned long READask_MFlight_Write(
    COUPLER_HANDLER CouplerHandler, unsigned char DataAddress,
    unsigned char DataLen, unsigned char *UpdateData,
    unsigned char *NewData );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>DataAddress</code>	Entrada	Endereço a partir do qual é feita a alteração do conteúdo dos dados
<code>unsigned char</code>	<code>DataLen</code>	Entrada	Número de bytes a alterar.
<code>unsigned char *</code>	<code>UpdateData</code>	Entrada	Dados com a alteração a aplicar no cartão
<code>unsigned char *</code>	<code>NewData</code>	Saída	Conteúdo dos dados do cartão após a alteração efetuada.

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MFlight_Read()`, `READask_MFlight_Update()`

3.4.8.2.9. READask_MIFARE_LoadKey

Descrição:

Permite carregar as chaves de autenticação de sectores de memória de um cartão MIFARE 1k.

Protótipo:

[illegible]

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char *	Key	Entrada	Chave para autenticação de um determinado sector
unsigned char	KeyIndex	Entrada	Índice do leitor

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

```
READask_MIFARE_ChangeKey()
```

3.4.8.2.10. READask_MIFARE_ChangeKey

Descrição:

Permite alterar as chaves de autenticação de sectores de memória de um cartão MIFARE 1k.

Protótipo:

[illegible]

```
, unsigned char FinalAutKey  
  
, unsigned char *outCardNumber);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>KeyType</code>	Entrada	Indicação de tipo de chave (A ou B)
<code>unsigned char</code>	<code>SectorNum</code>	Entrada	Identificação do sector
<code>unsigned char</code>	<code>KeyIndex</code>	Entrada	Índice do leitor
<code>unsigned char *</code>	<code>DataToWrite</code>	Entrada	Trama de dados: chave A + bits de acesso + chave B
<code>unsigned char</code>	<code>FinalAutKey</code>	Entrada	Indicação de tipo de chave (A ou B)
<code>unsigned char *</code>	<code>outCardNumber</code>	Saída	Numero de série do cartão

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MIFARE_LoadKey()`

3.4.8.2.11. READask_MIFARE_Authenticate

Descrição:

Permite autenticar um sector de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_Authenticate (COUPLER_HANDLER CouplerHandle,  
  
, unsigned char KeyType  
  
, unsigned char SectorNum
```

```
, unsigned char KeyIndex);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>KeyType</code>	Entrada	Indicação de tipo de chave (A ou B)
<code>unsigned char</code>	<code>SectorNum</code>	Entrada	Identificação do sector
<code>unsigned char</code>	<code>KeyIndex</code>	Entrada	Índice do leitor

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

3.4.8.2.12. READask_MIFARE_ReadBlock

Descrição:

Permite ler um bloco de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_ReadBlock (    COUPLER_HANDLER CouplerHandler
                                             , unsigned char BlockNum
                                             , unsigned char *Data);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>BlockNum</code>	Entrada	Identificação do bloco de memória
<code>unsigned char *</code>	<code>Data</code>	Saida	Dados do bloco

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MIFARE_WriteBlock()`

3.4.8.2.13. READask_MIFARE_WriteBlock

Descrição:

Permite escrever um bloco de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_WriteBlock (    COUPLER_HANDLER CouplerHandler
, unsigned char BlockNum
                                     , unsigned char *DataToWrite
                                     , unsigned char *DataVerification);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char unsigned</code>	<code>BlockNum</code>	Entrada	Identificação do bloco de memória
<code>char *</code>	<code>DataToWrite</code>	Entrada	Dados a escrever
<code>unsigned char *</code>	<code>DataVerification</code>	Saída	Verificação dos dados enviados

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MIFARE_SimpleWriteBlock()`

3.4.8.2.14. READask_MIFARE_IncrementValue

Descrição:

Permite incrementar o conteúdo de um bloco de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_IncrementValue (    COUPLER_HANDLER CouplerHandler
                                                , unsigned char BlockNum
                                                , unsigned long Increment
                                                , unsigned long *ControlValue);
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char	BlockNum	Entrada	Identificação do bloco de memória cujo conteúdo será incrementado
unsigned long	Increment	Entrada	Ordem de grandeza de incremento
unsigned long *	ControlValue	Saída	Valor de controlo da operação (CRC)

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

```
READask_MIFARE_DecrementValue()
```

3.4.8.2.15. READask_MIFARE_DecrementValue

Descrição:

Permite decrementar o conteúdo de um bloco de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_IncrementValue (    COUPLER_HANDLER CouplerHandler
```

```
, unsigned char BlockNum
, unsigned long Decrement
, unsigned long *ControlValue);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char</code>	<code>BlockNum</code>	Entrada	Identificação do bloco de memória cujo conteúdo será decrementado
<code>unsigned long</code>	<code>Decrement</code>	Entrada	Ordem de grandeza de decremento
<code>unsigned long *</code>	<code>ControlValue</code>	Saída	Valor de controlo da operação (CRC)

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MIFARE_IncrementValue()`

3.4.8.2.16. READask_MIFARE_CopyValue

Descrição:

Permite autenticar um sector de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_CopyValue (    COUPLER_HANDLER CouplerHandler
, unsigned char SourceBlock
, unsigned char DestBlock
, unsigned long *ControlValue);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
------------------------------	-----------------------------	---------	-------------------

<code>unsigned char</code>	<code>SourceBlock</code>	Entrada	Identificação do bloco de memória de origem
<code>unsigned char</code>	<code>DestBlock</code>	Entrada	Identificação do bloco de memória de Destino
<code>unsigned long *</code>	<code>ControlValue</code>	Saída	Valor de controlo da operação (CRC)

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

3.4.8.2.17. READask_MIFARE_SimpleWriteBlock

Descrição:

Permite escrever um bloco de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_WriteBlock (    COUPLER_HANDLER CouplerHandler
, unsigned char BlockNum
, unsigned char *DataToWrite);
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler do leitor
<code>unsigned char unsigned char *</code>	<code>BlockNum</code>	Entrada	Identificação do bloco de memória
	<code>DataToWrite</code>	Entrada	Dados a escrever

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

`READask_MIFARE_WriteBlock()`

3.4.8.2.18. READask_MIFARE_ReadSectorData

Descrição:

Permite ler um sector de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_ReadSectorData (    COUPLER_HANDLER CouplerHandler
                                                , unsigned char KeyType
                                                , unsigned char SectorNum
                                                , unsigned char KeyIndex
                                                , unsigned char *outCardType
                                                , unsigned char *outCardNumber
                                                , unsigned char *ReadData);
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char	KeyType	Entrada	Indicação de tipo de chave (A ou B)
unsigned char	SectorNum	Entrada	Identificação do sector
unsigned char	KeyIndex	Entrada	Índice do leitor
unsigned char *	outCardType	Saída	Tipo de cartão
unsigned char *	outCardNumber	Saída	Número do cartão
unsigned char *	ReadData	Saída	Dados lidos do sector

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

3.4.8.2.19. READask_MIFARE_WriteSectorData

Descrição:

Permite escrever um sector de memória de um cartão MIFARE 1k.

Protótipo:

```
unsigned long READask_MIFARE_ReadSectorData (    COUPLER_HANDLER CouplerHandler
                                                , unsigned char KeyType
                                                , unsigned char SectorNum
                                                , unsigned char KeyIndex
                                                , unsigned char *DataToWrite
                                                , unsigned char DataLen);
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler do leitor
unsigned char	KeyType	Entrada	Indicação de tipo de chave (A ou B)
unsigned char	SectorNum	Entrada	Identificação do sector
unsigned char	KeyIndex	Entrada	Índice do leitor
unsigned char *	DataToWrite	Entrada	Dados a serem escritos no sector
unsigned char	DataLen	Entrada	Indicação do tamanho da trama de dados

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

3.4.8.3. Acesso a cartões com microprocessador

3.4.8.3.1. READask_CALYPSO_TxRxAPDU

Descrição:

Permite enviar um comando ISO para um cartão com microprocessador (GTML-2, CD-Light, etc)


```
, unsigned char SID

, unsigned char NKEY

, unsigned char NRec

, unsigned char *Record );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler associado ao leitor.
<code>unsigned char</code>	<code>Type</code>	Entrada	Tipo de operação: 0x00: Personalização; 0x01: Carregamento; 0x02: Validação.
<code>unsigned char</code>	<code>SID</code>	Entrada	Nº de identificação curto do ficheiro
<code>unsigned char</code>	<code>NKEY</code>	Entrada	Posição, no SAM, da chave a utilizar
<code>unsigned char</code>	<code>NRec</code>	Entrada	Número do registo;
<code>unsigned char *</code>	<code>Record</code>	Saída	Dados retornados;

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

3.4.8.3.3. READask_CALYPSO_CloseSession

Descrição:

Permite fechar uma sessão de acesso segura a um cartão com microprocessador.

Protótipo:

```
unsigned long READask_CALYPSO_CloseSession(    COUPLER_HANDLER CouplerHandler

, unsigned char *Result
```



```
, unsigned long *cbResult );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler associado ao leitor
<code>unsigned char *</code>	<code>Result</code>	Saída	Resultado do comando;
<code>unsigned long *</code>	<code>cbResult</code>	Saída	Comprimento do resultado;

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

[`READask_CALYPSO_OpenSession`](#)

3.4.8.3.4. READask_CALYPSO_UpdateRecord

Descrição:

Permite alterar o conteúdo num determinado registo de uma área de memória dos cartões com microprocessador

Protótipo:

```
unsigned long READask_CALYPSO_UpdateRecord(    COUPLER_HANDLER CouplerHandler
                                              , unsigned char  AccMode
                                              , unsigned char  SID
                                              , unsigned char  LID
                                              , unsigned char  NKEY
                                              , unsigned char  NuRec
                                              , unsigned char  DataLen
                                              , unsigned char  *Data );
```

Parâmetros


```
, unsigned char NKEY

, unsigned char *Rec

, unsigned char RecSize );
```

Parâmetros

<code>COUPLER_HANDLER</code>	<code>CouplerHandler</code>	Entrada	Handler associado ao leitor
<code>unsigned char</code>	<code>AccMode</code>	Entrada	Modo de acesso: = Personalização; = Carregamento; = Validação
<code>unsigned char</code>	<code>SID</code>		Nº de identificação curto do ficheiro
<code>unsigned char</code>	<code>LID</code>		Nº de identificação longo do ficheiro
<code>unsigned char</code>	<code>NKEY</code>	Entrada	Posição, no SAM, da chave a utilizar
<code>unsigned char *</code>	<code>Rec</code>	Entrada	Dados a escrever;
<code>unsigned char</code>	<code>RecSize</code>	Entrada	Comprimento dos dados a escrever;

Retorno:

<code>unsigned long</code>	<code>= 0</code>	Função executada com sucesso
	<code>!= 0</code>	Função com erro.

Ver também:

3.4.8.3.6. READask_CALYPSO_Decrease

Descrição:

Permite descontar um determinado número de unidades num contador no cartão com microprocessador.

Protótipo:

```
unsigned long READask_CALYPSO_Decrease( COUPLER_HANDLER CouplerHandler,
```

```
, unsigned char  AccMode
, unsigned char  SID
, unsigned char  LID
, unsigned char  NKEY
, unsigned char  ICount
, unsigned long  Value
, unsigned long  *NewValue );
```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler associado ao leitor.
unsigned char	AccMode	Entrada	Modo de acesso: = Personalização; = Carregamento; = Validação
unsigned char	SID		Nº de identificação curto do ficheiro
unsigned char	LID		Nº de identificação longo do ficheiro
unsigned char	NKEY	Entrada	Posição, no SAM, da chave a utilizar
unsigned char	ICount	Entrada	Índice do contador;
unsigned long	Value	Entrada	Valor a decrementar;
unsigned long *	NewValue	Saída	Novo valor do contador (fora do modo de sessão);
Retorno:			
unsigned long	= 0		Função executada com sucesso
	!= 0		Função com erro.

Ver também:

3.4.8.3.7. READask_CALYPSO_ChangePIN

Descrição:

Altera o PIN (Personal Identification Number) de um cartão com microprocessador.

