• U C •

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores

Jóni Micael Jordão Cordeiro Neto

Leitor de Cartão do Cidadão com Interface USB

Dissertação de Mestrado Integrado em Engenharia Electrotécnica e de Computadores.

Coimbra

Fevereiro de 2016

· U C ·

Universidade de Coimbra

· U C ·

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores

Leitor de Cartão do Cidadão com Interface USB

por

Jóni Micael Jordão Cordeiro Neto

Dissertação de Mestrado Integrado em Engenharia Electrotécnica e de Computadores, ramo de especialização em Automação.

Orientador: Prof. Doutor Rui Alexandre de Matos Araújo
Co-Orientador: Eng. Jorge Renato Pereira Graça

Júri
Presidente: Prof. Doutor Vitor Manuel Mendes da Silva
Vogais: Prof. Doutor Marco Alexandre Cravo Gomes
Prof. Doutor Rui Alexandre de Matos Araújo

Fevereiro de 2016

Para a família, amigos e todos aqueles que me acompanharam no meu percurso académico.

"Escolha sempre o caminho que pareça o melhor, mesmo que seja o mais difícil; o hábito brevemente o tornará fácil e agradável."
Pitagóras de Samos

Agradecimentos

Rogério Cordeiro e Maria Celeste Cordeiro - Aos meus pais, ficarei eternamente grato por todo o esforço que fizeram em garantir o meu percurso académico;

Ema Neto - Um exemplo de força e motivação que me ajudou nos momentos mais difíceis do meu percurso;

Professor Doutor Rui Araújo - Professor experiente. Agradeço todo o rigor científico transmitido e paciência inesgotável ao longo desta dissertação;

Engenheiro Renato Graça - Agradeço todos os conselhos, opiniões e entusiasmo demonstrado no alcançe dos objetivos deste projeto;

Daniel Carvalho - Pela disponibilidade e auxilio na empresa;

Hugo Paiva - Pela partilha de ideias, conhecimentos e vivências académicas;

Professores - Agradeço a todos os professores que me acompanharam no meu percurso académico pelos ensinamentos transmitidos;

Familiares - Por terem sido fundamentais a formar o homem que sou hoje;

Amigos - Por terem ajudado a tornar o meu percurso agradável e inesquecível;

Universidade de Coimbra - Agradeço a toda a estrutura da universidade pelas condições singulares que proporcionou ao longo do meu percurso académico;

Acronym - Agradeço as condições e meios disponibilizados pela empresa que tornaram este projeto possível.

Resumo

As novas tecnologias permitem facilidade de acesso à informação, melhorar e simplificar o dia-a-dia dos cidadãos. Neste contexto surge o Cartão de Cidadão (CC) que visa simplificar o processo de identificação do cidadão portador. De acordo com o artigo 2.º da Lei n.º 7/2007, de 5 de Fevereiro: *"O Cartão de Cidadão é um documento autêntico que contém os dados de cada cidadão relevantes para a sua identificação e inclui o Número de Identificação Civil, o número de identificação fiscal, o número de utente dos serviços de saúde e o número de identificação da segurança social."*. Assim, a introdução do CC permitiu a redução do número de cartões de identificação do cidadão.

No entanto, as potencialidades deste cartão vão muito mais além. À semelhança de um *Smart Card* (SC), este é dotado por um chip eletrónico capaz de guardar dados pessoais para mecanismos de autenticação e validação. Adicionalmente, este sistema possui no seu circuito integrado uma zona de memória livre onde o titular pode armazenar informações pessoais, tais como: contacto pessoal, profissão e local de trabalho, ou contacto de emergência.

Neste âmbito, em parceria com a empresa *Acronym*, é pretendido o projeto e desenvolvimento de um leitor de CC com capacidade de extrair os seus dados automaticamente e enviar para um Computador Pessoal (PC), através de comunicação *Universal Serial Bus* (USB). Como objetivo secundário é pretendido a leitura e gravação de dados na memória livre do CC. O protótipo será estabelecido com tecnologia AVR8 da *Atmel*, que faça a ponte entre o chip do CC e o PC através de comunicação USB.

Com este projeto pretende-se conhecer o mais exaustivamente possível o conteúdo do CC e a forma de extrair os dados nele armazenados. O resultado desse estudo será posto em prática com a elaboração de um protótipo com a finalidade de resultar em um produto com viabilidade comercial.

Palavras-chave: Cartão de Cidadão, *Smart Card*, Terminal de Leitura, Protocolo T=0, ISO/IEC 7816, Comunicação USB.

Abstract

New technologies allow people to easily access the information, and improve and simplify the citizen life. In this context the Citizen Card (CC) arises to simplify the authentication process of his owner. According to the 2º article of Law nº7/2007, of 5 of February: *"The Citizen Card is an authentic document containing each citizen relevant data for his authentication and includes civil identification number, tax identification number, health identification number and social security number."*. Thus, the introduction of the CC enabled the reduction of the number of identifications cards of the citizens.

However, the potential of this card can go much further. Like a Smart Card (SC), the citizen card is provided by an electronic chip capable of storing personal data for authentication and validation mechanisms. Moreover, this system has in its chip a free memory zone where the owner can store personal data, such as: personal contact, profession and work place, or an emergency contact.

In this context, in partnership with *Acronym* company, it is intended the design and development of a Citizen Card Reader capable of extracting data from a CC automatically and send it to a Personal Computer (PC), through *Universal Serial Bus* (USB) communication. The prototype will be established with an AVR8 technology from *Atmel*, to make the bridge between the electronic chip of the CC and the PC through USB communication.

This project is intended to meet as comprehensively as possible the CC content and the way of extracting the data stored on it. The result of this study will be taken in practice with the development of a prototype in order to result in a product with commercial viability.

Keywords: Citizen Card, Smart Card, Reading Terminal, T=0 Protocol, ISO/IEC 7816, USB Communication.

Abreviaturas e Símbolos

Lista de Acrónimos

AC	*Algarismo de Controlo*
ADF	*Application Dedicated File*
APDU	*Application Protocol Data Unit*
API	*Application Programming Interface*
ATR	*Answer To Reset*
BI	*Bilhete de Identidade*
CC	*Cartão de Cidadão*
COMPACT-TLV	*COMPACT Tag-Lenght-Value*
D	*Fator de Ajustamento Bit-Rácio*
DF	*Dedicated File*
DLL	*Dynamic-Link Library*
EEPROM	*Electrically Erasable Programmable Read-Only Memory*
EF	*Elementary File*
ETU	*Elementary Time Unit*
F	*Fator de Conversão de Rácio de Relógio*
FID	*File ID*
GT	*Guardtime*
HID	*Human Interface Device*
IEC	*International Electrotechnical Commission*
ISO	*International Organzation for Standardization*
LED	*Light-Emitting Diode*
LCS	*Life Cycle State*
MF	*Master File*
MRZ	*Machine Readable Zone*
MUSCLE	*Movement for the Use of Smart Cards in a Linux Environment*
ND	*Número de Documento*
NIC	*Número de Identificação Civil*
NIF	*Número de Identificação Fiscal*

NSS	*Número de Segurança Social*
NUS	*Número de Utente de Saúde*
OCF	*Open Card Framework*
OSI	*Open Systems Interconnection*
PC	*Personal Computer (Computador Pessoal na versão portuguesa)*
PCB	*Printed Circuit Board (Placa de Circuito Impresso na versão portuguesa)*
PC/SC	*Personal Computer/Smart Card)*
PID	*ID de Produto*
PIN	*Personal Identification Number*
PPS	*Protocol Parameter Selection*
PTT	*Postal and Telecommunications Services*
RFID	*Radio-Frequency Identification*
RFU	*Reservado para Uso Futuro*
ROM	*Read-Only memory*
RT	*Rácio de Transmissão*
SC	*Smart Card*
SL	*Slot de Leitura*
SO	*Sistema Operativo*
TL	*Terminal de Leitura*
TLV	*Tag-Lenght-Value*
TPDU	*Transmission Protocol Data Units*
USB	*Universal Serial Bus*
VID	*ID de Vendedor*
WI	*Waiting Time*
WT	*Work Waiting Time*

Símbolos Gerais

D	*Fator de Ajustamento Bit-Rácio*
ETU	*Elementary Time Unit*
F	*Fator de Conversão de Rácio de Relógio*
F_{CLOCK}	*Frequência do Sinal de Relógio Aplicada*
WI	*Waiting Time*
WT	*Work Waiting Time*

Nomenclatura

0xXX	Representação de um Byte em formato Hexadecimal, onde X representa qualquer valor entre '0' e 'F'

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Enquadramento

Em Fevereiro de 2007 iniciou-se a substituição do Bilhete de Identidade (BI) português pelo Cartão de Cidadão (CC). Este novo documento de cidadania surgiu com o principal objetivo de facilitar a identificação e autenticação do seu portador. Para além de substituir o anterior BI, substituiu igualmente o cartão de beneficiário da Segurança Social, o cartão de utente do Serviço Nacional de Saúde e o cartão de Contribuinte.

Contudo, este documento não tem só estas capacidades, ele vai muito mais além. Uma observação mais atenta permite identificar o CC como um verdadeiro computador. Não se trata de um vulgar computador de secretária com monitor, teclado e rato, mas de um computador na sua definição de computação e permuta de dados. Isto porque o CC inclui na sua composição física um chip semicondutor que contem um micro-controlador e memória que lhe permite guardar e processar informação.

Da informação computorizada pelo chip eletrónico deste "computador de bolso" destacam-se as informações de identificação do seu cidadão portador, a sua morada, informações relativas ao cartão (tais como validade e versão), duas chaves criptográficas: uma chave de autenticação e uma chave destinada a assinar documentos digitalmente, e duas impressões digitais do cidadão. Adicionalmente, é possível escrever até 1000 caracteres numa zona de memória livre do cartão, em formato de "bloco de notas" [UCMA *et al.*, 2007].

A computação desta informação por parte do CC apenas faz sentido se a mesma for transmitida para o seu exterior. Nesse sentido é imperioso a utilização de um leitor externo para comunicação com o CC. Este leitor deverá servir de interface entre o nosso computador de bolso e o nosso computador de secretária fazendo recurso à comunicação USB *(Universal Serial Bus)*. No entanto, para que seja possível aceder aos seus dados, é necessário que esta interface obedeça a um conjunto de normas por forma a respeitar a segurança e integridade dos dados contidos no CC.

1.2 Estado da Arte

1.2.1 De Jurgen Dethloff e Helmult Grotrupp ao Cartão Inteligente

O aparecimento de pequenos cartões de identificação capazes de incorporar chips eletrónicos surgiu em 1968 pela mão dos inventores alemães Jurgen Dethloff e Helmult Grotrupp. Este tipo de cartões adotou o nome de Cartão Inteligente, ou Smart Card (SC) como são conhecidos atualmente. Dois anos mais tarde, Kunitaka Arimura desenvolvia um dispositivo semelhante no Japão. No entanto, foi em França no ano de 1974 que se deu o maior progresso neste tipo de tecnologia. Roland Moreno, registara uma patente onde a indústria de semicondutores seria capaz de fabricar e fornecer os circuitos integrados necessários aos cartões de Dethloff e Helmult a um preço razoável. A sua patente foi posta à prova na companhia telefónica francesa PTT (*Postal and Telecommunications Services*) em 1984, e veio massificar a utilização deste tipo de cartões. De imediato, esta nova tecnologia provou todas as expetativas quanto à alta confiabilidade e proteção de dados contidos no cartão contra manipulação [Ferrari *et al.*, 1998].