Protótipo:

```

unsigned long READask_CALYPSO_ChangePIN(    COUPLER_HANDLER CouplerHandler
                                           , unsigned char    NKEY
                                           , unsigned char    *OldPIN
                                           , unsigned char    *NewPIN
                                           , unsigned char    P1 );

```

Parâmetros

COUPLER_HANDLER	CouplerHandler	Entrada	Handler associado ao leitor
unsigned char	NKEY	Entrada	Posição, no SAM, da chave a utilizar
unsigned char *	OldPIN	Entrada	PIN actual;
unsigned char *	NewPIN	Entrada	Novo PIN;
unsigned char	P1	Entrada	Configuração do tipo de cartão

Retorno:

unsigned long	= 0	Função executada com sucesso
	!= 0	Função com erro.

Ver também:

4. ANEXO B - EXEMPLO DE UTILIZAÇÃO

De seguida apresenta-se um exemplo, em C, de como se pode utilizar a API Andante, pressupondo a utilização de um leitor da ASK, pelo que se elucida o modo de utilização da API da ASK e a sua interligação com a API Andante. Este exemplo foca essencialmente quais devem ser os procedimentos de inicialização na utilização das APIs.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include "READask.h"
```

```
#include "Andante.h"
```

```
#include "WINcalls.h"
```

```
HANDLE hCOM = INVALID_HANDLE_VALUE;
```

```
COUPLER_HANDLER ghReader = 0;
```

```
ANDANTE_HANDLER ghAntena = 0;
```

```
ANDANTE_HANDLER ghSAM = 0;
```

```
ANDANTE_HANDLER ghCard = 0;
```

```
char CurSerialNumber[12];
```

```
int InitOS(void)
```

```
{
```

```
    unsigned long ret = 0;
```

```
    ret |= Andante_SetCallback(0
```

```
        , CALLBACK_OS_MALLOC
```

```
        , (TYPE_CALLBACK_FUNCTION)malloc
```

```
, 0 );
```

```
ret |= Andante_SetCallback( 0
```

```
, CALLBACK_OS_FREE
```

```
, (TYPE_CALLBACK_FUNCTION)free
```

```
, 0 );
```

```
ret |= Andante_SetCallback( 0
```

```
, CALLBACK_OS_SPRINTF
```

```
, (TYPE_CALLBACK_FUNCTION)sprintf
```

```
, 0 );
```

```
return( (ret == 0) );
```

```
}
```

```
int OpenReaderCommunication(void)
```

```
{
```

```
DCB dcb;
```

```
if (hCOM != INVALID_HANDLE_VALUE) { CloseHandle(hCOM);
```

```
hCOM = INVALID_HANDLE_VALUE;
```

```
}
```

```
hCOM = CreateFile( "COM4"
```

```
, GENERIC_READ | GENERIC_WRITE
```

```
        , 0

        , NULL

        , OPEN_EXISTING

        , FILE_ATTRIBUTE_NORMAL

        , NULL );

if (hCOM == INVALID_HANDLE_VALUE) {
printf("Erro em \"CSC_Open\\\"\\n");

return(0);

}

ZeroMemory(&dcb,sizeof(DCB));

dcb.BaudRate = 115200;
dcb.ByteSize = 8; dcb.Parity
= NOPARITY; dcb.StopBits =
ONESTOPBIT;

dcb.fAbortOnError = FALSE;

if(!SetCommState(hCOM,&dcb)) {
printf("Erro em \"SetCommState\\\"\\n");

return(0);

}

if(!SetupComm(hCOM,512,512)) {
printf("Erro em \"SetupComm\\\"\\n");

return(0);
```



```
}
```

```
return(1);
```

```
}
```

```
int InitReader(void)
```

```
{ unsigned long ret
```

```
= 0;
```

```
char buf[500];
```

```
if ( !OpenReaderCommunication() ) return(0);
```

```
if (ghReader) {
```

```
    READask_Delete(ghReader);
```

```
ghReader = 0;
```

```
}
```

```
ghReader = READask_New(0);
```

```
if (!ghReader) {
```

```
    printf("Erro em \"READask_New\"\\n");
```

```
return(0);
```

```
}
```

```
ret |= READask_SetCallback( ghReader

    , CALLBACK_DELAY

    , (TYPE_READASK_CALLBACK_FUNCTION)Delay

    , (long)hCOM );

ret |= READask_SetCallback( ghReader

    , CALLBACK_FLUSH

    , (TYPE_READASK_CALLBACK_FUNCTION)FlushTxRx

    , (long)hCOM );

ret |= READask_SetCallback( ghReader

    , CALLBACK_TX

    , (TYPE_READASK_CALLBACK_FUNCTION)Tx

    , (long)hCOM );

ret |= READask_SetCallback( ghReader

    , CALLBACK_RX

    , (TYPE_READASK_CALLBACK_FUNCTION)Rx

    , (long)hCOM );

if (ret) {

    printf("Erro em \"READask_SetCallback\"\\n");

    return(0);

}
```

```
ret = READask_ResetReader(ghReader);

if (ret) {

    printf("Erro em \"READask_ResetReader\\n");

    return(0);

}

*buf = 0;

ret = READask_VersionCSC(ghReader, buf);

if (ret) {

    printf("Erro em \"READask_VersionCSC\\n");

return(0)
; }

printf("%s\\n", buf);

return(1);

}
```

```
int InitSAM(void)

{

    unsigned char atrSAM[255];
    unsigned long atrSAMLlen;
```

unsigned long ret;

atrSAMLen = sizeof(atrSAM);

ret = READask_ResetSAM(ghReader, atrSAM, &atrSAMLen);

if (ret) {

printf("Erro em \"READask_ResetSAM\\n");

return(0);

}

if (ghSAM) {

Andante_DeleteSAM(ghSAM);

ghSAM = 0;

}

ghSAM = Andante_NewSAM(atrSAM, (unsigned char)atrSAMLen, ANDANTE_SAM_CP); if
(!ghSAM) {

printf("Erro em \"Andante_NewSAM\\n");

return(0);

}

ret = Andante_SetCallback(ghSAM

, CALLBACK_SAM_APDU

, (TYPE_CALLBACK_FUNCTION)READask_SAM_TxRxAPDU

, (long)ghReader); if

(ret) {

printf("Erro em \"Andante_SetCallback\\n");

return(0);

```
}
```

```
return(1);
```

```
}
```

```
int InitAntenna(void)
```

```
{
```

```
    unsigned long ret = 0;
```

```
    if (ghAntena) {
```

```
        Andante_DeleteAntenna(ghAntena);
```

```
        ghAntena = 0;
```

```
    }
```

```
    ghAntena = Andante_NewAntenna();
```

```
    if (!ghAntena) {
```

```
        printf("Erro em \Andante_NewAntenna\\n");
```

```
    return(0);
```

```
    }
```

```
    ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CTS256B_READ
```

```
    , (TYPE_CALLBACK_FUNCTION)READask_CTS256B_Read
```

```
, (long)ghReader);
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CTS256B_UPDATE  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CTS256B_Update  
  
    , (long)ghReader);
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CTS512B_READ  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CTS512B_Read  
  
    , (long)ghReader);
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CTS512B_UPDATE  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CTS512B_Update  
  
    , (long)ghReader);
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CTS512B_WRITE  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CTS512B_Write  
  
    , (long)ghReader);
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_MFLIGHT_READ  
  
    , (TYPE_CALLBACK_FUNCTION)READask_MFlight_Read  
  
    , (long)ghReader);
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_MFLIGHT_UPDATE  
  
    , (TYPE_CALLBACK_FUNCTION)READask_MFlight_Update  
  
    , (long)ghReader );
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_MFLIGHT_WRITE  
  
    , (TYPE_CALLBACK_FUNCTION)READask_MFlight_Write  
  
    , (long)ghReader );
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CALYPSO_APDU  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_TxRxAPDU  
    , (long)ghReader );
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CALYPSO_OPEN  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_OpenSession  
  
    , (long)ghReader );
```

```
ret |= Andante_SetCallback( ghAntena  
, CALLBACK_CALYPSO_CLOSE  
  
    , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_CloseSession  
  
    , (long)ghReader );
```

```
ret |= Andante_SetCallback( ghAntena  
  
    , CALLBACK_CALYPSO_UPDATE_REC
```

```
        , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_UpdateRecord

        , (long)ghReader );

ret |= Andante_SetCallback( ghAntena

        , CALLBACK_CALYPSO_APPEND_REC

        , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_AppendRecord

        , (long)ghReader );

ret |= Andante_SetCallback( ghAntena

        , CALLBACK_CALYPSO_DECREASE

        , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_Decrease

        , (long)ghReader );

ret |= Andante_SetCallback( ghAntena

        , CALLBACK_CALYPSO_CHANGEPIN

        , (TYPE_CALLBACK_FUNCTION)READask_CALYPSO_ChangePIN

        , (long)ghReader );

if
(ret)
{
    printf("Erro em \Andante_SetCallback\\n");
return(0);

}

return(1);

}
```



```
int API_Init(void)
```

```
{
```

```
    *CurSerialNumber = 0;
```

```
    if ( !InitOS() ) return(0);
```

```
    if ( !InitReader() )
```

```
        return(0); if ( !InitSAM()
```

```
    ) return(0);
```

```
    if ( !InitAntenna() ) return(0);
```

```
    return(1);
```

```
}
```

```
void API_Close(void)
```

```
{
```

```
    if (ghCard) {
```

```
        Andante_DeleteCard(ghCard);
```

```
    }
```

```
    if (ghAntena) {
```

```
        Andante_DeleteAntenna(ghAntena);
```

```
    }
```

```
    if (ghSAM) {
```

```
Andante_DeleteSAM(ghSAM);

}

if (ghReader) {

    READask_AntennaOFF(ghReader);

    READask_Delete(ghReader);

}

if (hCOM != INVALID_HANDLE_VALUE) {

    CloseHandle(hCOM);

}

}

void SearchCard()

{

    unsigned char TypeCardFound; unsigned
char Atr[100]; unsigned char AtrLen;
unsigned long Ret;

    char SerialNumber[12];

    READask_AntennaOFF(ghReader);

    AtrLen = sizeof(Atr);

    Ret = READask_SearchCard(ghReader
```

```
        , READASK_DETECT_ISO_B

        | READASK_DETECT_CTS256B

        | READASK_DETECT_CTS512B

        , &TypeCardFound

        , Atr
&AtrLen ); if (Ret) {

    *CurSerialNumber = 0;

    return; }

if (AtrLen == 0) {    *CurSerialNumber = 0;

    return;

}

Ret = Andante_GetCardInfo(Atr, AtrLen, SerialNumber); if (Ret) {

    *CurSerialNumber = 0;

    return;

}

if ( !strcmp(SerialNumber, CurSerialNumber) ) return; strcpy(CurSerialNumber, SerialNumber);

printf("\rCartao: %s\n", SerialNumber);

if (ghCard) {

    Andante_DeleteCard(ghCard);

    ghCard = 0;

}
```

```
    ghCard = Andante_NewCard(ghAntena, Atr, AtrLen);    if (!ghCard)
return;

    Ret = Andante_SetSAM(ghCard, ghSAM);

    if (!Ret) return;

}
```

```
int main(int argc, char* argv[])

{

    BOOL Ok;    int Count;

    argc = argc;

    argv = argv;

    printf("Tecla 't' para terminar...\n");

    printf("=====\n");

    Ok = API_Init();

    Count = 0;    for (;;) {    if ( _kbhit() )
{    if (_getch() == 't') {    break;

        }

    }

}
```

```
switch (Count) { case 0: printf("\r-"); break; case  
1: printf("\r\\"); break; case 2: printf("\r|"); break;  
case 3: printf("\r/"); break;  
  
}  
  
Count = (Count + 1) % 4;  
  
  
if (Ok) SearchCard();  
  
  
Sleep(100);  
  
}  
  
  
API_Close();  
  
return(0);  
  
}
```

ANEXO E “Interface de Dados entre SBO e SCiiM”

Trata-se de uma interface de dados entre SBO e SCiiM (PCGBi) particularmente alinhada com o Modelo de Dados e API Andante.

Ficha Técnica

História

Versão	Autor	Organização	Criação	Actualização
01.01	Carlos Tomás Henrique Parente José Cruz José João José Jesus Luís Gama	Octal	2004-01-03	Primeira versão
01.02	Carlos Tomás José Cruz José Jesus Luís Gama	Octal	2004-02-10	Revisão segundo norma ISO 14904
01.03	Carlos Tomás Henrique Parente José Cruz Luís Faria Luís Gama	Octal	2004-03-25	Revisão para Domínios; Listas Negras; Configuração
01.04	Carlos Tomás José Cruz Luís Faria Luís Gama	Octal	2004-04-20	Revisão Configuração
01.05	Carlos Tomás José Cruz Luís Gama	Octal	2004-05-20	Revisão a elementos de: Fiscalização; Fidelização de Clientes; Parametrização de Equipamentos
01.06	Carlos Tomás José Cruz	Octal	2004-07-05	Informação da viagem em validadores embarcados
01.07	José Cardoso	Octal	2011-05-12	Informação de Embedder code e layout
01.08	José Cardoso Aurélio Garcia Pedro Lopes	Novabase TIP	2014-05-13	Validação estendida
01.09	José Cruz Henrique Parente Marco Alves	Card4B	2019-04-04	Registo de validação retornado pela API Andate

Nome	Organização	Contribuição

Propriedades

Os dados desta tabela, assim como os da capa, cabeçalho e rodapé, são actualizados automaticamente a partir das propriedades do documento (sumário) e da informação de sistema, pelo que não devem ser alterados manualmente.

Responsável	Processador de texto
Luís Gama	Microsoft Office Word
Gestor	Ficheiro
	RELATÓRIO_TIPITESP040107.DOC
Nº. de palavras / Nº. De caracteres	Actualização (última gravação)
6.958 / 48.498	2019-04-04 / 18:03
Palavras chave	Distribuição (última impressão)
TIP. Sistema Intermodal. Interfaces	2019-06-12 / 12:42

Documentos Relacionados

Título	Edição	Precedência	Conteúdo

Convenções

Notação	Significado
Negrito	Títulos
<u>Sublinhado</u>	Partes de texto a realçar
<i>Itálico</i>	Definição de termos
Courier	Aspectos técnicos, como exemplos de utilização ou de programação
AAAA-MM-DD	Datas

Termo	Significado
MVA	Máquina de venda automática
MVA – S	Máquina de venda automática simples
MVM	Máquina de venda manual assistida por operador
MPC	Máquina de personalização de cartões
MVP	Máquina de venda portátil assistida por operador
VAL	Máquina de validação
VAL-E	Máquina de validação embarcada
MPF	Máquina portátil de fiscalização
MB	Multibanco
MCO	Máquina Controlador de Operador
MCL	Máquina de controlador local
PCGBi	Posto Central de Gestão de Bilhética Intermodal

Índice

1	Introdução	4
2	Arquitectura	5
2.1	Princípios fundamentais	5
2.2	Mensagens entre Sistema e Equipamentos	6
3	Interfaces	7
3.1	Tipos de Dados	7
3.2	Mensagens	7
3.2.1	Cabeçalho	8
3.2.2	Formato da Mensagem	8
	Classe de Mensagem	8
	Tipo de Mensagem	9
3.2.3	Corpo da Mensagem	9
	Transacção	12
	Estado Do Equipamento	19
	Evento	20
	Lista de Estados	22
	Lista de Títulos	23
	Tabela de textos	25
	Tarifário	26
	Combinações	26
	Menus	27
	Utilizadores e autorizações	28
	Topologia	30
	Fidelização	30
	Parametrização de Validadores	31
3.2.4	Corpo da Mensagem de Recepção	34
	Transacções	34
	Estado do Equipamento	34
	Evento de Excepção	35
3.3	Exemplos XML	35
3.3.1	Transacção	35
3.3.2	Informação	37
3.3.3	Evento	38
3.3.4	Lista Negra	38
3.3.5	Títulos e Tarifário	39
3.3.6	Configuração	41
3.3.7	Parametrização de Validadores	43

1 INTRODUÇÃO

A integração de informação entre vários Operadores de Transporte e respetivos Sistemas de Bilhética pressupõe a existência de uma plataforma comum de comunicação que possibilite a troca da informação e respetiva interpretação. Em particular e no âmbito do Sistema de Bilhética Intermodal será necessário garantir a troca coerente dos conceitos com os vários operadores de transportes envolvidos.

Neste documento apresenta-se a especificação da interface suportada pelo PCGB Intermodal para comunicação com os diversos equipamentos que constituem o sistema de bilhética intermodal para os casos dos operadores rodoviários e ferroviários incluindo também a validação do tipo vflat ou validação estendida.

Tal como no caso dos restantes mecanismos de interoperabilidade, como sejam o Modelo de Dados e a respetiva API, a presente especificação poderá sofrer alterações/extensões, no sentido de melhor a adequar a requisitos funcionais futuros.

2 ARQUITECTURA

2.1 PRINCÍPIOS FUNDAMENTAIS

O Sistema de Bilhética Intermodal dos TIP tem de possibilitar a integração da informação proveniente de todos os equipamentos de bilhética (validadores, POS, MVA e outros), independentemente do operador onde estejam instalados.

Para que isto seja possível foi definido uma interface composta por um conjunto de mensagens que abrangem todas os diferentes tipos de transações existentes no sistema:

- Vendas;
- Validações;
- Fiscalizações;
- Eventos;
- Títulos;
- Tarifários;
- Listas Negras;
- Permissões de utilizadores;
- Configuração de interfaces com os clientes.

Os diversos equipamentos de bilhética terão de disponibilizar uma interface compatível com o especificado neste documento, para o conjunto das transações que suportam.

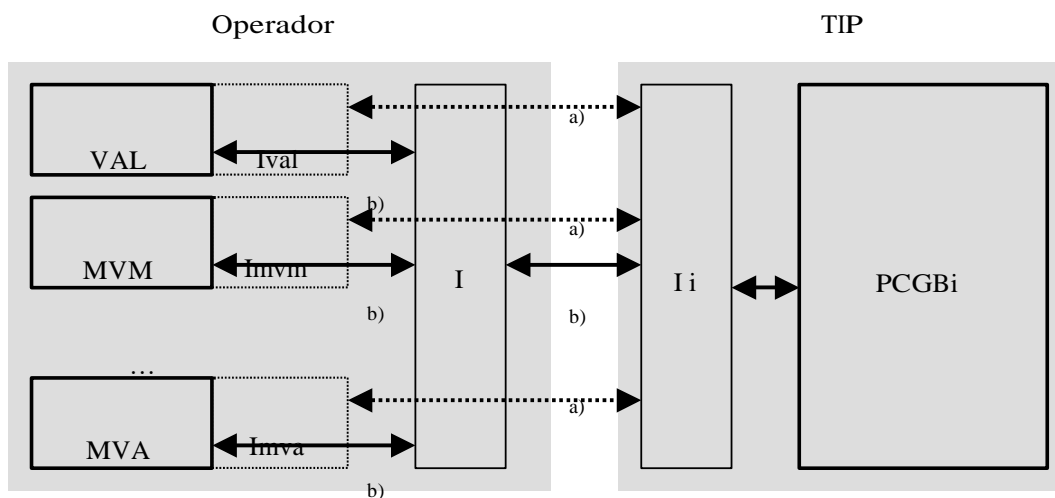


Figura 1 – Interfaces entre Sistema Bilhética Intermodal e equipamentos de bilhética

A figura 1 documenta duas aproximações distintas para implementação da interface com o PCGB Intermodal:

- a) Situação em que a interface é garantida pelos próprios equipamentos de bilhética (Ival, Imvm, Imva, ...), pelo que estes podem comunicar directamente como PCGB;
- b) Situação em que os equipamentos de bilhética são ligados através de um concentrador intermédio, Concentrador Local ou de Operador – MCL ou MCO, sendo este o responsável por garantir a interface com o PCGB.

2.2 MENSAGENS ENTRE SISTEMA E EQUIPAMENTOS

A troca de informação entre equipamentos e o PCGB é realizada através da troca de mensagens. Estas mensagens utilizam a linguagem XML, proporcionando um mecanismo aberto e normalizado, com aplicações reconhecidas na troca de informação entre sistemas e entidades (Business to Business). Esta interface pode ser facilmente desenvolvida por qualquer sistema recorrendo a um largo conjunto de ferramentas de desenvolvimento disponíveis, sendo independente da plataforma dos sistemas a integrar.

3 INTERFACES

Definem-se neste capítulo as mensagens a trocar entre os equipamentos e o PCGB. É apresentado para cada elemento (ou campo) da mensagem uma descrição, se é obrigatório e o seu tipo de dados.

3.1 TIPOS DE DADOS

Na tabela seguinte apresentam-se os tipos de dados possíveis para o conteúdo de cada campo:

Nome	Descrição
Boleano	Especifica um valor boleano que pode tomar os valores verdadeiro (S) ou falso (N)
DataHora	Representa um valor composto por data hora no formato AAAA/MM/DD HH:NN:SS AAAA: Ano (obrigatoriamente 4 dígitos) MM: Mês (obrigatoriamente 2 dígitos, 01 a 12) DD: Dia (obrigatoriamente 2 dígitos, 01 a 31) HH: Hora (obrigatoriamente 2 dígitos, 00 a 23) NN: Minutos (obrigatoriamente 2 dígitos, 00 a 59) SS: Segundos (obrigatoriamente 2 dígitos, 00 a 59)
I1	Numérico ≤ 255
I2	Numérico ≤ 65.535
I4	Numérico $\leq 4.294.967.295$
I<n>	Numérico <n> bytes. Valor $\leq (2^{(8*n)})-1$
N(n)	Numérico com <n> dígitos
N(n,m)	Numérico com <n> dígitos e <m> casas decimais
Texto(n)	Texto alfanumérico com tamanho máximo de n caracteres
Data	Representa uma data no formato AAAA/MM/DD AAAA: Ano (obrigatoriamente 4 dígitos) MM: Mês (obrigatoriamente 2 dígitos, 01 a 12) DD: Dia (obrigatoriamente 2 dígitos, 01 a 31)
Hex	Hexadecimal
B64	Dados binários representados no formato base 64

Tipo Variável	Descrição Variável
DTS	DateTime em Segundos desde 1997/01/01
AP	Automatic padding (para posicionar próximo campo alinhado ao byte)

3.2 MENSAGENS

As mensagens XML estão estruturadas segundo a norma ISO/TS 14904. Obedecendo a um protocolo específico são estruturadas em dois blocos principais: cabeçalho (header) e trama de mensagem (message frame).

Elemento	Descrição
Header	Informação com identificador da versão. Ver elemento Cabecalho .
Message Frame	Conjunto de dados divididos pelos seguintes elementos: Classe, Tipo, Identificação do emissor, Identificação do receptor, Identificação da mensagem e Corpo da mensagem. Ver elemento Formato da Mensagem .

3.2.1 **CABECALHO**

No cabeçalho da mensagem está identificado a sua versão permitindo, como primeiro elemento da mensagem, a sintonia entre os emissor e receptor da versão de protocolo usado.

Elemento	Descrição	Obriga- tório	Tipo de Dados
Message Header	Informação da versão do protocolo usado.	Sim	I1

3.2.2 **FORMATO DA MENSAGEM**

A mensagem (message frame) contém a informação principal dos dados a enviar ou receber e está dividida nos seguintes elementos:

Elemento	Descrição	Obriga- tório	Tipo de Dados
Message Class	Classe da mensagem. Identifica o objectivo da mensagem. Ver elemento Classe de Mensagem .	Sim	Texto(30)
Message Type	Identificação do tipo de mensagem a transmitir. Ver elemento Tipo de Mensagem .	Sim	Texto(20)
Sender ID	Identificação do Emissor da mensagem.	Sim	I1
Receiver ID	Identificação do Receptor da mensagem.	Sim	I1
Message ID	Identificação da Mensagem. Juntamente com o Emissor ou Receptor este elemento identifica univocamente uma mensagem.	Sim	I4
Message Body	Sequência de dados com a informação específica da mensagem a transmitir (classe Notification – ver elemento Corpo da Mensagem) ou para mensagens de resposta (classe Notification Acknowledgement – ver elemento Corpo da Mensagem de Recepção)	Sim	Elemento

Classe de Mensagem

Define o objetivo da mensagem. Serão usados as seguintes classes:

Classe de mensagem	Descrição
Notification	Classe que indica ao receptor que o emissor (ou terceiro) efectuou alguma acção. Esta classe será usada para os Tipos de Mensagem Transaction, Event Exception, Equipment Status e Status List.
Notification Acknowledgement	Classe de mensagem que pode ser enviada para uma resposta a uma mensagem da classe Notification.
Advice	Classe que contém informação essencial para o correcto funcionamento do sistema. Esta classe será usada para mensagens de configuração e do Tipo de Mensagem Services List e Fare Products List.
Advice Response	Classe de mensagem que pode ser enviada para uma resposta a uma mensagem da classe Advice.

Tipo de Mensagem

De acordo com os padrões e a norma já definida, existe um conjunto de Tipos de Mensagens, usados para agrupar certos conjuntos de mensagens, facilitando a sua interpretação.

Neste contexto e para este propósito, serão utilizados os seguintes tipos de mensagens:

Tipo de mensagem	Descrição
Transaction	Conjunto de informação com o detalhe de transações associadas ao operador de transporte. Inclui registos de Venda, Documentos e Validações.
Event Exception	Informação de eventos nos diversos equipamentos.
Equipment Status	Registos com de estado de equipamentos.
Status List	Tipo de mensagem usado para envio de informação relativo ao estado de um utente, fornecedor (Provider) ou outro operador do sistema.