1.2.2 ISO/IEC 7816

Um SC é apenas um pequeno componente de um sistema mais complexo. Posto isto, tornou-se imperativo que a interface produzida entre o cartão e o resto do sistema fosse precisamente especificada, o que obrigou à adoção de medidas padrão entre fabricantes. Desta forma, os SCs produzidos por fabricantes diferentes podem interagir com diferentes sistemas, evitando a necessidade de diferentes cartões para aplicações distintas.

Assim, o início da jornada em direção à interoperabilidade mundial residiu no estabelecimento de normas concisas sobre o SC e os seus equipamentos de leitura. Neste sentido as organizações ISO (*International Organization for Standardization*) e IEC (*International Electrotechnical Commission*) definiram a norma ISO/IEC 7816. Esta norma segue os padrões anteriormente estabelecidos para os cartões de identificação que incluam bandas magnéticas e/ou gravuras com relevo, nomeadamente as normas 7810, 7811, 7812 e 7813. Por forma a promover o SC com as tecnologias previamente presentes, a compatibilidade entre as normas já existentes foi um pré-requisito na produção da norma ISO/IEC 7816 [Rankl and Effing, 2010].

A norma ISO/IEC 7816 define um conjunto de padrões e procedimentos para cartões com circuitos integrados de contacto (sendo necessário a introdução do cartão num dispositivo de leitura e a troca de informação procede-se pelo contacto dos terminais entre o cartão e o dispositivo) e subdivide-se atualmente 14 partes. No entanto, apenas as quatro primeiras partes desta norma tomaram utilidade nesta dissertação, as quais se enumeram:

- ISO/IEC 7816-1 - Caraterísticas Físicas: define as caraterísticas físicas de um cartão com chip de contacto, assim como testes de certificação necessários [Iso.org, 2011].

- ISO/IEC 7816-2 - Dimensão e Localização dos Contactos: define o tamanho e a posição dos contactos do chip de um SC, assim como a localização do próprio chip eletrónico no cartão [Iso.org, 2013a].

- ISO/IEC 7816-3 - Sinais Elétricos e Protocolos de Transmissão: especifica as caraterísticas elétricas de um SC, tais como a tensão de alimentação, níveis de corrente admissíveis, convenção de paridade, procedimento de operação, mecanismos de transmissão e comunicação com o SC [Iso.org, 2006].

- ISO/IEC 7816-4 - Organização, Segurança e Comandos: elucida o conteúdo do par "comando-resposta" APDU (*Application Protocol Data Unit*) entre a interface de leitura e o cartão, assim como o significado das respostas devolvidas. Define também a estrutura e as caraterísticas de operação de um SC pela sua cadeia de *Answer To Reset* (ATR), métodos de acesso a ficheiros, a organização de ficheiros assim como a estrutura para aplicações e dados contidos no cartão, arquitetura de segurança, métodos para mensagens seguras e canais lógicos [Iso.org, 2013b].

Por forma a satisfazer as necessidades da indústria e acompanhar os novos avanços tecnológicos, a norma ISO/IEC 7816 está em constante reformulação/desenvolvimento, sem nunca perder a interoperabilidade entre as normas já existentes, mantendo os padrões anteriormente estabelecidos.

1.2.3 O Cartão de Cidadão

Respeitando as normas existentes para o SC, nomeadamente a norma ISO 7816, o CC teve a sua fase embrionária em Fevereiro de 2007 na região autónoma dos Açores e veio substituir o antigo documento de identificação português, o BI. Produzido na *Imprensa Nacional-Casa da Moeda*, este novo título de identificação veio aperfeiçoar o modelo de funcionamento dos diversos organismos abrangidos, ao alinhar os processos de modernização a nível organizacional e tecnológico, fomentando a utilização de serviços eletrónicos mediante o recurso de mecanismos de autenticação e assinaturas digitais. De acordo com o artigo 6.º, n.º1, da Lei n.º 7/2007, de 5 de Fevereiro, *"O Cartão de Cidadão é um documento de identificação múltipla que inclui uma zona específica destinada a leitura ótica e incorpora um circuito integrado."*. Assim, o Cartão de Cidadão possui duas vertentes: documento físico e documento digital. Como documento físico, permite ao seu titular *"Provar a sua identidade perante terceiros através da leitura de elementos visíveis, coadjuvada pela leitura ótica de uma zona específica."* (DR 25-Série I, Lei nº7/2007). Como documento digital permite *"Provar a sua identidade perante terceiros através de autenticação eletrónica"* (DR 25-Série I, Lei nº7/2007).

Como documento único de identidade do cidadão, o CC agrega num único cartão os cartões de Contribuinte, Utente do Serviço Nacional de Saúde e Segurança Social. O CC permite que o cidadão se identifique presencialmente nas entidades públicas e privadas e/ou se autentique eletronicamente, interagindo com o chip de contacto disposto no cartão através

Figura 1.1: Informação visível na parte frontal do CC (Fonte: [UCMA *et al.*, 2007]).

Figura 1.2: Informação visível na parte traseira do CC (Fonte: [UCMA *et al.*, 2007]).

de um leitor especifico para o efeito. É através do chip eletrónico que o cidadão se poderá autenticar e assinar de forma simples e segura nos vários canais de interação [UCMA *et al.*, 2007].

Como documento físico de identificação, o CC apresenta a informação ilustrada na Figuras 1.1 e 1.2. Como documento eletrónico, o chip do CC armazena todas as informações visíveis nas Figuras 1.1 e 1.2, além de informações relativas ao cartão. Para além destas informações, o conteúdo eletrónico do CC dispõe da informação da morada do seu titular. Contudo, apenas é possível o seu acesso introduzindo o PIN de Morada fornecido aquando a entrega do cartão, tal como enuncia o artigo 13.º da Lei n.º 7/2007 de 5 de Fevereiro: *"Carece de autorização do titular, a efetivar mediante inserção prévia do código pessoal (PIN), o acesso à informação sobre a morada arquivada no circuito integrado do cartão de cidadão, sem prejuízo do acesso direto das autoridades judiciárias e das entidades policiais para conferência da identidade do cidadão no exercício das competências previstas na lei."*. A aplicação oficial do CC[1] acessa e exibe a informação básica contida no cartão com auxílio de um leitor de cartão suportado.

[1]Disponível em: http://www.cartaodecidadao.pt .

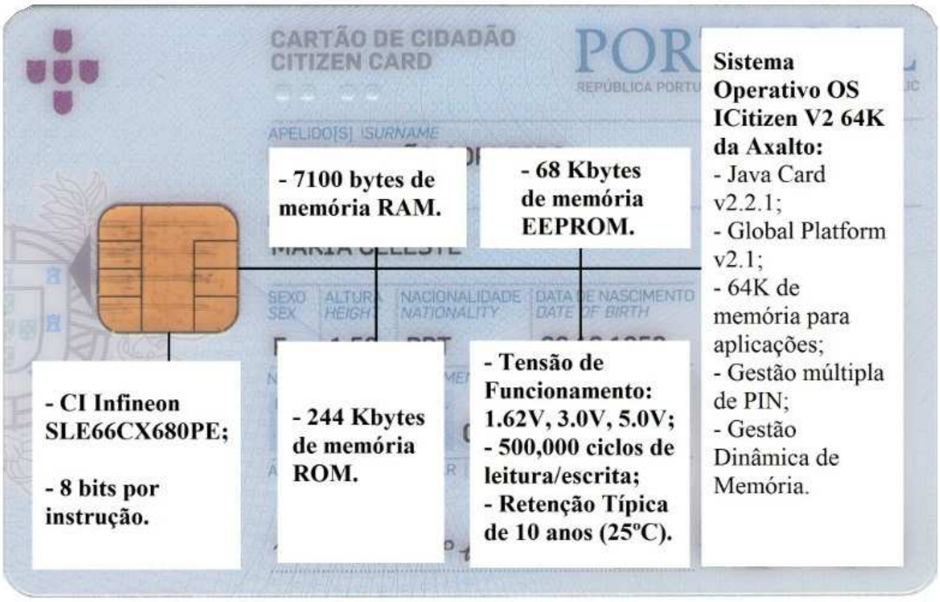

Figura 1.3: Caraterísticas técnicas do CC.

Adicionalmente, é possível escrever numa zona de memória livre do cartão. Esta zona de memória de introdução livre permite ao seu portador inserir até 1000 caracteres em formato de "bloco de notas". No entanto, apesar do acesso a esta zona de memória não se encontrar protegida por mecanismos de segurança, operações como gravar ou apagar dados são limitadas à validação do PIN de Autenticação fornecido aquando a entrega do documento.

O chip eletrónico contém ainda duas chaves criptográficas. Uma chave destinada a assinar documentos digitalmente com valor legal e uma chave de autenticação com a finalidade de validar a identidade do cidadão. O armazenamento eletrónico do CC inclui também duas impressões digitais do seu titular. Cada impressão digital permite aferir de forma inequívoca a correspondência entre o portador e o seu titular, fazendo recurso de um leitor externo de recolha de impressões digitais [UCMA *et al.*, 2007].

O CC é equipado com um circuito integrado da Infineon, nomeadamente o modelo SLE66CX680PE, que executa o Sistema Operativo (SO) OS ICitizen V2 64K do fabricante Axalto [Mesquita, 2010]. A Figura 1.3 contém uma ilustração das principais caraterísticas técnicas do CC. Neste chip, são dispostas aplicações informáticas, baseadas na tecnologia Java Card, que servem de suporte às funcionalidades existentes no cartão, tais como: *IAS* - Aplicação responsável pelas operações de autenticação e assinatura eletrónica; *EMV-CAP* - Aplicação responsável pela geração de palavras-chave únicas por canais alternativos; e *Mach-on-card* - Aplicação responsável pela verificação biométrica de impressões digitais [UCMA *et al.*, 2007].

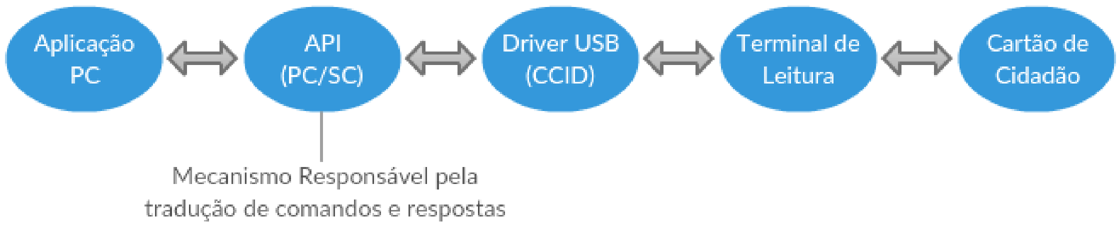

Figura 1.4: Princípio de funcionamento de um TL *Online*.

1.2.4 Terminais de Leitura

O CC por si só não constitui qualquer valor eletrónico adicional face ao anterior título de identificação português. O CC é um dispositivo de comunicação, e como qualquer dispositivo de comunicação não faz sentido que este comunique sozinho. Para que o CC possa afirmar todas as suas vantagens, é necessário a comunicação com um Terminal de Leitura (TL) por forma a extrair e validar os dados existentes no cartão.

Um TL pode ser classificado de acordo com a sua utilização. Neste sentido, um TL pode obter classificação tipo "*Online*" ou tipo "*Offline*" [Rankl and Effing, 2010]. Um TL do tipo *Online* possui uma conexão ininterrupta a um Computador Pessoal (PC) durante a sua operação e o PC assume o controlo das suas funções. Desta forma, o TL apenas se limita a enviar e receber os comandos entre os dois pontos de comunicação, sem efetuar qualquer processamento de dados, servindo de intermediário entre o cartão e um Computador Pessoal. São exemplos de leitores atualmente comercializados os dispositivos Reflex USB v3 da *Axalto*, SCR335 da *SCM*, ou o miniLector Evo da *Bit4id*. Estes dispositivos comerciais promovem um caminho para o conteúdo dos dados alojados no CC, partindo de uma aplicação alojada no PC. No entanto, a aplicação do lado do PC não conhece os padrões de comunicação do CC ou o seu conjunto de instruções. Esse reconhecimento é realizado por uma API (*Application Programming Interface*) que realiza a tradução das instruções enviadas pela aplicação para comandos suportados pelo cartão. Além da troca de comandos e respostas, a API é responsável pela enumeração dos leitores conectados ao PC. As APIs PC/SC (*Windows*), OCF (*Windows/Unix*) e MUSCLE (*Linux*) são exemplos deste tipo de interface.

Sendo a maioria dos dispositivos de leitura portadores de interface USB para comunicação com um PC, existe ainda a necessidade de uma driver USB por forma a estabelecer uma ponte de comunicação entre o TL e a API. Este requisito pode levar à exigência de instalação de *Software* adicional no PC. Porém, os dispositivos mais recentes no mercado já suportam a classe CCID (*Circuit(s) Cards Interface Devices*). Esta classe particulariza uma interface USB, estabelecendo as caraterísticas e protocolos para comunicação com um SC suportado, contornando assim a necessidade de instalação de eventuais drivers USB adicionais.

A Figura 1.4 exemplifica o principio de funcionamento de um TL do tipo *Online*, onde se ilustra o que foi descrito acima. Este tipo de implementação permite comunicar com uma

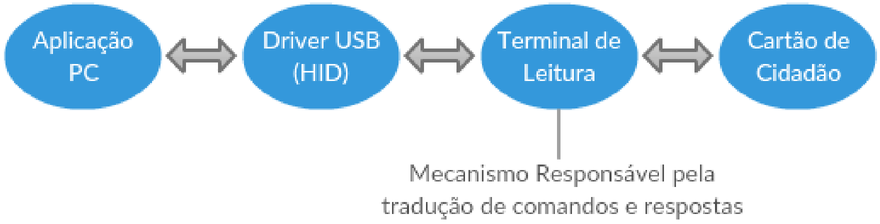

Figura 1.5: Princípio de funcionamento de um TL *Offline*.

vasta gama de SCs (no caso de estudo leia-se cartões de identificação) sem a necessidade de cada fabricante produzir o seu próprio protocolo de comunicação. No entanto, limitam bastante as potencialidades de utilização do CC em projetos não comerciais, nomeadamente em estruturas de controlo de acessos e assiduidade, visto ser necessário a conexão do leitor a um PC que suporte a API e os drivers USB necessários para ser possível estabelecer comunicação.

Em oposição ao TL "*Online*", existe o TL tipo "*Offline*". O terminal *Offline* opera de forma totalmente independente e discreta do sistema inserido. Neste sentido, este tipo de terminais são responsáveis pelo envio de comandos e processamento de respostas enviadas pelo o CC. Por outras palavras, um dado sistema requer um campo de dados ao TL, e este por sua vez processa toda a informação necessária para a obtenção desse campo, não necessitando de nenhum mecanismo de tradução de comandos e respostas, tal como ilustrado na Figura 1.5.

1.2.5 Projetos com o Cartão de Cidadão

Um exemplo de sucesso resultante da incorporação do CC é o desbloqueio das máquinas de venda automática de tabaco. Criada pela empresa portuguesa Movensis em 2012, a incorporação de um TL embebido nas máquinas de venda de tabaco veio facilitar a forma como os proprietários de estabelecimentos de venda limitam o seu acesso a menores de 18 anos [Mota, 2012]. O leitor incorporado nas máquinas de venda de tabaco apenas se destina a ler o campo da data de nascimento do titular do cartão, e com base neste campo e na data atual, determina se o cidadão possui ou não direitos de consumo.

Mais recentemente, fruto do esforço do governo português em informatizar o setor público, em Agosto de 2015 arrancou o projeto titulado "Receita Eletrónica", uma iniciativa das *Farmácias Portuguesas* num projeto patrocinado pela *Mylan*. O objetivo deste projeto é substituir as tradicionais prescrições médicas em papel por prescrições médicas eletrónicas armazenadas nos servidores do Serviço Nacional de Saúde. O papel do CC é crucial neste processo uma vez que a identificação do cidadão, o Número de Utente de Saúde (NUS) e Número de Identificação Fiscal (NIF) contidos no CC serão necessários na dispensa de medicamentos pela entidade farmacêutica.

1.3 Objetivos

Através de uma leitura atenta do estado da arte é possível perceber as imensas potencialidades do CC, porém, foi necessário estabelecer compromissos entre o tempo e os recursos disponíveis para o cumprimento deste projeto. O objetivo desta dissertação passa por conhecer de forma mais exaustiva possível o processo de troca de informação com o CC, por forma fomentar a criação de um TL do tipo *Offline* a integrar em projetos desenvolvidos pela empresa *Acronym*. Atualmente, a *Acronym* emprega cartões RFID (*Radio-Frequency Identification*) nos seus projetos de controlo de acessos e autenticação, e a utilização do cartão de cidadania português abre uma oportunidade para alterar este paradigma. Desta forma é pretendida a conceção de um leitor de baixo custo que opere de forma independente do sistema ao qual está inserido.

O TL e o PC deverão comunicar através de uma interface USB. Por forma a manter o conceito do projeto (independência do leitor do sistema introduzido), pretende-se libertar o TL produzido de eventuais drivers USB adicionais. Desta forma, será implementada a classe USB HID (*Human Interface Device*).

Tendo estes compromissos delineados, como metas básicas espera-se a análise e implementação de mecanismos de leitura com o cartão, nomeadamente a extração dos seguintes dados:

- Número de Série do Cartão;
- Número de Identificação Civil;
- Nome e Apelido do Cidadão.

Como resultados suplementares espera-se a aquisição de uma das impressões digitais armazenadas no CC e a análise e implementação de mecanismos de escrita com o cartão, em particular, escrever na memória livre.

O maior desafio desta dissertação será combater a falta de documentação existente sobre a comunicação entre o CC e um TL ao nível do par "comando-resposta" (comunicação de baixo nível). Será igualmente desafiante descobrir a estrutura de armazenamento do CC, por forma a ser possível aceder aos ficheiros de interesse no âmbito desta dissertação, face à carência de documentação oficial.

1.4 Estratégia e Planeamento

Para superar os objetivos inerentes a esta dissertação foi fundamental conhecer os mecanismos do protocolo de comunicação utilizado no CC, assim como a organização e estrutura de dados armazenados na memória do cartão. Este processo seria seriamente facilitado pela presença de documentação oficial que elucidasse estes pressupostos, no entanto, tal não se verificou.

Posto este cenário, a estratégia de superação das dificuldades encontradas passou por confrontar o delineado na norma ISO/IEC 7816, que define um conjunto de princípios e procedimentos para SCs. O resultado deste estudo permitiu retirar conclusões quanto à transmissão de dados com o cartão, mecanismos de ativação e desativação, tradução do significado da cadeia de caracteres emitida pelo cartão após ativação (ATR), negociação dos parâmetros de comunicação, e troca de comandos e respostas.

Para reconhecimento da estrutura de dados armazenada no cartão e seleção dos dados de interesse, a solução adotada foi executar um processo de captura de dados ao CC (*Sniffing*) durante um processamento de leitura de dados requerido pela aplicação oficial, quando conectado um TL do tipo *Online*. O resultado deste processo permitiu retirar conclusões essenciais, não só quanto à estrutura de dados do cartão mas também, quanto às instruções de comandos a enviar ao CC para obter os dados objetivo. O processo de *Sniffing* foi realizado com recurso ao analisador lógico AX da USBee©[CWAV, 2005] e ao TL comercial SCR335 da SCM Microsystems©[Microsystems, 2011].

1.5 Contribuições e Implementação

A principal contribuição desta dissertação consiste no estudo e divulgação de mecanismos de comunicação ao nível do par comando-resposta com o CC, para implementação de um TL capaz de operar de forma independente ao sistema inserido. Para tal, foi confrontado o disposto na norma ISO/IEC 7816 que define um conjunto de padrões e procedimentos para cartões com circuito integrado, como é o caso do CC. Não menos importante, é exposta a organização dos dados de interesse do CC mediante os objetivos propostos.

A implementação do TL do tipo *Offline* consistiu na estrutura de *Hardware* descrita na Subsecção 1.5.1, que suporta a estrutura de *Software* exposta na Subsecção 1.5.2, nos quais se descreve sucintamente a sua função e ferramentas utilizadas para a sua implementação.

1.5.1 Diagrama de Hardware

A Figura 1.6 ilustra o Diagrama de *Hardware* de alto-nível desenvolvido no âmbito desta dissertação, no qual se identifica:

- **Microcontrolador ATMEGA16U2-AU:**

  O microcontrolador utilizado será responsável pela ativação e estabelecimento de comunicação com o microcontrolador embutido no CC, para além de ser responsável pelo estabelecimento da comunicação USB com o PC. O TL projetado é servido pelo microcontrolador de 8 bits ATMEGA16U2-AU da *Atmel* [ATMEL, 2010], cujos módulos são programados em linguagem *C#*.