Fare Products List	Tipo de mensagem a usar para envio de informação dos Títulos e Tarifário
Services List	Tipo de mensagem a usar para envio de outros dados de configuração

3.2.3 CORPO DA MENSAGEM

Aqui apresentam-se os elementos nucleares usados para transmissão da informação.

Elemento	Descrição	Obrigatório	Tipo de Dados
DateIssued	Data de geração da mensagem	Sim	DataHora
Transactions	Colecção de transacções incluídas na mensagem onde se incluem dados de vendas e validações		Elemento
EquipmentStatus	Informação sobre o estado dos equipamentos		Elemento
EventException	Informação sobre os eventos gerados pelos equipamentos		Elemento
StatusList	Colecção de listas de títulos a colocar na Lista Negra		Elemento
FareProductsList	Informação dos tipos de título , descritivos , respectivo tarifário e regras de associação de títulos e suportes		Elemento
ServicesList	Informação de outros dados de configuração como Menus de equipamentos, descritivos , utilizadores , topologia do sistema, fidelização de clientes e parametrização de validadores		Elemento

Deve existir sempre um dos elementos Transactions, EquipmentStatus, EventException, StatusList, FareProductsList ou ServicesList. Cada um deles é usado para enviar ou receber vários tipos de mensagens (carregamento, venda, validação, listas negras, etc.). A utilização de cada um destes elementos é exemplificado na tabela seguinte:

Mensagem	Venda	Carregamento	Validação	Invalidação	Fiscalização	Evento	Estado do Equipamento	Listas Negras	Tarifário	Configuração
Elemento										
Transaction	✓	✓	✓	✓	✓					
EquipmentStatus							✓			
EventException						✓				
StatusList								✓		
FareProductsList									✓	
ServicesList										✓

Para cada um dos elementos Transaction, EquipmentStatus, EventException, StatusList, FareProductsList ou ServicesList a combinação dos elementos a utilizar depende igualmente do tipo de mensagem a enviar/receber. Esta combinação de elementos é apresentada na tabela seguinte:

Mensagem	Venda	Carrega- mento	Valida- ção	Invali- dação	Fisca- liza- ção	Even- to	Esta- do do Equi- pa- mento	Listas Ne- gras	Tari- fário	Confi- gura- ção
Elemento										
Load		✓								
Shift	✓	✓	✓	✓	✓					
Ticket	✓ ¹	✓ ²	✓	✓ ³	✓					
Equipment	✓	✓	✓	✓	✓	✓	✓			
PaymentMean	✓	✓								
Receipt	✓	✓								
CreditNote	✓	✓								
Discount	✓	✓								
Validation			✓ ⁴	✓ ⁵						
Fiscalization					✓					
TicketList									✓	
TextTable									✓	✓
PriceList									✓	
CombineRules									✓	
Menu										✓
UserList										✓
Topology										✓

¹ No caso de uma venda sem carregamento a informação do título só contém os dados do suporte (número de série, tipo de cartão).

² No caso de um carregamento a informação relacionada com viagens ou zonas carregadas, provider do título, tarifário e preço são transmitidas no elemento carregamento.

³ No caso de uma invalidação pode não existir a informação relacionado com o título.

⁴ No caso de uma validação o valor do atributo ValidResultCode deve ser 0.

⁵ No caso de uma invalidação o valor do atributo ValidResultCode deve ser 1.

Transacção

Elemento	Descrição	Obriga- tório	Tipo de Dados
Type	Tipo de mensagem (0 - Load, 1 – Valid, 2– Fiscalização)	Sim	N(1)
Version	Versão da estrutura da mensagem. 1- Validação standard; 2- validação estendida	Sim	N(2)
ID	Identificador único e sequencial da transacção	Sim	N(10)
TimeStamp	Instante de ocorrência da transacção	Sim	DataHora
Shift	Identificação do turno onde foi gerada a transacção.		Elemento
Equipment	Identificação do equipamento onde foi realizada a operação (Venda, carregamento, Validação, Fiscalização....)	Sim	Elemento
Items	Colecção de elementos do tipo Item .	Sim	Elemento
Payment	Informação do pagamento . Este elemento deve estar presente sempre que se trate de uma mensagem do tipo Load, no caso de mensagens do tipo Valid não deve existir		Elemento
IssuedDocs	Documentos emitidos na transacção. Este elemento deve estar presente sempre que se trate de uma mensagem do tipo Load, no caso de mensagens do tipo Valid não deve existir		Elemento

Turno

Informação que permite a identificação do turno de determinada transacção, onde pode ser eventualmente registado o operador (utilizador) para efeitos de acções de fiscalização ou validação.

Elemento	Descrição	Obriga- tório	Tipo de Dados
ShiftCode	Código do Turno	Sim	N(12)
StartDate	Data em que o Turno foi iniciado	Sim	DataHora
EndDate	Data em que o Turno foi terminado	Sim	DataHora

Equipamento

Em termos de codificação e mapeamento do Equipamento, o código deve ser um registo do Tipo Equipamento duma entidade genérica (chamada Domínio) existente no PCGB Intermodal.

Tal como o Equipamento, a informação do Local de instalação do equipamento deve ser também um dos elementos da entidade genérica Domínio.

Elemento	Descrição	Obrigatório Ver.=1	Obrigatório Ver.=2	Tipo de Dados
Code	Código do equipamento	Sim	Sim	I4
SerialNumber	Número de série do equipamento	Sim	Sim	Texto(20)
Local	Local da instalação do equipamento	Sim	Sim	Elemento
ProviderCode	Operador do local	Sim	Sim	N(2)
Zones	Zonas atribuídas ao local			Elemento
SwVersion	Versão de software do validador	Não	Não	Texto(20)
ValidatorPos	Endereços de polling do validador, nos casos em que um autocarro tenha mais do que um validador estes podem ser distinguidos entre si pelo seu endereço de polling. Valor 1 para o 1º validador, valor 2 para o 2º validador	Não	Sim	N(4)
Vehicle	Número do veículo onde foi realizada a validação	Não	Sim	N(4)

Locais

Em termos de codificação e mapeamento, o Local pode reflectir informação que depende do tipo de equipamento, características e tipologia de instalação. Exemplos de locais são estações, átrios, vias, paragens, viaturas, etc. A codificação é definida pelo PCGB Intermodal.

Existe, no entanto, um conjunto de informação que poderá variar para o mesmo local. As Linhas e Direcção são disso um exemplo já que, para instalações embarcadas de equipamentos, essa informação depende do percurso efectuado.

Elemento	Descrição	Obrigatório Ver=1	Obrigatório Ver=2	Tipo de Dados
Code	Código da paragem onde foi realizada a validação	Sim	Sim	I4
Lines	Conjunto de linhas atribuídas ao local	Sim	Sim	Elemento
Direction	Sentido (0 – Ascendente, 1 – Descendente)			N(1)
Variante	Variante da linha	Não	Sim	N(1)
TsIniViagem	Data/Hora de início da viagem do autocarro	Não	Sim	DTS30

Linha

Elemento	Descrição	Obrigatório	Tipo de Dados
Line	Linha atribuída ao local	Sim	N(3)

Zona

Elemento	Descrição	Obrigatório	Tipo de Dados
Zone	Zona atribuída ao local	Sim	Texto(4)

Item

Elemento	Descrição	Obrigatório	Tipo de Dados
Load	Informação do carregamento efectuado no título		Elemento
Ticket	Informação relativa ao título vendido e/ou carregado. Se a informação relativa ao carregamento existir então existiu uma venda e um carregamento, caso contrário apenas foi vendido o título.		Elemento
Validation	Informação sobre validações ou invalidações realizadas. No caso das invalidações pode não existir a informação do título.		Elemento
Fiscalization	Registo das fiscalizações efectuadas		Elemento
AndanteValidation	Registo de validação retornado pela API Andante		B64

Carregamento

Elemento	Descrição	Obrigatório	Tipo de Dados
ReloadCount	Contador de carregamentos. Valor retornado pela API Andante no acto de carregamento de um título de transporte num cartão	Sim	N(3)
ContractCopy	Cópia do contracto		Elemento
ValidDate	Período de validade .	Sim	Elemento
ReductionRate	Código da taxa de redução aplicada no título.	Sim	N(2)
ProviderCode	Código do provider do título	Sim	N(3)
Trips	Informação das viagens carregadas e disponíveis após o carregamento		Elemento
ZoneInfo	Informação relativa às zonas do título afectadas pelo carregamento		Elemento

Deve existir obrigatoriamente um dos elementos Trips ou ZoneInfo.

Cópia do contracto

Elemento	Descrição	Obrigatório	Tipo de Dados
ContractIdx	Identificação do contracto.	Sim	N(10)
ContractContent	Conteúdo do contracto	Sim	Hex

Período de Validade

Elemento	Descrição	Obrigatório	Tipo de Dados
StartDate	Data de início de validade do título	Sim	Data
EndDate	Data de fim de validade do título	Sim	Data

Viagens

Elemento	Descrição	Obrigatório	Tipo de Dados
Used	Número de viagens utilizadas do título		N(2)
Total	Número de viagens disponíveis no título		N(2)
Load	Número de viagens carregadas no título		N(3)

Os atributos do elemento Viagens devem ser utilizados de acordo com o elemento onde o elemento é utilizado.

	Carregamento	Validação
Used		✓
Total	✓	✓
Load	✓	

Zonas do Título

Elemento	Descrição	Obrigatório	Tipo de Dados
NumberOfZones	Número de zonas	Sim	N(3)
ValidZones	Lista das zonas de validade do título separadas por ponto e virgula	Sim	Texto (1000)

Título

Elemento	Descrição	Obriga- tório	Tipo de Dados
TicketCode	Código do tipo de título	Sim	N(6)
SerialNumber	Número de série do título	Sim	N(32)
CardType	Tipo de suporte do título 1: Andante Blue (CTS256B) 2: Andante Gold (GTML2) 3: Andante Blue (CTS512B) ...	Sim	N(1)
Profile	Código do desconto do título (Criança, 3ª Idade)		N(1)
ValidDate	<u>Período de validade</u> do título		Elemento
ReductionRate	Taxa de desconto.		N(2)
ProviderCode	Provider do título.		N(3)
Trips	Informação relativa às <u>viagens</u> do Título		Elemento
ZoneInfo	Informação sobre as <u>zonas de validade</u> do título		Elemento
ContractCopy	<u>Cópia do contracto</u>		Elemento
Price	Valor do título		N(5,2)
TariffCode	Código da tarifa do título		N(2)
Priority	0- Título normal; 1 - KlikZ		
EmbedderCode	Código do Fabricante do suporte		N(2)
Layout	Layout do Suporte		N(2)
HolderCompany	Código da empresa, este campo só estará preenchido no caso de validações realizadas com títulos Andante Empresa.		N(1)
DataModel	Versão do modelo de dados utilizados na validação		N(1)
Origem	Campo onde pode ser registada o ponto de entrada		N(1)
Destino	Campo onde pode ser registada o ponto de saída		N(1)

O segmento Título é utilizado em várias mensagens, sendo alguns dos campos do segmento de utilização obrigatória consoante o tipo de mensagem a construir.

	Venda	Carregamento	Validação	Invalidação	Fiscalização
TicketCode	✓	✓	✓	✓	
SerialNumber	✓	✓	✓	✓	✓
CardType	✓	✓	✓	✓	
Profile			✓	✓	
ValidDate			✓	✓	
ReductionRate		✓	✓	✓	
ProviderCode		✓	✓	✓	
Trips		✓	✓	✓	
ZoneInfo		✓	✓	✓	
ContractCopy			✓	✓	
Price			✓	✓	
TariffCode			✓	✓	
Priority			✓	✓	
EmbederCode		✓			
Layout		✓			
HolderCompany		✓	✓	✓	
DataModel			✓	✓	
Origem			✓	✓	
Destino			✓	✓	

Validação

Elemento	Descrição	Obrigatório	Tipo de Dados
Action	2 – Entrada 3 - Trânsito 4 – Saída	Sim	N(1)
TimeStamp	Data / Hora da validação	Sim	DataHora
ZoneUsed	Zona de validação	Sim	Texto(4)
ValidResultCode	Resultado da validação (0-Validação efectuada com sucesso, 1-Validação efectuada sem sucesso)	Sim	N(1)
TripNumber	Número da viagem onde foi efectuada a validação.		N(3)
TripFinal	Data / Hora do fim da viagem	Sim	DataHora

Pagamento

Elemento	Descrição	Obrigatório	Tipo de Dados
Amount	Quantia paga	Sim	N(6,2)