- **Slot de Leitura (SL):** Mecanismo responsável pela introdução e contato com o CC. A escolha da SL para a realização deste projeto teve em consideração o disposto na norma

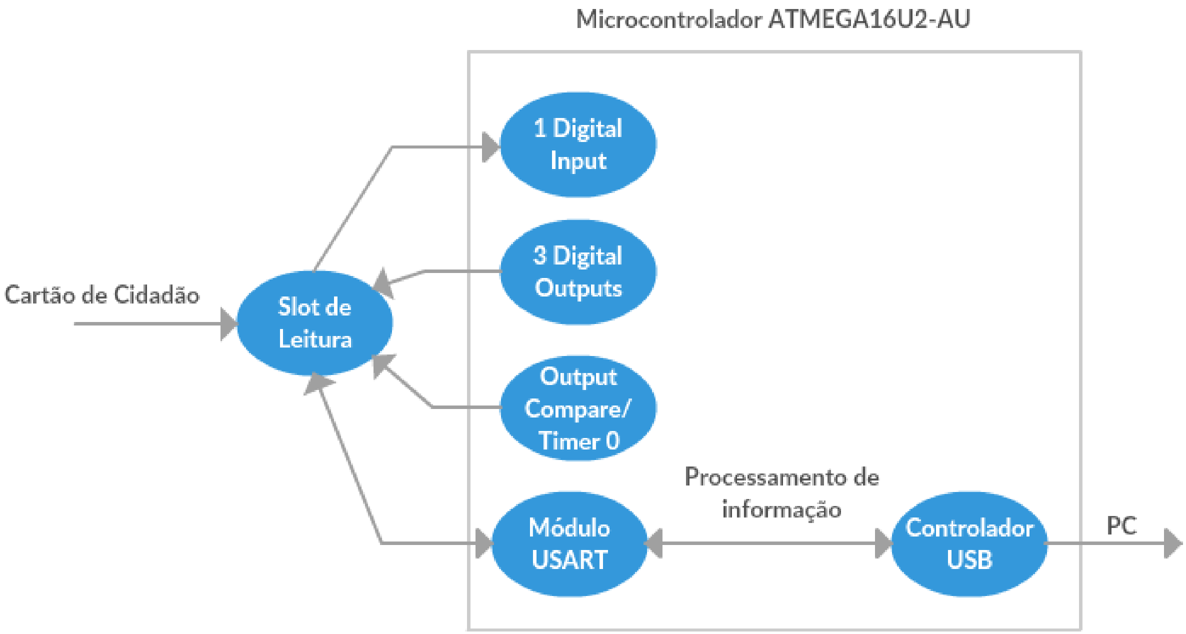

Figura 1.6: Diagrama de Hardware de alto-nível.

ISO/IEC 7816 parte 2, relativa à dimensão e localização dos contatos do chip do CC [Iso.org, 2013a]. Adicionalmente, foi objeto de apreciação a presença de mecanismos que possibilitem a deteção da introdução de um cartão. Assim, a SL utilizada neste projeto foi o modelo 7431E0225S01 da FCI©[FCI, 2005].

- **Digital Input:** A deteção da introdução de um cartão é realizada através de um *switch* mecânico incorporado na SL eleita. Assim, é necessário ativar uma entrada lógica do microcontrolador selecionado, por forma a analisar o estado deste *switch* (aberto ou fechado) para promover o desencadeamento de ações programadas no microcontrolador.

- **Digital Output:** A ativação elétrica dos pinos do microcontrolador do CC é efetivada pela ativação de 3 portos digitais no microcontrolador selecionado para o TL desenvolvido, sendo 2 portos necessários para a alimentação elétrica do cartão (VCC) e 1 porto para a ativação da linha de reset do cartão (RST).

- **Output Compare/Timer:** O módulo *Output Compare* do microcontrolador *Atmega* irá comparar continuamente o valor dos seus registos com o valor de registo do temporizador *Timer 0*. Sempre que esta comparação coincidir, o nível lógico da saída do módulo *Output Compare* será comutada. Aproveitando esta relação, o valor do registo do temporizador *Timer 0* deverá ser programado por forma a assegurar que a saída logica do módulo *Output Compare* comute a uma frequência de sinal de relógio desejada, aplicada ao pino de sinal de relógio do microcontrolador do CC (CLK).

- **Módulo USART:** Módulo de comunicação série do microcontrolador selecionado responsável pela transmissão e receção de dados com a linha de dados do CC (I/O). As linhas de RX e TX deste módulo são conectadas entre si por forma a assegurar uma comunicação *half-duplex* com o CC.

- **Controlador USB:** O microcontrolador da *Atmel* selecionado para este projeto é dotado de um Controlador USB. Este módulo permite estabelecer uma comunicação série com um PC, habilitando a permuta de dados entre os dois dispositivos.

1.5.2 Diagrama de Software

A Figura 1.7 ilustra o Diagrama de *Software* de alto-nível desenvolvido no âmbito desta dissertação, no qual se identifica:

- **Setup Hardware:** Módulo responsável pela iniciação de mecanismos do microcontrolador empregue no TL produzido. Neste módulo são inicializados: portos lógicos digitais, *Timer 0* e módulo *Ouput Compare*, módulo USART e o Controlador USB.

- **Firmware de Comunicação com o CC:** Este módulo engloba um conjunto de funções que permite operar com o CC, através do envio e receção de dados do *buffer* do módulo USART do microcontrolador utilizado. Estas funções incorporam o conjunto de funcionalidades estudadas definidas na norma ISO/IEC 7816, nomeadamente a parte 3 [Iso.org, 2006] e parte 4 [Iso.org, 2013b], cuja análise se encontra descrita nos Capítulos 2 e 3. Dos mecanismos implementados por este módulo destacam-se: mecanismos de ativação e desativação do CC, estabelecimento da comunicação, estrutura de troca de comandos e respostas, extração e escrita de dados, deteção de erros, e transposição de mecanismos de segurança.

- **Protocolo de Mensagens USB:** Por forma a estabelecer comunicação entre uma aplicação instalada no PC e o TL desenvolvido, foi criado um protocolo de comunicação baseado na troca de mensagens. O módulo estabelecido, define um conjunto de mensagens de 8 bytes de dados que governam pedidos e informações do estado do processo, nomeadamente: solicitação de comandos e respostas, desencadeamento de ações, informação de erros no processo e sincronização da comunicação. Este módulo encontra-se enunciado na Subsecção 4.2 e descrito no Anexo A.6.

- **LUFA:** O *firmware* de comunicação USB foi aplicado com recurso ao uso da biblioteca de funções LUFA [Camera, 2013], que permitiu a implementação da classe USB-HID para o TL criado. A biblioteca LUFA é aplicável em microcontroladores da ATMEL habilitados com Controlador USB e a sua integração permitiu auxiliar o TL criado no processo de reconhecimento, enumeração e troca de relatórios de dados USB com um PC.

- **Descritores USB:** Módulo responsável pela identificação e processo de enumeração do TL por um PC. Neste módulo encontram-se definidos os descritores para a classe USB-

HID, fundamentais ao processo de enumeração do dispositivo, enunciados na Subsecção 4.2 e descritos no Anexo A.5.

- **Gestor de Dispositivos USB:** Mecanismo de gerenciamento de dispositivos USB em Windows©.
- **PyUSB:** Módulo de funções responsável pela identificação e estabelecimento de comunicação entre as DLLs (*Dynamic-Link Library*) em Windows, responsáveis pelo reconhecimento e acesso de dispositivos USB, e o módulo da aplicação produzida. Este módulo foi aplicado com recurso ao uso da biblioteca de funções PyUSB[2], concebida para utilização em aplicações escritas em linguagem *Python*™.
- **Acronym - Leitor CC:** Aplicação responsável pela visualização dos dados adquiridos e interação com o utilizador. A aplicação foi escrita em linguagem Python e a sua interface foi realizada com recurso à ferramenta *Qt Designer*©versão 5.4.2 [Qt, 2015], tendo sido criados ficheiros executáveis em ambiente Windows©por forma a facilitar a sua distribuição.

1.6 Trabalho Realizado

Os resultados alcançados nesta dissertação vão muito mais além dos objetivos inicialmente propostos. Para além serem atingidos os objetivos delineados (com exceção da extração de impressões digitais), foram ainda lidos outros dados contidos no chip do CC, tendo sido estudados os mecanismos de leitura e validação envolvidos. Para tal, foram implementados alguns dos comandos de baixo nível definidos na norma ISO 7816 parte 4 [Iso.org, 2013b].

Com o desenvolvimento deste projeto foram extraídos da memória do CC os seguintes dados relativos ao cidadão:

- Nome(s) e Apelido(s) do cidadão;
- Número de Identificação Civil (NIC) e Número de Documento (ND);
- Nacionalidade;
- Filiação e Data de Nascimento;
- Altura e Sexo;
- Número de Identificação Fiscal (NIF);
- Número de Segurança Social (NSS);
- Número de Utente de Saúde (NUS);

[2]Disponível em https://walac.github.io/pyusb/

- Zona de Memória Livre do CC;
- Fotografia do Cidadão.

Por forma a garantir a veracidade da informação extraída foram implementados algoritmos de verificação da autenticidade dos identificadores do ND, NIF e NSS seguindo o método exposto por [Teixeira, 2015].

Adicionalmente, foram extraídos dados relativos ao cartão em si, tais como:

- Entidade Emissora;
- Local de Pedido;
- Data de Emissão e Data de Validade;
- Machine Readable Zone (MRZ);
- Número de Série;
- Versão.

Satisfazendo um dos objetivos suplementares, foram estudados e implementados os mecanismos de escrita no cartão por forma a apagar e escrever na memória livre do CC. O resultado deste estudo permitiu a gravação até 1000 caracteres de texto em formato "bloco de notas". Para tal, foram estudados mecanismos de validação do PIN de Autenticação, visto ser um requisito essencial para alcançar este objetivo.

Adicionalmente, foram também estudados mecanismos de validação do PIN de Morada com a finalidade de obter a morada do cidadão armazenada no cartão. Após superados os mecanismos de validação de acesso da morada, são extraídos com sucesso os seguintes campos de dados:

- Tipo de Via, Designação da Via e Número de Porta;
- Andar e Lado (se existente);
- Código-Postal e Localidade Postal;
- Localidade, Freguesia, Conselho e Distrito.

A fase experimental foi concebida tendo por base um *Arduino™UNO Rev3*[3], com recurso ao micro-controlador ATMEGA16U2 da *Atmel* [ATMEL, 2010]. Este micro-controlador de baixo custo, possui um módulo de transferência de dados USB 2.0 que auxilia a transferência dos dados lidos do cartão para um PC. Para a transferência de dados via USB, foi implementada a classe USB-HID com recurso à *framework* LUFA[4] [Camera, 2013].

[3]https://www.arduino.cc/en/Main/ArduinoBoardUno
[4]Disponível em http://www.lufa-lib.org

Como prova de conceito, foi produzido um *Software* escrito em linguagem *Python*™que devolve os resultados produzidos nesta dissertação, para visualização dos dados armazenados no CC por parte do utilizador. Foi criado um protocolo de comunicação entre o TL e o programa produzido, por forma a tornar possível o envio e receção de dados entre ambas as partes. Assim, o programa produzido envia e recebe os dados da interface USB e devolve os resultados da comunicação com o cartão.

1.7 Organização da Dissertação

Esta dissertação é estruturada em cinco capítulos, organizados da seguinte forma:

- No Capítulo 1, é feita uma introdução às potencialidades do CC, assim como os objetivos e metodologias seguidas na dissertação;
- No Capítulo 2, é abordada a preparação e transmissão de dados entre o chip de contacto do CC e um TL;
- No Capítulo 3, é revelada a estrutura do SO do CC, nomeadamente a sua organização de ficheiros, protocolos de transmissão, estrutura de mensagens e comandos básicos de comunicação;
- No Capítulo 4, são expostos os mecanismos implementados por forma a atingir os objetivos propostos nesta dissertação;
- No Capítulo 5, são apresentadas as conclusões acerca do trabalho desenvolvido e objetivos não concluídos, assim como possíveis trabalhos futuros.

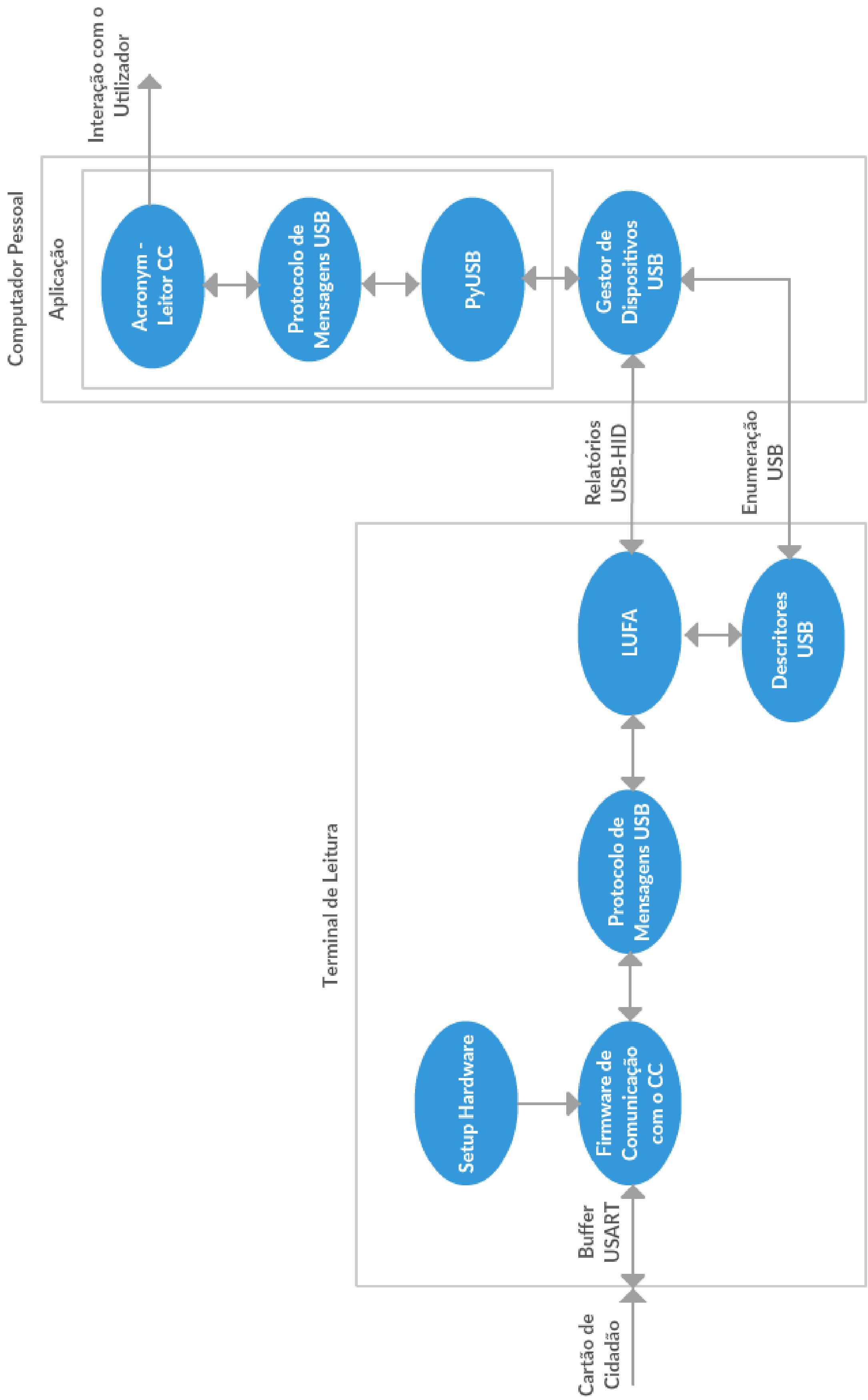

Figura 1.7: Diagrama de Software de alto-nível.

Capítulo 2

Transmissão de Dados

Para efeitos de utilização viável do CC na slot da interface de leitura, os seus contactos elétricos deverão possuir um contacto mecânico consistente com os contactos elétricos do TL. Ao inserir o CC no TL, cinco contactos elétricos são ativados de forma sequencial, por forma a satisfazer um mecanismo chamado de *"Cold Reset"*. Satisfeito este mecanismo, o micro-controlador contido no CC inicia a execução do seu *Software* e envia ao TL uma cadeia de caracteres denominada por ATR (*Answer To Reset*) [Rankl, 2007]. Esta sequência de caracteres contem vários parâmetros e informações relativos ao conteúdo do CC, assim como parâmetros de transmissão de dados, que permitem alterar, entre outros, o Rácio de Transmissão (RT) numa operação denominada por PPS (*Protocol Parameter Selection*).

Após estes processos, o TL e o CC encontram-se preparados para comunicar entre si, num processo que envolve a troca de comandos e respostas (ver Figura 2.1).

2.1 Conexão com o CC

O componente de maior importância na discussão desta dissertação é sem dúvida o micro-controlador embebido no CC. É neste chip que se encontram os dados de interesse para os objetivos deste projeto. O chip do CC possui 8 contactos que são responsáveis pelo fornecimento de energia ao cartão e permitem a comunicação de dados com o exterior. Deste modo, são necessárias ligações elétricas entre os contactos do chip do CC e um dispositivo de leitura externo. As ligações elétricas do CC seguem a norma ISO/IEC 7816 parte 2 [Iso.org, 2013a]. Seguindo o disposto nesta norma é possível identificar os contactos do chip do CC exibidos na Figura 2.2, e descritos na Tabela 2.1.

Tabela 2.1: Designação dos contactos e suas respetivas funções. Fonte: [Iso.org, 2013a].

Contacto	**Designação**	**Função**
C1	VCC	Tensão de alimentação
C2	RST	Linha de reset
C3	CLK	Linha de sinal de relógio

C4	RFU	Reservado para uso futuro (não conectado)
C5	GND	Ligação à terra (ground)
C6	VPP	Tensão de programação (não conectado)
C7	I/O	Canal de comunicação série
C8	RFU	Reservado para uso futuro (não conectado)

- Tensão de Alimentação (VCC)

  O CC obtém a sua tensão de alimentação, e correspondente corrente de alimentação, através do contacto C1 (Figura 2.2). Segundo a norma ISO/IEC 7816 parte 3 [Iso.org, 2006], a tensão de alimentação do chip eletrónico do CC é de 5.0 volts (em referência com o terminal terra) com uma tolerância máxima de 10%, o que segue o mesmo padrão de um SC comercial classe A.

  A corrente de alimentação está diretamente relacionada com a tensão de alimentação e a frequência do sinal de relógio fornecida. Em [Iso.org, 2006] é especificada uma corrente de alimentação máxima de 60 mA para cartões pertencentes à classe de alimentação A (5.0 V) em operação à frequência máxima aplicável.

- Linha de Reset (RST)

  Para ativar/desativar o micro-controlador do CC por forma a iniciar/finalizar a comunicação com um TL, é necessário fornecer ao pino C2 um sinal de reset (Figura 2.2). O CC efetua as operações de ativação e desativação do micro-controlador durante o estado ativo-baixo da linha de reset. Para ativar o micro-controlador é requerida uma transição do tipo ativo-baixo para ativo-alto [Iso.org, 2006].

- Linha de Relógio (CLK)

  O micro-controlador presente no chip do CC não possui um gerador de sinal de relógio interno. Desta forma torna-se imperativo fornecer um sinal de relógio externo através do pino C3 (Figura 2.2). Este sinal de relógio servirá de referência para o RT entre o cartão e o terminal.

  Conforme especificado em [Iso.org, 2006], a ativação do cartão deve de ser realizada com uma de frequência de sinal de relógio entre 1 MHz e 5 MHz, com um ciclo de trabalho (*duty cycle*) de 50% com uma tolerância de ±20%.

  No âmbito deste projeto foi fornecido um sinal de relógio de 4 MHz ao micro-controlador do CC, proveniente do TL.

- Canal de Comunicação (I/O)

  O canal de comunicação existente no CC (pino C7, Figura 2.2), é usado como entrada de dados (modo de receção) e saída de dados (modo de transmissão). Dado que apenas existe um canal de comunicação, a troca de informação entre o TL e o CC deverá ocorrer

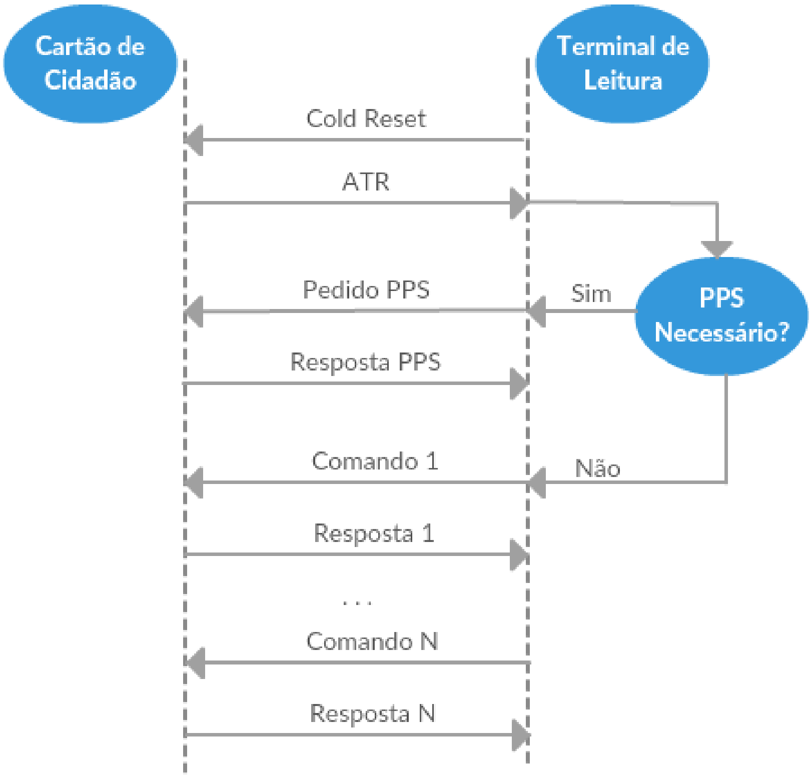

Figura 2.1: Estabelecimento da comunicação entre o TL e o CC após inserção do cartão.