PaymentMeans	Especificação dos meios de pagamento utilizados neste pagamento	Sim	Elemento
Discount	Desconto realizado no pagamento		Elemento

Meio de Pagamento

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de Meio de pagamento (0-Dinheiro, 1-Cheque, 2-Multibanco, 3-Nota de Crédito)	Sim	N(1)
DocNumb	Número do documento (Cheque, Nota de Crédito, ...)		Texto(20)
Amount	Valor em Euro	Sim	N(6,2)
EquipIssuer	Código da máquina que emitiu a nota de crédito		I4
DateIssue	Data de emissão (Nota de Crédito)		DataHora
MoreInfo	Informação complementar do meio de pagamento		Elemento

Conforme o tipo de pagamento alguns dos campos do segmento meio de pagamento são obrigatórios.

	Dinheiro	Cheque	Multibanco	Nota de Crédito
Type	✓	✓	✓	✓
DocNumb		✓		✓
Amount	✓	✓	✓	✓
EquipIssuer				✓
DateIssue				✓
MoreInfo		✓		

Informação complementar do meio de pagamento

Elemento	Descrição	Obrigatório	Tipo de Dados
Info	Valor do campo	Sim	Texto(15)
Type	0-BI, 1-Telefone	Sim	N(1)

Desconto

Elemento	Descrição	Obrigatório	Tipo de Dados
ReasonCode	Código da razão do desconto	Sim	I1
ReasonTxt	Descrição do desconto	Sim	Texto(60)
Amount	Valor do desconto	Sim	I4
DocNumber	Identificação do documento que habilita o desconto		Texto(20)
EntityID	Numero de identificação “autor” da requisição. Este campo é obrigatório no caso dos descontos em transacções. Nos restantes casos não pode existir.		I4

Documento Emitidos

Elemento	Descrição	Obrigatório	Tipo de Dados
Receipt	Informação do recibo		Elemento
CreditNote	Informação da nota de crédito emitida		Elemento

Recibo

Elemento	Descrição	Obrigatório	Tipo de Dados
Amount	Valor, em Euro, do recibo emitido	Sim	N(6,2)
Number	Identificação do recibo. Esta identificação deve ser única no contexto de um operador, por equipamento de venda	Sim	I4

Nota de Crédito Emitida

Elemento	Descrição	Obrigatório	Tipo de Dados
Number	Identificação da nota de crédito emitida pelo equipamento de venda	Sim	Texto(20)
Amount	Valor da Nota de Crédito	Sim	N(6,2)

Fiscalização

Elemento	Descrição	Obrigatório	Tipo de Dados
Checks	Lista de verificações realizadas no período de fiscalização	Sim	Elemento

Verificações

Elemento	Descrição	Obrigatório	Tipo de Dados
Date	Data da fiscalização	Sim	DataHora
ZoneUsed	Zona da fiscalização	Sim	Texto(4)
Error	Erro na fiscalização	Sim	I4

Estado Do Equipamento

Este elemento permite obter informação sobre o estado do equipamento e respectivos módulos internos.

Elemento	Descrição	Obrigatório	Tipo de Dados
Status	Estado do Equipamento	Sim	Elemento

Estado

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de estado (0- InternalModule, ...)	Sim	N(1)
Version	Versão da estrutura da mensagem	Sim	N(2)
ID	Identificador do Estado	Sim	N(10)
TimeStamp	Data / Hora do estado	Sim	DataHora
Equipment	Dados do equipamento		Elemento
Module	Módulo interno do equipamento		Elemento

Módulo

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de Módulo: 'M': Moedas 'H': Depósito de trocos 'N': Notas 'S': Multibanco 'C': Leitor sem contacto 'D': Dispensador de bilhetes 'R': Recibos	Sim	Texto
SerialNumber	Número do série do módulo	Não	N(12)
NominalValue	Para os seguintes tipos de módulo: 'M', 'H', 'N' especifica o valor nominal, em cêntimos de euro, de cada uma das Moedas/Notas que sejam manipuladas pelos módulos. Para os tipos de módulo 'S' e 'P', especifica o valor total, em Euro.		N(5,2)
Quantity	Para os tipos de módulo 'S' e 'P', deve ser 1. Para os tipos de módulo 'M', 'H', 'N', indica a quantidade existente.		N(4)

Evento

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	0-On, 1-Off, 2-Info	Sim	N(1)
Version	Versão da estrutura da mensagem	Sim	N(2)
ID	Identificador do Evento	Sim	N(10)
TimeStamp	Data / Hora do evento	Sim	DataHora
EventType	Código do evento gerado pelo equipamento	Sim	N(3)
Equipment	Dados do equipamento		Elemento
Infos	Conjunto de registo para outras informações		Elemento

Informação de Evento

Elemento	Descrição	Obriga- tório	Tipo de Dados
Info	Informação Genérica	Sim	Texto(60)

Listade Estados

Elemento	Descrição	Obriga- tório	Tipo de Dados
List	Colecção de elementos a enviar para a Lista Negra	Sim	Elemento

Lista

Elemento	Descrição	Obriga- tório	Tipo de Dados
Type	Tipo de lista, no caso de uma lista negra este atributo deve ter o valor BlackList	Sim	Texto(15)
Version	Versão da lista	Sim	N(2)
StartDate	Início de entrada em vigor desta lista. Se ausente, tem efeitos imediatos	Sim	DataHora
EndDate	Fim de validade desta lista. Se não for especificado então a lista é válida para sempre ou até ao de uma nova lista.	Sim	DataHora
Items	Colecção de itens que compõe a lista	Sim	Elemento

Item

Elemento	Descrição	Obriga- tório	Tipo de Dados
ElementType	Tipo de elemento da lista, os valores possíveis para este atributo são: Card - elemento da lista negra é referenciado pelo seu número de série Profile.- elemento da lista negra é referenciado pelo perfil dos cartões TicketCode - elemento da lista negra é referenciado pelo código do título	Sim	Texto(15)
Type	Tipo de identificador utilizado no elemento, este atributo pode ter um dos seguintes valores: Element - elemento da lista negra identifica uma só entidade do ElementType Range - elemento da lista negra identifica uma gama de valores referentes à entidade do ElementType	Sim	Texto(15)
StartDate	Início de entrada em vigor do elemento da lista. Se ausente, herda valor da lista onde o elemento está.		DataHora
EndDate	Fim de validade do elemento da lista. Se ausente, herda valor da lista onde o elemento está.		DataHora
InfoText	Texto descritivo. Pode ser por exemplo a razão/descrição de acções a tomar.		Texto(60)
NumberOfElements	Este atributo só pode estar definido no caso de elementos com type = Range, caso contrário deverá ser ignorado. Neste caso, o atributo Id indica o primeiro "código" (da gama de valores) a colocar em lista negra.		N(3)
Id	Identificação do elemento da lista, o valor deste atributo depende do atributo ElementType.		Texto(15)

	Card - Numero de série do cartão (número completo do suporte, código do suporte concatenado com o número de série do cartão). Profile - Código do perfil no cartão (se existir) TicketCode - Código do título.		
ApplyToProvider	Engloba a lista de código de providers (dos equipamentos) onde o elemento da lista negra deve ser aplicado. Se ausente, engloba todos os providers		Elemento

Operador

Elemento	Descrição	Obrigatório	Tipo de Dados
Id	Código do provider onde o elemento da lista negra deve ser aplicado.	Sim	N(3)

Listade Títulos

Elemento	Descrição	Obrigatório	Tipo de Dados
Attribute	Conjunto de atributos existentes		Elemento
AttributeDescription	Conjunto de valores aceitáveis para os atributos e respectivos códigos dos textos descritivos . O código de texto identifica o texto que descreve um valor.		Elemento
FareProduct	Títulos existentes		Elemento

Atributo

Elemento	Descrição	Obrigatório	Tipo de Dados
AttributeCode	Código que identifica o atributo	Sim	I4
TextCode	Código que identifica o texto descritivo do atributo	Sim	I4
AttributeType	Tipo do atributo (A – atributo marcante, V – variável de venda, C – característica)		Texto(1)
DataType	Tipo de dado dos valores aceitáveis para o atributo (N – Numérico, C – Alfanumérico)		Texto(1)
Size	Para atributos numéricos, define o número de bits correspondente; para atributos alfanuméricos, define o número máximo de caracteres		I4
Sequence	Ordem em que o valor do atributo deve ser considerado ao construir a descrição do título		I4
Keyword	Palavra chave que identifica o atributo	Sim	Texto(10)

Descrição de Atributo

Elemento	Descrição	Obrigatório	Tipo de Dados
AttributeCode	Código que identifica o atributo	Sim	I4
Value	Valor do atributo	Sim	Texto(50)
TextCode	Código do texto descritivo correspondente ao valor em questão	Sim	I4

Título

Elemento	Descrição	Obrigatório	Tipo de Dados
TicketCode	Código que identifica o título, único em todo o sistema	Sim	I4
Description	Descritivo do título (construído à custa dos textos associados aos valores de alguns dos seus atributos)		Texto(120)
Active	Valor que indica se o título está activo (disponível para venda)	Sim	Texto(1)
ServiceCode	Código interno das MVAs para a emissão do título		I4
LayoutCode	Código para o layout gráfico do título, a utilizar caso haja lugar à impressão de títulos no momento da venda		I4
CodifCode	Código de codificação do título, a utilizar caso haja esquemas de codificação alternativos em função do título		I4
TicketAttributes	Valores dos atributos associados ao título		Elemento

Valores dos atributos

Elemento	Descrição	Obrigatório	Tipo de Dados
<Attribute_Keyword>	Valor para o atributo cuja keyword é identificada pelo nome do elemento. Ex: <PERFIL>1</PERFIL> define o valor '1' para o atributo identificado pela keyword 'PERFIL'	Sim	Texto(80)

Tabela de textos

Elemento	Descrição	Obriga- tório	Tipo de Dados
TextType	Tipos de texto existentes. Ex.: traduções para Inglês, Francês, etc. (equipamentos de venda automática), abreviaturas para diferentes dispositivos de interface - ecrã, display de cliente - (equipamentos de venda manual, validadores), texto para recibos, etc.	Sim	Elemento
Texts	Valores definidos para os vários textos existentes, e para cada tipo de texto.	Sim	Elemento

Tipo de Texto

Elemento	Descrição	Obriga- tório	Tipo de Dados
TextTypeID	Código que identifica o tipo de texto	Sim	I4
Description	Descrição		Texto(50)
Keyword	Palavra chave que identifica o tipo de texto. Ex.: MVA_UK, MVM_ECRA	Sim	Texto(10)
Sequence	No caso de se tratar de um tipo de texto que represente uma linguagem para uma máquina automática, este valor define a ordem em que a linguagem é disponibilizada na interface com o cliente.		I4

Texto

Elemento	Descrição	Obriga- tório	Tipo de Dados
TextCode	Código que identifica o texto	Sim	I4
TextValues	Valores que o texto assume para os tipos de texto que lhe estejam associados.	Sim	Elemento

Valor de texto

Elemento	Descrição	Obriga- tório	Tipo de Dados
<TextTypeKeyword>	Texto atribuído ao tipo de texto cuja palavra chave é identificada pelo nome do elemento. Ex.: <MVA_FR>2 Voyages</MVA_FR> define o valor '2 Voyages' para o tipo de texto com keyword MVA_FR.	Sim	Texto(80)

Tarifário

Elemento	Descrição	Obriga- tório	Tipo de Dados
TariffCode	Código que identifica o tarifário	Sim	I4
StartDate	Data a partir da qual o tarifário entra em vigor	Sim	Data
Description	Descritivo do tarifário		Texto(200)
Price	Conjunto de valores do preço e do IVA definidos para os títulos que se encontram associados ao tarifário	Sim	Elemento

Preço

Elemento	Descrição	Obriga- tório	Tipo de Dados
TicketCode	Código que identifica o título	Sim	I4
Price	Preço do título	Sim	N(7,2)
VAT	IVA aplicado ao título		N(7,2)

Combinações

Elemento	Descrição	Obriga- tório	Tipo de Dados
CombCode	Código que identifica a combinação	Sim	I4
Name	Nome atribuído à combinação	Sim	Texto(50)
Description	Descritivo da combinação		Texto(200)
Keyword	Keyword que identifica a combinação	Sim	Texto(10)
Attributes	Conjunto de atributos associados à combinação	Sim	Elemento