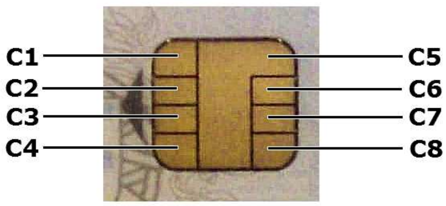

Figura 2.2: Contactos Elétricos do chip do CC.

em modo "*half-duplex*" [Rankl, 2007]. Como consequência, o CC e o TL não poderão enviar e receber dados simultaneamente. Isto implica que a comunicação se proceda por uma sequência de comando-resposta em que um dos lados da comunicação envia um comando enquanto o outro lado da comunicação se encontra à escuta do comando enviado.

A comunicação com o cartão é iniciada sempre pela interface de leitura. O CC limita-se a responder aos comandos enviados pelo TL, o que significa que o CC em circunstância alguma irá enviar uma resposta sem que para a qual não tenha sido efectuado um pedido pelo TL. Esta descrição enquadra-se perfeitamente numa comunicação mestre-escravo, onde o TL é o mestre e o CC é o escravo.

Por forma a manter a transmissão de dados responsiva, deverá ser garantido que o TL e o módulo de comunicação do CC não se encontram ambos em estado de transmissão ou de receção de dados [Rankl, 2007].

- Ligação à terra (GND)

  O terminal de ligação à terra (pino C5, Figura 2.2) será o terminal de referência às tensões aplicadas no semicondutor do CC, e por isso é importante assegurar a sua correta ligação.

- Tensão de Programação (VPP)

  Os SCs que surgiram nos anos 80 necessitavam da aplicação de uma tensão externa para programar e apagar a memória EEPROM (*Electrically Erasable Programmable Read-Only Memory*) contida no chip [Everett, 1992]. Contudo, a partir dos anos 90, foi uma prática comum entre fabricantes gerar esta tensão diretamente do chip semicondutor usando circuitos bombeadores de carga [Jurgensen and Guthery, 2002], tornando o uso do contacto C6 desnecessário (Figura 2.2). Apesar de desconectado internamente, o chip do CC ainda apresenta este contacto em comprimento do que é especificado na norma ISO 7816 parte 2 [Iso.org, 2013a].

- Reservado para Uso Futuro (RFU)

  Em comprimento com a norma ISO 7816 parte 2 [Iso.org, 2013a], estes contactos são presentes na interface de ligação do CC. No entanto, os contactos C4 e C8 (Figura 2.2) são reservados para futuras utilizações e encontram-se desligados do micro-controlador do CC. Outro motivo da sua existência é oferecer estabilidade ao alojamento do chip semicondutor na estrutura física do cartão.

2.2 Sequência de Ativação e Desativação

No momento de introdução do CC, os contactos elétricos da interface de comunicação encontram-se desativados. O chip embebido no cartão poderia ser seriamente danificado se ao introduzir o cartão o TL já possuísse os seus contactos elétricos ativos. Desta forma, após a deteção da correta introdução do CC na slot, é efetuado um procedimento denominado de *"Cold Reset"* [Iso.org, 2006]. Este procedimento consiste na ativação elétrica dos contactos do TL de modo sequencial, por forma a inicializar a memória ROM (*Read-Only Memory*) do micro-controlador presente no cartão.

O procedimento de *Cold Reset* consiste nos seguintes passos:

1. Colocação da linha RST a nível lógico baixo;

2. Ativação da linha de VCC;

3. Fornecer o sinal de relógio ao micro-controlador do CC;

4. Manter a linha de RST a nível lógico baixo durante 400 ciclos de relógio;

5. Colocar a linha de RST a nível lógico alto.

De modo inverso, o processo de desativação consiste na desativação elétrica dos contactos do TL por forma a diminuir a probabilidade de dano do chip do cartão no momento de remoção. O procedimento de desativação do cartão consiste na seguinte sequência [Iso.org, 2006]:

1. Colocação da linha RST a nível lógico baixo;
2. Desabilitar o sinal de relógio fornecido ao micro-controlador do CC;
3. Colocação da linha I/O a nível lógico baixo;
4. Desativação da linha de VCC.

Existe ainda outro modo de reset ao micro-controlador do cartão, conhecido como *"Warm Reset"* [Iso.org, 2006]. Neste procedimento de reset apenas é aplicado um sinal de reset ao contacto C2 do cartão com a interface elétrica ativa.

2.3 Transmissão de Caracteres

Como o sinal de relógio do micro-controlador do CC é fornecido pelo TL, os temporizadores do micro-controlador do CC irão ser influenciados por este sinal de relógio externo. Desta forma, o tempo de cada bit de dados transmitido irá estar intimamente ligado ao valor da frequência do sinal de relógio aplicado. O tempo de transmissão de um único bit é chamado de *Elementary Time Unit* (ETU) e é determinado pela expressão 2.1 [Iso.org, 2006]:

$$ETU = \frac{F}{D \times \mathbf{F}_{CLOCK}}, \tag{2.1}$$

onde F denota o Fator de Conversão de Rácio de Relógio aplicado, isto é, o número de ciclos de relógio necessários para reconhecer e transmitir cada bit para o buffer de entrada/saída do micro-controlador (por defeito $F = 372$). Por sua vez, F_{CLOCK} denota a Frequência de Relógio aplicada e D denota o Fator de Ajustamento Bit-Rácio (por defeito $D = 1$).

O ETU inicial está inversamente relacionado com o RT. O número de bits transmitidos por segundo é definido pela expressão 2.2:

$$RT = \frac{1}{ETU}. \tag{2.2}$$

O valor de RT não tem que ser imperiosamente dado pelo valor da expressão (2.2). Por razões técnicas, a norma ISO/IEC 7816 parte 3 [Iso.org, 2006] permite uma tolerância de ± 0.2 ETU por cada byte de dados transmitido.

A transmissão de dados entre o TL e o CC é uma comunicação série assíncrona. Cada conjunto de 8 bits forma um byte, que é enviado numa sequência de bytes em série [Iso.org, 2006]. A Figura 2.3 ilustra a moldura de transmissão de um caractere.

Na Figura 2.3 identificam-se:

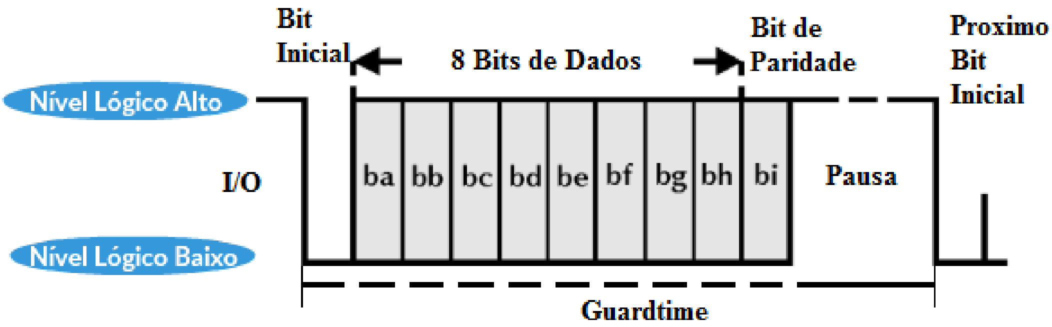

Figura 2.3: Ilustração da transmissão de um caractere.

- Bit Inicial

  É definido como uma transição de nível lógico alto para nível lógico baixo e demarca o início da transmissão, por forma a promover sincronismo entre bits.

- Bits de Dados

  São denominados por 'ba' a 'bh', e representam os 8 bits que definem o caractere transmitido.

- Bit de Paridade

  Representado por 'bi', é definido para alcançar paridade par, isto é, fazer com que seja par o número bits com valor '1' no conjunto dos 8 bits de dados transmitidos e do bit de paridade. Caso não seja atingido um número par, é detetado um erro na transmissão de dados e o caractere será retransmitido.

- Pausa

  É definido por defeito, como nível lógico alto durante dois períodos de bit (para $EGT = 0$). Durante este período o transmissor analisa a linha de dados e se, através da verificação do bit de paridade, se verifique um erro na transmissão do caractere precedente, o recetor coloca a linha de transmissão a nível lógico baixo a meio do primeiro período de bit. Assim, o transmissor deteta esta alteração na linha de dados e retransmite novamente o caractere enviado.

- Guardtime (GT)

  Define o atraso mínimo entre dois caracteres consecutivos, representados pelo Bit Inicial, por forma a assegurar a correta receção do caractere por parte do cartão. Por defeito têm-se $GT = 12ETU$. Caso seja especificado na cadeia de caracteres ATR, é possível requerer um GT superior ao definido por defeito (ver Secção 2.4.3).

  Em oposição, o *Waiting Time* (WT) define o atraso máximo entre dois caracteres consecutivos, e caso este tempo seja excedido o cartão é dado como inativo (sem resposta).

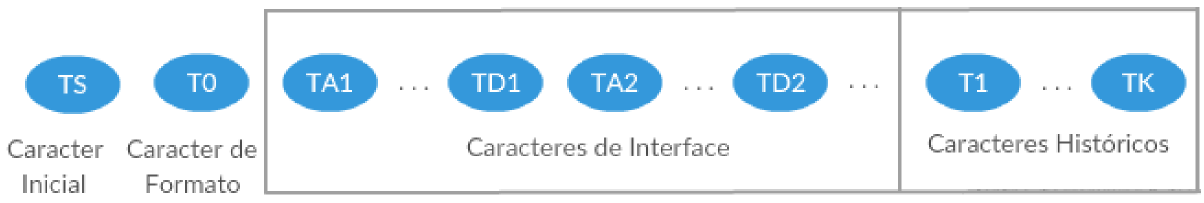

Figura 2.4: Configuração geral da ATR.

- Próximo Bit Inicial

  Marca o início da cadeia de bits seguinte.

2.4 Answer To Reset

Após estar completo o procedimento de *Cold Reset*, o micro-controlador incorporado no CC inicia a sua sequência de iniciação e responde com uma série de caracteres denominada por *"Answer To Reset"* (ATR) [Iso.org, 2006]. Esta cadeia de caracteres é emitida na linha de dados do cartão após 400 a 40,000 ciclos de relógio, depois de completo o procedimento de reset. Caso o procedimento de reset não desencadeie a ATR dentro do prazo estabelecido, o TL deverá desativar eletricamente o cartão. Entre outros, a ATR divulga detalhes dos parâmetros de comunicação entre a interface produzida e o CC, auxiliando a identificação de um SC introduzido no TL.

A cadeia de caracteres devolvida obedecerá à estrutura da Figura 2.4. Esta cadeia de caracteres é sempre transmitida com um RT dado pela expressão (2.2), onde $F = 372$ e $D = 1$. Sendo que foi aplicado uma frequência de sinal de relógio de 4 MHz ao cartão proveniente do TL, resulta em $RT = 10752$ bits/s.

Atualmente são conhecidas três ATRs devolvidas pelo CC na sua iniciação, listadas na Tabela 2.2. O conteúdo da ATR é descrito em detalhe na norma ISO/IEC 7816 parte 3 [Iso.org, 2006] e parte 4 [Iso.org, 2013b] e pode ser facilmente traduzida através do website https://smartcard-atr.appspot.com.