Values	Conjuntos de valores aceitáveis para os atributos associados à combinação	Sim	Elemento

Atributos de Combinação

Elemento	Descrição	Obriga- tório	Tipo de Dados
CombAttribID	Código que identifica a associação entre um atributo e uma combinação (um atributo pode estar presente numa combinação mais do que uma vez)	Sim	I4
AttributeCode	Código que identifica o atributo	Sim	I4

Detalhe de Combinação

Elemento	Descrição	Obrigatório	Tipo de Dados
DetailCode	Código que identifica um conjunto de valores para os atributos associados a uma combinação	Sim	I4
CombAtrib	Valor para o atributo em questão (o nome do elemento identifica o código da associação entre o atributo e a combinação(valor de CombAtribID, no Atributo de Combinação . Ex: o elemento com nome <CombAtrib3> define um valor para a associação com código 3.	Sim	Texto(50)

Menus

Elemento	Descrição	Obrigatório	Tipo de Dados
Menus	Menus existentes	Sim	Elemento
MenuTag	Conjunto de opções associadas a cada menu (MenuTag)	Sim	Elemento
MenuCommand	Comandos associados aos menus	Sim	Elemento
MenuCondition	Condicionantes associados às opções dos menus	Sim	Elemento

Menu

Elemento	Descrição	Obrigatório	Tipo de Dados
MenuCode	Código que identifica o menu	Sim	I4
Description	Descritivo (nome) do menu		Texto(50)
Object	Lista de objectos a apresentar ao utilizador (Som, Imagem)		Elemento

Lista de Objectos

Elemento	Descrição	Obrigatório	Tipo de Dados
ObjectType	Tipo do objecto (0 – Som, 1 – Imagem)	Sim	I4
Object	Objecto	Sim	B64

Tag de Menu

Elemento	Descrição	Obrigatório	Tipo de Dados
MenuCode	Código que identifica o menu	Sim	I4
TagCode	Código que identifica a opção no contexto do menu	Sim	I4
TextCode	Código que identifica o texto da opção		I4

Comando

Elemento	Descrição	Obrigatório	Tipo de Dados
CommandCode	Código que identifica o comando	Sim	I4
MenuCode	Código que identifica o menu	Sim	I4
KeyIn			Texto(10)
Command	Keyword que representa o tipo de comando		Texto(4)
Arguments	Lista de valores dos argumentos do comando. Os valores são separados por vírgulas.		Texto(20)
Sequence	Ordem em que o comando é executado relativamente aos restantes, associados ao menu		I4

Condicionantes

Elemento	Descrição	Obrigatório	Tipo de Dados
CommandCode	Código que identifica o comando	Sim	I4
MenuCode	Código que identifica o menu	Sim	I4
TagCode	Código que identifica a opção	Sim	I4
Command	Keyword que representa o tipo de comando		Texto(4)
Arguments	Lista de valores para os argumentos do comando		Texto(20)

Utilizadores e autorizações

Elemento	Descrição	Obrigatório	Tipo de Dados
Users	Utilizadores existentes	Sim	Elemento
Authorisations	Associações entre utilizadores e grupos (Autorização). Cada associação é definida no contexto de um ou mais elementos do domínio. Estas associações são agrupadas por domínio.	Sim	Elemento
Group	Grupos de utilizadores existentes	Sim	Elemento
Profile	Perfis de acesso existentes	Sim	Elemento

Utilizador

Elemento	Descrição	Obrigatório	Tipo de Dados
UserCode	Código que identifica o utilizador	Sim	I4
Username		Sim	Texto(10)
Password		Sim	Texto(50)
Name	Nome do utilizador	Sim	Texto(25)
NumericId	Valor que indica se a identificação (username / password) do utilizador é estritamente numérica	Sim	Boleano

Autorização

Elemento	Descrição	Obrigatório	Tipo de Dados
Domain	Conjunto de equipamentos para o qual a associação entre utilizadores e grupos é válida. (Domínio)	Sim	Elemento
Groups	Conjunto de grupos e respectivos utilizadores, válidos no domínio em questão (Grupo - Autorização)	Sim	Elemento

Domínio

Elemento	Descrição	Obrigatório	Tipo de Dados
EquipmentCode	Código que identifica o equipamento e que é único em todo o sistema	Sim	I4

Grupo - Autorização

Elemento	Descrição	Obrigatório	Tipo de Dados
GroupCode	Código que identifica o grupo	Sim	I4
Users	Conjunto de utilizadores pertencentes ao grupo ⁶	Sim	Elemento

Grupo

Elemento	Descrição	Obrigatório	Tipo de Dados
GroupCode	Código que identifica o grupo	Sim	I4
GroupName	Nome atribuído ao grupo	Sim	Texto(50)
Description			Texto(200)
Profiles	Conjunto de perfis de acesso associados ao grupo ⁷	Sim	Elemento

Perfil

Elemento	Descrição	Obrigatório	Tipo de Dados
ProfileCode	Código que identifica o perfil	Sim	I4
ProfileName	Nome do perfil	Sim	Texto(50)
Description			Texto(200)
EquipmentType	Tipo de equipamento para o qual o perfil foi definido (MVA, MVM,...)		Texto(10)
Functions	Conjunto de funcionalidades associadas ao perfil	Sim	Elemento

⁶ Identificados apenas pelo atributo UserCode.

⁷ Identificados apenas pelo elemento ProfileCode

Funcionalidade

Elemento	Descrição	Obrigatório	Tipo de Dados
FunctionName	Nome da funcionalidade	Sim	Texto(50)

Topologia

Elemento	Descrição	Obrigatório	Tipo de Dados
Entities	Conjunto de entidades lógicas (estação, linha, zona, operador, etc.), definidas no sistema (Elemento de sistema)	Sim	Elemento
Equipments	Conjunto de equipamentos que fazem parte do sistema (máquinas automáticas, bilheteiras, etc.) (Elemento de sistema)	Sim	Elemento

Elemento de Sistema (Entidade/Equipamento)

Elemento	Descrição	Obrigatório	Tipo de Dados
ElementID	Código que identifica o elemento, único em todo o sistema	Sim	I4
Parent	Código que identifica o elemento “pai” na hierarquia (árvore) de elementos do sistema	Sim	I4
ElementType	Tipo de elemento. Ex.: Estação, MVA	Sim	Texto(50)
Name	Nome do elemento no sistema	Sim	Texto(80)
Properties	Conjunto de propriedades atribuídas ao tipo de elemento em questão e respectivos valores	Sim	Elemento

Propriedade

Elemento	Descrição	Obrigatório	Tipo de Dados
PropertyName	Nome da propriedade	Sim	Texto(50)
Value	Valor da propriedade	Sim	Texto(50)

Fidelização

Elemento	Descrição	Obrigatório	Tipo de Dados
InfoPontos	Informação relativa aos pontos que determinado cliente possui em determinada altura.		Elemento
PointExchange	Trocado de pontos realizada por um cliente		Elemento
PointValues	Informação de configuração dos pontos atribuídos a cada um dos títulos		Elemento

Informação de Pontos

Elemento	Descrição	Obrigatório	Tipo de Dados
Client	Código do cliente	Sim	I4
Points	Número de pontos que o cliente possui	Sim	I4
Date	Data da informação de pontos	Sim	DataHora

Troca de Pontos

Elemento	Descrição	Obrigatório	Tipo de Dados
Client	Código do cliente	Sim	I4
TicketCode	Titulo onde foram utilizados os pontos	Sim	I4
Points	Quantidade de pontos utilizados	Sim	I4
AquisitionDate	Data de aquisição do titulo onde foram utilizados pontos	Sim	DataHora

Valor de Pontos

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de característica de atribuição de pontos.(Carregamento, validação, etc...)	Sim	I4
Points	Pontos a atribuir	Sim	I4

Parametrização de Validadores

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de Lista parâmetros, comandos, ...	Sim	N(1)
Version	Versão da Lista	Sim	N(2)
ParamValid	Lista de parâmetros a enviar para os equipamentos		Elemento
Command	Lista de comandos a enviar para os equipamentos		Elemento

Parâmetros do Validador

Elemento	Descrição	Obrigatório	Tipo de Dados
EquipmentList	Lista de Equipamentos onde devem ser aplicados os parâmetros	Sim	Elemento
Messages	Lista de elementos do validador para os quais se pretende alterar o comportamento		

Lista de Equipamentos

Elemento	Descrição	Obrigatório	Tipo de Dados
Code	Código do equipamento	Sim	I4

Elementos do validador

Elemento	Descrição	Obrigatório	Tipo de Dados
Beeper	Lista de mensagens de parametrização do beeper do validador .	Sim	I4
Leds	Lista de mensagens de parametrização dos leds do validador .		

Mensagem de Configuração do Beeper

Elemento	Descrição	Obrigatório	Tipo de Dados
Status	Estado da mensagem (OK, NOK, BLACK_LIST)	Sim	I4
On	Tempo ou número de ciclos em que o beeper está ON		
Off	Tempo ou número de ciclos em que o beeper está OFF		
Duration	Duração da sinalização		
Volume	Percentagem do volume (0-100%)		

Mensagem de Configuração dos Leds

Elemento	Descrição	Obrigatório	Tipo de Dados
Status	Estado da mensagem (OK, NOK, BLACK_LIST)	Sim	I4
Colors	Coleção de cores a apresentar no led		
On	Tempo em milissegundos ou ciclos que o led deve estar acesso		
Off	Tempo em milissegundos ou ciclos que o led deve estar apagado		
Duration	Duração da sinalização		

Duração

Elemento	Descrição	Obrigatório	Tipo de Dados
ValueType	Unidade de especificação da duração (Milissegundos ou Ciclos)	Sim	I4
Value	Valor da sinalização por beeper ou por led		

Cor a apresentar nos Leds

Elemento	Descrição	Obrigatório	Tipo de Dados
Color	Cor a apresentar no led	Sim	I4

Comandos do Validador

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de Lista parâmetros, Comandos, ...	Sim	Elemento
Version	Versão da lista		
Comand	Comando a enviar		

Comando

Elemento	Descrição	Obrigatório	Tipo de Dados
EquipmentList	Lista de equipamentos a que se destinam os comandos	Sim	Elemento
Cmd	Especificação do comando a enviar para o equipamento		

Especificação do Comando

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de comando a enviar (Service, Restart, etc...)	Sim	Elemento
Status	Estado em que o equipamento deve ficar (on, off)		
TimeTo	Intervalo de tempo até que o equipamento execute o comando		

3.2.4 CORPO DA MENSAGEM DE RECEPÇÃO

Elemento	Descrição	Obrigatório	Tipo de Dados
DateProcessed	Data de processamento da mensagem	Sim	DataHora
Transactions	Colecção de transacções incluídas na mensagem	Não	Elemento
EquipmentStatus	Informação sobre o estado dos equipamentos	Não	Elemento
EventException	Informação sobre os Eventos	Não	Elemento

Transacções

Elemento	Descrição	Obrigatório	Tipo de Dados
ErrorCode	Código do erro da transacção	Sim	N(2)
Transaction	Colecção das transacções incluídas na mensagem e respectiva informação de Erro/Sucesso	Sim	Elemento

Transacção

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo da transacção (Load, Valid)	Sim	N(1)
Version	Versão da estrutura da mensagem	Sim	N(2)
ID	Identificador da transacção	Sim	N(10)
ErrorCode	Código do erro da transacção	Sim	N(2)

Estado do Equipamento

Elemento	Descrição	Obrigatório	Tipo de Dados
ErrorCode	Código do erro dos estados contidos na mensagem	Sim	N(3)
Status	Informação de erro de cada um dos Estado da mensagem	Sim	Elemento

Estado

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	Tipo de estado (0- DepositContent)	Sim	N(1)
Version	Versão da estrutura da mensagem	Sim	N(2)
ID	Identificação do estado	Sim	N(10)
ErrorCode	Código do erro do estado	Sim	N(2)

Evento de Excepção

Elemento	Descrição	Obrigatório	Tipo de Dados
ErrorCode	Código do erro dos eventos enviados na mensagem	Sim	N(3)
Event	Informação de erro de cada um dos Evento da mensagem	Sim	Elemento

Evento

Elemento	Descrição	Obrigatório	Tipo de Dados
Type	0-On, 1-Off, 2-Info	Sim	N(1)
Version	Versão da estrutura da mensagem	Sim	N(2)
ID	Identificação do evento	Sim	N(10)
ErrorCode	Código do erro do evento	Sim	N(2)

3.3 EXEMPLOS XML

Nos capítulos seguintes alguns exemplos de mensagens XML para os elementos Transacção (Venda ou Carregamento e Validação), Estado do Equipamento, Evento e Lista Negra.

3.3.1 TRANSACÇÃO