2.4.1 Caractere Inicial (TS)

O Caractere Inicial TS é o primeiro caractere devolvido pela ATR [Iso.org, 2006]. Este caractere é responsável por definir o padrão da comunicação para todos os caracteres subsequentes, nomeadamente a sua lógica binária. Os primeiros três bits do Caracter TS são usados para sincronizar a comunicação, ao passo que os três bits seguintes são usados para indicar a convenção utilizada na comunicação.

O Caractere TS admite duas configurações possíveis: Convenção Direta e Convenção Inversa, ilustradas nas Figuras 2.5 e Figura 2.6, respetivamente. Na Convenção Direta o bit

Tabela 2.2: Diferentes ATRs conhecidas para o CC.
Fonte: https://www.eftlab.com.au/index.php/site-map/knowledge-base/171-atr-list-full.

Tipo	ATR
1	3B 7D 95 00 00 80 31 80 65 B0 83 11 00 C8 83 00 90 00 TS=0x3B T0=0x7D TA1=0x95 TB1=0x00 TC1=0x00 Caracteres Históricos: 80 31 80 65 B0 83 11 00 C8 83 00 90 00
2	3B 7D 95 00 00 80 31 80 65 B0 83 11 C0 A9 83 00 90 00 TS=0x3B T0=0x7D TA1=0x95 TB1=0x00 TC1=0x00 Caracteres Históricos: 80 31 80 65 B0 83 11 C0 A9 83 00 90 00
3	3B 95 95 40 FF D0 00 54 01 32 TS=0x3B T0=0x95 TA1=0x95 TD1=0x40 TC2=0xFF Caracteres Históricos: D0 00 54 01 32

menos significativo é transmitido primeiro e o nível lógico '1' representa o estado ativo-alto. Pelo contrário, na Convenção Inversa o bit mais significativo é transmitido primeiro e o nível lógico '1' representa o estado ativo-baixo. A seleção da lógica apropriada será feita através de um valor de TS de '00111011' para a lógica direta e um valor de TS de '00111111' para a lógica inversa, ou em código hexadecimal 0x3B e 0x3F, respetivamente.

Pela Tabela 2.2, é possível verificar que o CC apenas admite TS=0x3B, logo emprega a Convenção Direta.

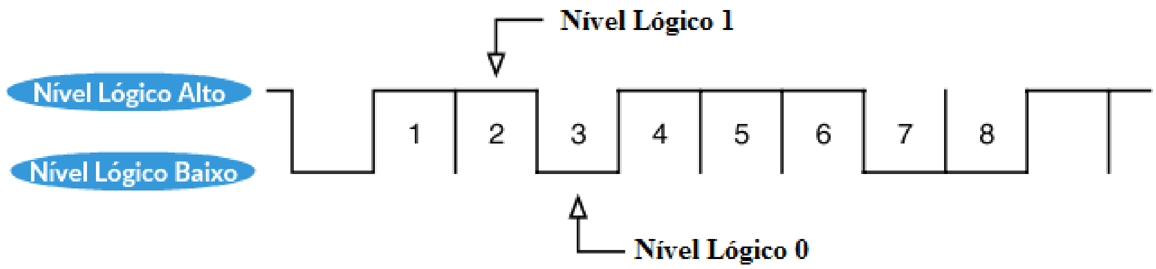

Figura 2.5: Convenção Direta de dados (TS=0x3B) (Fonte: [Rankl and Effing, 2010]).

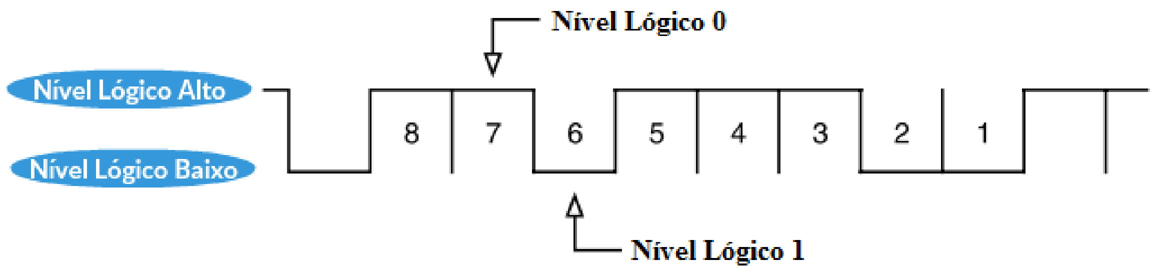

Figura 2.6: Convenção Inversa de dados (TS=0x3F) (Fonte: [Rankl and Effing, 2010]).

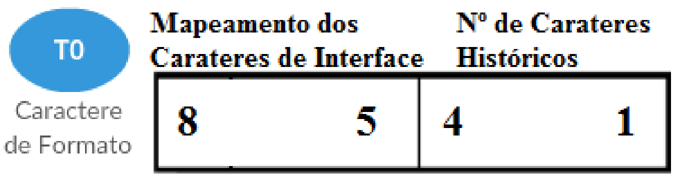

Figura 2.7: Configuração do Caractere de Formato T0.

2.4.2 Caractere de Formato (T0)

O Caractere de Formato T0 (Tabela 2.2) devolve a informação necessária para compreender a restante sequência proveniente da ATR. Este caractere contém informação sobre a presença dos Caracteres de Interface e do número de Caracteres Históricos [Iso.org, 2006].

Os 4 bits mais significativos do Caractere T0 indicam a presença dos Caracteres de Interface TD1, TC1, TB1 e TA1, respetivamente. Ao passo que, os 4 bits menos significativos devolvem o número de Caracteres Históricos em formato binário (de 0 a 15), tal como ilustrado na Figura 2.7.

Nas respostas devolvidas pelo CC, as respostas do tipo 1 e do tipo 2 possuem T0=0x7D. Interpretando este caractere T0 é possível reter que são transmitidos 3 caracteres de interface (TA1=0x95, TB1=0x00 e TC1=0x00) e 13 caracteres históricos (0x80, 0x31, 0x80, 0x65, 0xB0, 0x83, 0x11, 0x00 (ou oxC0), 0xC8 (ou 0xA9), 0x83, 0x00, 0x90 e 0x00). Na resposta tipo 3 obtém-se T0=0x95, o que significa que são transmitidos 3 Caracteres de Interface (TA1=0x95, TD1=0x40 e TC2=0xFF) e 5 Caracteres Históricos (0xD0, 0x00, 0x54, 0x01 e 0x32).

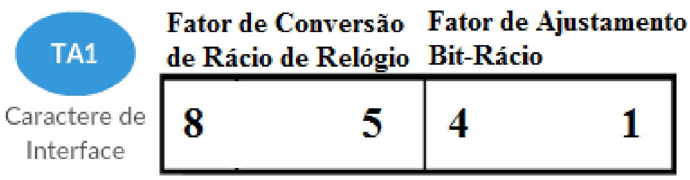

Figura 2.8: Configuração do Caractere de Interface TA1.

2.4.3 Caracteres de Interface (TAi, TBi, TCi e TDi)

Estes caracteres definem a interface de comunicação. Tal como referido anteriormente, o Caractere de Formato T0 indica a presença ou a ausência dos Caracteres de Interface e por isso estes caracteres são ditos opcionais. Nesta subsecção são analisados os Caracteres de Interface emitidos na ATR do CC, quando presentes.

- Caractere TA1

  O Caractere TA1 elucida as caraterísticas básicas da comunicação série, revelando informação que permite modificar o tempo de bit de um caractere na transmissão de dados. Este caractere fornece os novos fatores F e D permitidos no comando PPS (Subsecção 2.5), que irão influenciar o RT pela expressão (2.2) [Iso.org, 2006].

  A Figura 2.8 ilustra a estrutura do caractere TA1. Os 4 bits mais significativos do Caractere TA1 determinam o novo fator F, seguindo a Tabela A.1. Por outro lado, os 4 bits menos significativos determinam o novo fator D, conforme a Tabela A.2.

  Por defeito, são utilizados os valores de $F = 372$ e $D = 1$ na permuta de dados subsequentes, caso estes valores não sejam alterados posteriormente numa operação PPS (ver Subsecção 2.5). Utilizando os valores por defeito, resulta o mesmo RT definido inicialmente na transmissão de caracteres na ATR [Iso.org, 2006].

  Por observação da Tabela 2.2 verifica-se para ambas as ATRs que TA1=0x95. Descodificando este caractere obtêm-se como candidatos F=512 e D=16, para uma frequência de sinal de relógio máxima de 5 MHz.

- Caractere TB1

  O Caractere TB1 transmite os requerimentos de tensão e corrente a aplicar no terminal VPP (pino C6, Figura 2.2) caso o micro-controlador do cartão necessite da aplicação de uma tensão externa para programar e apagar a sua memória EEPROM [Iso.org, 2006].

  Apenas nas respostas ATR do tipo 1 e 2 será transmitido este caractere. Quando transmitido, o seu valor será TB1=0x00 o que indica que o terminal VPP não se encontra ligado internamente ao circuito do cartão, e por isso a sua interpretação poderá ser descartada [Rankl and Effing, 2010].

- Caractere TC1

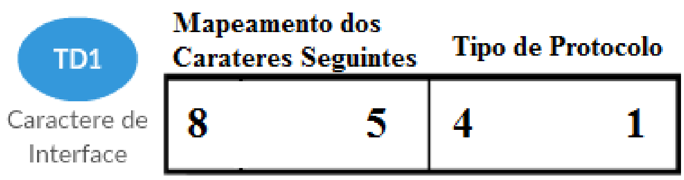

Figura 2.9: Configuração do Caractere de Interface TD1.

O Caractere de Interface TC1 é usado para determinar o atraso mínimo entre dois caracteres consecutivos (*Guardtime*). Assim, o Caractere TC1 corresponde ao número de ETUs a adicionar à Pausa entre dois caracteres consecutivos.

Apenas nas respostas ATR do tipo 1 e tipo 2 será transmitido este caractere. Quando transmitido, o seu valor será TC1=0x00 o que corresponde ao valor GT definido por defeito ($GT = 12ETU$) (ver Subsecção 2.3) [Iso.org, 2006].

- Caractere TD1

  O Caractere TD1 à semelhança do Caractere de Formato T0, indica a presença de novos Caracteres de Interface TD2, TC2, TB2 e TA2, assim como o protocolo admitido na comunicação de dados subsequentes [Iso.org, 2006]. A Figura 2.9 ilustra a sua estrutura.

  Os 4 bits mais significativos do Caractere TD1 indicam a presença dos Caracteres de Interface TD2, TC2, TB2 e TA2, respetivamente. Os 4 bits menos significativos transmitem o protocolo de comunicação, entre os quais se destacam a Transmissão Assíncrona de Bytes (Protocolo $T = 0$, Secção 3.2) ou Transmissão Assíncrona de Blocos (Protocolo $T = 1$). Quando ausente na transmissão, o protocolo empregue por defeito é o Protocolo $T = 0$ [Iso.org, 2006].