```

<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Notification</MessageClass>
  <MessageType>Transaction</MessageType>
  <SenderID>33</SenderID>
  <ReceiverID>0</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <Transactions>
      <Transaction Type="Load" Version="1">
        <ID>767673</ID>
        <TimeStamp>2003/12/30 00:01:00</TimeStamp>
        <Equipment>
          <Code>123</Code>
          <SerialNumber>43300322</SerialNumber>
          <ProviderCode>2</ProviderCode>
          <Zones>
            <Zone>L</Zone>
            <Zone>J</Zone>
          </Zones>
          <Local>
            <Code>55</Code>
            <Direction>A;D</Direction>
            <Lines>
              <Line>53</Line>
              <Line>22</Line>
            </Lines>
          </Local>
        </Equipment>
        <Items>
          <Item>
            <Ticket>
              <TicketCode>21</TicketCode>
              <SerialNumber>34442333</SerialNumber>
              <CardType>1;2;3</CardType>
            </Ticket>
          </Item>
        </Items>
      </Transaction>
    </Transactions>
  </MessageBody>
</Message>
  
```

```
<Price>250</Price>
<TariffCode>2</TariffCode>
</Ticket>
<Load>
  <ReloadCount>2</ReloadCount>
  <Profile>5</Profile>
  <ReductionRate>0</ReductionRate>
  <ProviderCode>2</ProviderCode>
  <ContractIdx>1</ContractIdx>
  <ValidDate>
    <StartDate>2004/01/01</StartDate>
    <EndDate>2004/01/31</EndDate>
  </ValidDate>
  <Trips>
    <Load>5</Load>
    <Total>12</Total>
  </Trips>
  <ZoneInfo>
    <NumOfZones>5</NumOfZones>
    <ValidZones>I;J;K</ValidZones>
  </ZoneInfo>
  <ContractCopy>
    <ContractIdx>1</ContractIdx>
    <ContractContent>FA01990E12</ContractContent>
  </ContractCopy>
</Load>
</Item>
</Items>
<Payment>
  <Amount>320</Amount>
  <Discount>
    <Amount>10</Amount>
    <ReasonCode>2</ReasonCode>
    <ReasonTxt>Jovem</ReasonTxt>
    <EntityID>43</EntityID>
    <DocNumber>43434</DocNumber>
  </Discount>
  <PaymentMeans>
    <PaymentMean>
      <Type>Cash;Electronic Payment(Debit)Check;Nota de Crédito</Type>
      <Amount>10</Amount>
      <DocNum>53</DocNum>
      <DateIssue>2003/01/01</DateIssue>
      <EquipIssuer>5552</EquipIssuer>
      <MoreInfo>
        <Info Type="BI;Telef">jkhkjhj</Info>
        <Info Type="BI;Telef">jkhkjhj</Info>
      </MoreInfo>
    </PaymentMean>
  </PaymentMeans>
</Payment>
<IssuedDocs>
  <Receipt>
    <Number>232323</Number>
    <Amount>150</Amount>
  </Receipt>
  <CreditNote>
    <Number>34343</Number>
    <Amount>1000</Amount>
  </CreditNote>
</IssuedDocs>
</Transaction>
<Transaction Type="Valid" Version="1">
  <ID>767673</ID>
  <TimeStamp>2003/12/31 00:01:30</TimeStamp>
  <Equipment>
    <Code>456</Code>
    <SerialNumber>1213212</SerialNumber>
    <ProviderCode>2</ProviderCode>
    <Zones>
      <Zone>L</Zone>
      <Zone>J</Zone>
    </Zones>
    <Local>
      <Code>56</Code>
      <Direction>A;D</Direction>
    </Local>
    <Lines>
      <Line/>
      <Line/>
    </Lines>
  </Equipment>
</Transaction>
```

```

        </Lines>
      </Local>
    </Equipment>
  </Items>
  <Item>
    <Validation>
      <Action>"EN;TR;EX"</Action>
      <TimeStamp>2002/01/30</TimeStamp>
      <ValidResultCode>0</ValidResultCode>
      <ZoneUsed>L</ZoneUsed>
    </Validation>
    <Ticket>
      <SerialNumber>323231323</SerialNumber>
      <CardType>1;2;3</CardType>
      <Profile>5</Profile>
      <ReductionRate>0</ReductionRate>
      <ProviderCode>2</ProviderCode>
      <ValidDate>
        <StartDate>2004/01/01</StartDate>
        <EndDate>2004/01/31</EndDate>
      </ValidDate>
      <Trips>
        <Used>1</Used>
        <Total>7</Total>
      </Trips>
      <ZoneInfo>
        <NumOfZones>5</NumOfZones>
        <ValidZones>A;J;K</ValidZones>
      </ZoneInfo>
      <ContractCopy>
        <ContractIdx>1</ContractIdx>
        <ContractContent>FA01990E12</ContractContent>
      </ContractCopy>
    </Ticket>
  </Item>
</Items>
</Transaction>
</Transactions>
</MessageBody>
</Message>

```

3.3.2 INFORMAÇÃO

```

<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Notification</MessageClass>
  <MessageType>EquipmentStatus</MessageType>
  <SenderID>33</SenderID>
  <ReceiverID>0</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <EquipmentStatus>
      <Status Type="DepositContent" Version="1">
        <ID>767673</ID>
        <TimeStamp>2004/01/28 00:00:00</TimeStamp>
        <Equipment>
          <Code/>
          <SerialNumber/>
          <Local>
            <Code/>
            <Lines>
              <Line>53</Line>
              <Line>22</Line>
            </Lines>
            <Direction>A;D</Direction>
          </Local>
        </Equipment>
        <Module>
          <Type>M</Type>
          <SerialNumber>55</SerialNumber>
          <NominalValue>1</NominalValue>
          <Quantity>10</Quantity>
        </Module>
        <Module>
          <Type>H</Type>
          <SerialNumber>56</SerialNumber>
          <NominalValue>2</NominalValue>

```

```
</Quantity>10</Quantity>
</Module>
</Status>
</EquipmentStatus>
</MessageBody>
</Message>
```

3.3.3 EVENTO

```
<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Notification</MessageClass>
  <MessageType>EventException</MessageType>
  <SenderID>33</SenderID>
  <ReceiverID>0</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <EventException>
      <Event Type="On/Off/Info" Version="1">
        <ID>767673</ID>
        <TimeStamp>2004/01/12 13:45:00</TimeStamp>
        <Equipment>
          <Code/>
          <SerialNumber/>
          <Local>
            <Code/>
            <Lines>
              <Line>53</Line>
              <Line>22</Line>
            </Lines>
            <Direction>A;D</Direction>
          </Local>
        </Equipment>
        <EventType>35</EventType>
        <Infos>
          <Info>Texto genérico</Info>
          <Info>Texto genérico</Info>
        </Infos>
      </Event>
    </EventException>
  </MessageBody>
</Message>
```

3.3.4 LISTA NEGRA

```
<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Notification</MessageClass>
  <MessageType>StatusList</MessageType>
  <SenderID>33</SenderID>
  <ReceiverID>0</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <StatusList>
      <List Type="BlackList" Version="1">
        <ID>767673</ID>
        <StartDate>2003/12/30 00:01:00</StartDate>
        <EndDate>2004/12/30 23:59:59</EndDate>
        <Items>
          <Item ElementType="Card" Type="Element">
            <StartDate>2003/12/30 00:01:00</StartDate>
            <EndDate>2004/12/30 23:59:59</EndDate>
            <InfoText>Text</InfoText>
            <Id>10004500001</Id>
            <ApplyToProvider>
              <Id>0</Id>
            </ApplyToProvider>
          </Item>
          <Item ElementType="Card" Type="Element">
            <StartDate>2003/12/30 00:01:00</StartDate>
            <EndDate>2004/12/30 23:59:59</EndDate>
            <InfoText>Text</InfoText>
            <Id>10003802102</Id>
          </Item>
        </Items>
      </List>
    </StatusList>
  </MessageBody>
</Message>
```

```

        <ApplyToProvider>
          <Id>2</Id>
          <Id>3</Id>
        </ApplyToProvider>
      </Item>
      <Item ElementType="Card" Type="Range">
        <StartDate>2003/12/30 00:01:00</StartDate>
        <EndDate>2004/12/30 23:59:59</EndDate>
        <InfoText>Text</InfoText>
        <Id>10003300000</Id>
        <NoOfElements>1000</NoOfElements>
        <ApplyToProvider>
          <Id>2</Id>
          <Id>3</Id>
        </ApplyToProvider>
      </Item>
      <Item ElementType="Profile" Type="Element">
        <StartDate>2003/12/30 00:01:00</StartDate>
        <EndDate>2004/12/30 23:59:59</EndDate>
        <InfoText>Text</InfoText>
        <Id>2</Id>
        <ApplyToProvider>
          <Id>2</Id>
          <Id>3</Id>
        </ApplyToProvider>
      </Item>
      <Item ElementType="TicketCode" Type="Element">
        <StartDate>2003/12/30 00:01:00</StartDate>
        <EndDate>2004/12/30 23:59:59</EndDate>
        <InfoText>Text</InfoText>
        <Id>23</Id>
        <ApplyToProvider>
          <Id>1</Id>
          <Id>6</Id>
        </ApplyToProvider>
      </Item>
    </Items>
  </List>
</StatusList>
</MessageBody>
</Message>

```

3.3.5 TÍTULOS E TARIFÁRIO

```

<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Advice</MessageClass>
  <MessageType>FareProductsList</MessageType>
  <SenderID>0</SenderID>
  <ReceiverID>33</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <FareProductsList>
      <List Type="TitleInfo" Version="1">
        <Attributes>
          <Attribute>
            <AttributeCode>1</AttributeCode>
            <TextCode>1001</TextCode>
            <AttributeType>A</AttributeType>
            <DataType>N</DataType>
            <Size>4</Size>
            <Sequence>1</Sequence>
            <Keyword>TYPE</Keyword>
          </Attribute>
          <Attribute>
            <AttributeCode>2</AttributeCode>
            <TextCode>1002</TextCode>
            <AttributeType>A</AttributeType>
            <DataType>C</DataType>
            <Size>10</Size>
            <Sequence>2</Sequence>
            <Keyword>ZONES</Keyword>
          </Attribute>
        </Attributes>
        <AttributesDescription>
          <AttributeDescription>

```

```

        <AttributeCode>1</AttributeCode>
        <Value>1</Value>
        <TextCode>1011</TextCode>
    </AttributeDescription>
    <AttributeDescription>
        <AttributeCode>2</AttributeCode>
        <Value>XYZ</Value>
        <TextCode>1021</TextCode>
    </AttributeDescription>
</AttributesDescription>
<FareProducts>
    <FareProduct>
        <TicketCode>1</TicketCode>
        <Description>Bilhete Simples XYZ</Description>
        <Active>$</Active>
        <ServiceCode>11</ServiceCode>
        <LayoutCode>1</LayoutCode>
        <CodifCode>1</CodifCode>
        <TicketAttributes>
            <TYPE>1</TYPE>
            <ZONES>XYZ</ZONES>
        </TicketAttributes>
    </FareProduct>
    ...
</FareProducts>
</List>
<List Type="TextTable" Version="1">
    <TextTypes>
        <TextType>
            <TextTypeID>1</TextTypeID>
            <Description>default</Description>
            <Keyword>DEFAULT</Keyword>
            <Sequence>1</Sequence>
        </TextType>
        <TextType>
            <TextTypeID>2</TextTypeID>
            <Description>Abrev. para VFD de bilheteira</Description>
            <Keyword>MVM_VFD</Keyword>
            <Sequence />
        </TextType>
        <TextType>
            <TextTypeID>3</TextTypeID>
            <Description>Trad. para inglês (máq.)</Description>
            <Keyword>MVA_UK</Keyword>
            <Sequence>2</Sequence>
        </TextType>
    </TextTypes>
    <Texts>
        <Text>
            <TextCode>12</TextCode>
            <TextValues>
                <DEFAULT>2 zonas</DEFAULT>
                <MVM_VFD>Z2</MVM_VFD>
                <MVA_UK>2 zones</MVA_UK>
            </TextValues>
        </Text>
        <Text>
            <TextCode>21</TextCode>
            <TextValues>
                <DEFAULT>1 viagem</DEFAULT>
                <MVM_VFD>1vg</MVM_VFD>
                <MVA_UK>1 trip</MVA_UK>
            </TextValues>
        </Text>
        ...
    </Texts>
</List>
<List Type="PriceList" Version="1">
    <PriceList>
        <TariffCode>2</TariffCode>
        <StartDate>01-04-2004</StartDate>
        <Description>Tarifário simplificado.</Description>
        <Prices>
            <TicketPrice>
                <TicketCode>1</TicketCode>
                <Price>36,35</Price>
                <VAT>5</VAT>
            </TicketPrice>
            ...
        </Prices>
    </PriceList>
</List>

```

```

    </Prices>
  </PriceList>
</List>
<List Type="CombineRules" Version="1">
  <CombineRule>
    <CombineCode>20</CombineCode>
    <Name>COMB. EXEMPL0</Name>
    <Description>Combinação para exemplo</Description>
    <Keyword>TESTE</Keyword>
    <Attributes>
      <CombAttrib>
        <CombAttribID>68</CombAttribID>
        <AttributeCode>1</AttributeCode>
      </CombAttrib>
      <CombAttrib>
        <CombAttribID>69</CombAttribID>
        <AttributeCode>1</AttributeCode>
      </CombAttrib>
      <CombAttrib>
        <CombAttribID>70</CombAttribID>
        <AttributeCode>2</AttributeCode>
      </CombAttrib>
    </Attributes>
    <Values>
      <CombDetail>
        <DetailCode>1</DetailCode>
        <CombAttrib68>1</CombAttrib68>
        <CombAttrib69>7</CombAttrib69>
        <CombAttrib70>XYZZY</CombAttrib70>
      </CombDetail>
      <CombDetail>
        <DetailCode>2</DetailCode>
        <CombAttrib68>2</CombAttrib68>
        <CombAttrib69>2</CombAttrib69>
        <CombAttrib70>XPT0</CombAttrib70>
      </CombDetail>
    </Values>
  </CombineRule>
  ...
</List>
</FareProductsList>
</MessageBody>
</Message>

```

3.3.6 CONFIGURAÇÃO

```

<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Advice</MessageClass>
  <MessageType>ServicesList</MessageType>
  <SenderID>0</SenderID>
  <ReceiverID>33</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <ServicesList>
      <List Type="Menu" Version="1">
        <Menus>
          <Menu>
            <MenuCode>1</MenuCode>
            <Description>Menu Inicial</Description>
          </Menu>
          <Menu>
            <MenuCode>2</MenuCode>
            <Description>Menu Viagens</Description>
          </Menu>
        </Menus>
        <MenusTag>
          <MenuTag>
            <MenuCode>1</MenuCode>
            <TagCode>1</TagCode>
            <TextCode>310</TextCode>
          </MenuTag>
          <MenuTag>
            <MenuCode>2</MenuCode>
            <TagCode>6</TagCode>
          </MenuTag>
        </MenusTag>
      </List>
    </MessageBody>
  </ServicesList>
</Message>

```

```

        <TextCode>303</TextCode>
    </MenuTag>
</MenusTag>
<MenuCommand>
    <MenuCommand>
        <CommandCode>5</CommandCode>
        <MenuCode>1</MenuCode>
        <KeyIn>KEY3</KeyIn>
        <Command>A</Command>
        <Arguments>4,4</Arguments>
        <Sequence>1</Sequence>
    </MenuCommand>
    ...
</MenuCommand>
<MenuCondition>
    <MenuCondition>
        <CommandCode>5</CommandCode>
        <MenuCode>1</MenuCode>
        <TagCode>1</TagCode>
        <Command>SRV</Command>
        <Arguments>6</Arguments>
    </MenuCondition>
    ...
</MenuCondition>
</List>
<List Type="UserList" Version="1">
    <Users>
        <User>
            <UserCode>24</UserCode>
            <UserName>octal</UserName>
            <Password>2e5c5e174cbf0cdab8b05f9771e9b73e</Password>
            <Name>0ctal</Name>
            <NumericId>0</NumericId>
        </User>
        <User>
            <UserCode>25</UserCode>
            <UserName>9999</UserName>
            <Password>de5c5e174cbf0cdab8b05f9771e9b73e</Password>
            <Name>0ctal</Name>
            <NumericId>1</NumericId>
        </User>
    </Users>
    <Authorisations>
        <Authorisation>
            <Domain>
                <EquipmentCode>1016</EquipmentCode>
                ...
            </Domain>
            <Groups>
                <Group>
                    <GroupCode>7</GroupCode>
                    <Users>
                        <User>24</User>
                        <User>25</User>
                    </Users>
                </Group>
                ...
            </Groups>
        </Authorisation>
    </Authorisations>
    <Groups>
        <Group>
            <GroupCode>7</GroupCode>
            <GroupName>MAQ_Manut</GroupName>
            <Description>Operadores de manutenção</Description>
            <Profiles>
                <ProfileCode>6</ProfileCode>
                <ProfileCode>7</ProfileCode>
            </Profiles>
        </Group>
    </Groups>
    <Profiles>
        <Profile>
            <ProfileCode>6</ProfileCode>
            <ProfileName>Consumíveis</ProfileName>
            <Description>Substituição de consumíveis</Description>
            <EquipmentType>MVA</EquipmentType>
            <Functions>
                <FunctionName>EC01</FunctionName>
            </Functions>
        </Profile>
    </Profiles>

```



```

    <FunctionName>EC0101</FunctionName>
  </Functions>
</Profile>
</Profiles>
</List>
<List Type="Topology" Version="1">
  <Entities>
    <Element>
      <ElementID>1</ElementID>
      <Parent>0</Parent>
      <ElementType>Operador</ElementType>
      <Name>Transportes Intermodais do Porto, ACE</Name>
      <Properties>
        <Property>
          <PropertyName>NIF</PropertyName>
          <Value>506240266</Value>
        </Property>
        <Property>
          <PropertyName>Morada</PropertyName>
          <Value>Av. Fernão de Magalhães, 1862 - 9º</Value>
        </Property>
      </Properties>
    </Element>
    ...
  </Entities>
  <Equipments>
    <Element>
      <ElementID>1716</ElementID>
      <Parent>37</Parent>
      <ElementType>MVA</ElementType>
      <Name>MVBCMS6</Name>
      <Properties>
        <Property>
          <PropertyName>Numero</PropertyName>
          <Value>123</Value>
        </Property>
        <Property>
          <PropertyName>PosID</PropertyName>
          <Value>228741</Value>
        </Property>
      </Properties>
    </Element>
    ...
  </Equipments>
</List>
</ServicesList>
</MessageBody>
</Message>

```

3.3.7 PARAMETRIZAÇÃO DE VALIDADORES

```

<?xml version="1.0" encoding="UTF-8"?>
<Message Version="1">
  <MessageClass>Advice</MessageClass>
  <MessageType>ServicesList</MessageType>
  <SenderID>0</SenderID>
  <ReceiverID>33</ReceiverID>
  <MessageID>123456</MessageID>
  <MessageBody>
    <DateIssued>2004/01/05 10:00</DateIssued>
    <ServicesList>
      <List Type="ParamValid" Version="1">
        <ParamValid>
          <EquipmentList>
            <Code>23</Code>
            <Code>57</Code>
          </EquipmentList>
          <Messages>
            <Beeper>
              <Message Status="OK">
                <On>N milisec</On>
                <Off>M milisec</Off>
                <Duration ValueType="CYCLES">4</Duration>
                <Volume>60%</Volume>
              </Message>
              <Message Status="NOK">
                <On>N milisec</On>
                <Off>M milisec</Off>
              </Message>
            </Beeper>
          </Messages>
        </ParamValid>
      </List>
    </ServicesList>
  </MessageBody>
</Message>

```

```

    <Duration ValueType="TIME">X milisec</Duration>
    <Volume>40%</Volume>
  </Message>
  <Message Status="BLACK_LIST">
    <On>N milisec</On>
    <Off>M milisec</Off>
    <Duration ValueType="TIME">X milisec</Duration>
    <Volume>90%</Volume>
  </Message>
</Beeper>
<Leds>
  <Message Status="OK">
    <Colors>
      <Color>Green</Color>
    </Colors>
    <On>N milisec</On>
    <Duration ValueType="CYCLES">4</Duration>
  </Message>
  <Message Status="NOK">
    <Colors>
      <Color>Yellow</Color>
      <Color>Red</Color>
    </Colors>
    <On>N milisec</On>
    <Off>M milisec</Off>
    <Duration ValueType="TIME">X milisec</Duration>
  </Message>
  <Message Status="BLACK_LIST">
    <Colors>
      <Color>Red</Color>
    </Colors>
    <On>N milisec</On>
    <Duration ValueType="TIME">X milisec</Duration>
  </Message>
</Leds>
</Messages>
</ParamValid>
</List>
<List Type="Command" Version="1">
  <Command>
    <EquipmentList>
      <Code>23</Code>
      <Code>57</Code>
    </EquipmentList>
    <Cmd Type="Service">
      <Status>Off</Status>
      <TimeTo>0seconds</TimeTo>
    </Cmd>
  </Command>
</List>
<List Type="Command" Version="1">
  <Command>
    <EquipmentList>
      <Code>1563</Code>
    </EquipmentList>
    <Cmd Type="Service">
      <Status>On</Status>
      <TimeTo>10seconds</TimeTo>
    </Cmd>
  </Command>
</List>
<List Type="Command" Version="1">
  <Command>
    <EquipmentList>
      <Code>100</Code>
    </EquipmentList>
    <Cmd Type="Restart">
      <TimeTo>10seconds</TimeTo>
    </Cmd>
  </Command>
</List>
</ServicesList>
</MessageBody>
</Message>

```