  No caso específico do CC, as ATRs do tipo 1 e tipo 2 não transmitem este caractere, logo não é transmitido mais nenhum Caractere de Interface e tem-se por defeito o protocolo de comunicação $T = 0$. No caso da resposta do tipo 3 obtém-se TD1=0x40, o que compromete o envio do Caractere de Interface TC2 e igualmente protocolo de comunicação $T = 0$.

- Caractere TC2

  O Caractere TC2 é um caractere especifico ao protocolo de comunicação $T = 0$ (comunicação assíncrona de transmissão de bytes) e transmite o tempo WI ("*Waiting Time*"). O parâmetro WI permite determinar o tempo WT, responsável pelo intervalo máximo entre caracteres consecutivos antes da comunicação com o CC se manifestar bloqueada ou sem resposta [Iso.org, 2006]. O tempo WI e o tempo WT relacionam-se pela seguinte expressão:

$$WT = 960 \times WI \times ETU. \tag{2.3}$$

Tabela 2.3: Significados possíveis do Byte Indicador de Categoria (T1).

Valor	**Significado**
0x00	Dados emitidos em formato *COMPACT-TLV* seguidos de três Bytes Indicadores de Categoria (não em *COMPACT-TLV*)
0x10	Os dados emitidos são referentes a uma diretória do cartão
0x80	Dados emitidos em formato *COMPACT-TLV*, cujo o último campo refere-se ao Indicador de Estado
0x81 a 0x8F	Reservado para uso futuro
Outros Valores	Formato Proprietário

Nas respostas tipo 1 e tipo 2 este caractere não é transmitido e, por defeito, tem-se $WI = 10$. No entanto, na resposta do tipo 3 temos TC2=0xFF, o que implica que $WI = 255$.

2.4.4 Caracteres Históricos

Os Caracteres Históricos são remetidos na resposta ATR após devolvidos os Caracteres de Interface e indicam as caraterísticas operacionais do cartão. Estes caracteres são opcionais[1] e indicam, entre outros, informações sobre o acesso a ficheiros, informações sobre o emissor do cartão e o estado do cartão [Rankl and Effing, 2010]. Embora nem todos os Caracteres Históricos resultantes da ATR se encontrem normalizados, existe um esforço refletido na norma ISO/IEC 7816 parte 4 para normalizar estes caracteres por forma a facilitar a identificação de um SC pela interface de leitura. Por este motivo, apenas será efetuada uma breve descrição sobre o significado dos Caracteres Históricos devolvidos.

2.4.4.1 Estrutura Geral

O primeiro Caractere Histórico a ser transmitido numa resposta ATR é denominado por "Byte Indicador de Categoria" (T1), e indica o formato da cadeia de Caracteres Históricos transmitida. Caso este caractere tenha valor T1=0x00, T1=0x10 ou T1=0x8X o seu significado correspondente é resumido na Tabela 2.3. Quaisquer outros valores indicam um formato proprietário do emissor e por isso não é possível a análise dos Caracteres Históricos envolvidos [Iso.org, 2013b].

Análisando os Caracteres Históricos obtidos nas ATRs emitidas no CC, nas respostas tipo 1 e tipo 2 (Tabela 2.2) obtém-se T1=0x80, o que significa que os Caracteres Históricos são devolvidos no formato *COMPACT-TLV* (*COMPACT Tag-Lenght-Value*), onde o último campo transmitido neste formato reflete o estado do cartão.

[1] Carateres opcionais são caracteres que podem ser omitidos da sua transmissão.

O primeiro byte de cada bloco transmitido no formato *COMPACT-TLV* define o significado (4 bytes mais significativos) e tamanho do campo de dados (4 bytes menos significativos) seguinte [Ferrari *et al.*, 1998].

No caso da ATR do tipo 3 da Tabela 2.2, obtém-se o Byte Indicador de Categoria com valor T1=0xD0. Segundo o disposto na Tabela 2.3, este caractere reflete um formato proprietário na transmissão de caracteres e por isso não é possível retirar mais nenhuma conclusão dos Caracteres Históricos emitidos para este tipo de cartão.

2.4.4.2 Elementos de Dados

Continuando a análise para as respostas do tipo 1 e 2, no primeiro byte do primeiro bloco de dados no formato *COMPACT-TLV* tem-se T2=0x31, o que reflete uma *Tag=3* e *Lenght=1*. Assim sendo, o próximo caractere será denominado por "Byte de Dados do Serviço" [Iso.org, 2013b], que descreve a disposição da estrutura de ficheiros do SO contido no CC. O valor T3=0x80 indica que é possível a seleção de aplicações pela especificação do nome ADF (*Application Dedicated File*) e que o cartão possui um ficheiro MF (*Master File*) (descrito na Secção 3.1).

O quarto Caractere Histórico transmitido refere-se ao próximo bloco de dados no formato *COMPACT-TLV*. Assim, tendo T4=0x65 obtêm-se *Tag=6* e *Lenght=5*. Desta forma, os próximos 5 caracteres (*Lenght=5*) transmitidos referem-se a dados de pré-emissão (*Tag=6*) [Iso.org, 2013b]. Estes dados são próprios do fabricante, não existindo até à data uma normalização inequívoca sobre estes caracteres e por isso não foi possível a sua interpretação.

Por fim, o último bloco de dados, em comprimento com o disposto no caracter T1, deverá referir-se ao Indicador de Estado do cartão. O último bloco de dados tem como cabeçalho T10=0x83, o que traduz o estado do cartão (*Tag=8*) nos restantes 3 bytes devolvidos (*Lenght=3*) [Iso.org, 2013b]. O próximo caracter a ser transmitido é chamado de byte "*Life Cycle State*" (LCS). No entanto, temos T11=0x00, o que indica que o LCS não é indicado. Por fim, os restantes dois caracteres são denominados por "Bytes de Estado" SW1 e SW2 respetivamente. No caso em estudo tem-se T12=0x90 (SW1=0x90) e T13=0x00 (SW2=0x00), que expressa que o CC se encontra totalmente operacional e aguarda a receção de comandos por parte do TL.

A Figura 2.10 resume a estrutura e significado dos Caracteres Históricos para as respostas ATR do tipo 1 e tipo 2.

2.5 Protocol Parameter Selection (Opcional)

Após a sua ativação, o CC emite a cadeia de caracteres ATR através dos seus parâmetros de comunicação por defeito. Caso estes parâmetros não sejam alterados, a comunicação com o cartão decorre com os parâmetros estabelecidos por defeito. No entanto, estes parâmetros podem não ser ideais face à variedade de leitores disponíveis no mercado. Posto este cená-

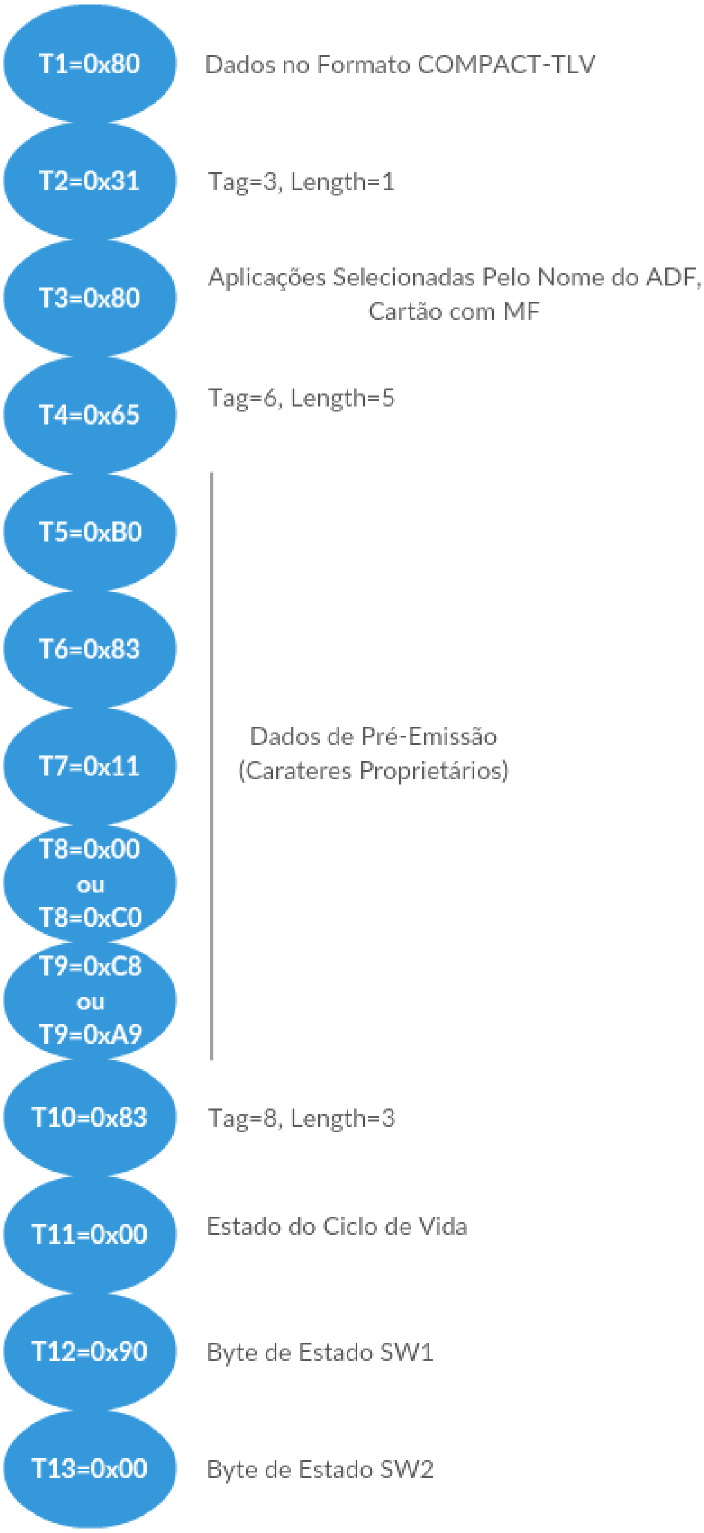

Figura 2.10: Significado dos Caracteres Históricos para as respostas ATR do tipo 1 ou tipo 2.

rio, o CC admite a possibilidade de alterar os parâmetros de comunicação através de um procedimento denominado por "*Protocol Parameter Selection*" (PPS) [Iso.org, 2006].

Caso na ATR devolvida seja indicado um novo valor de Fator F e/ou um novo Fator D

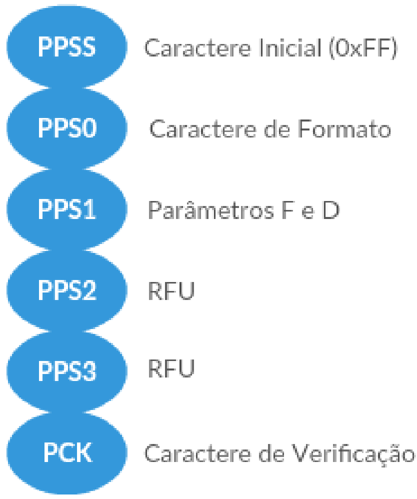

Figura 2.11: Formato do comando PPS.

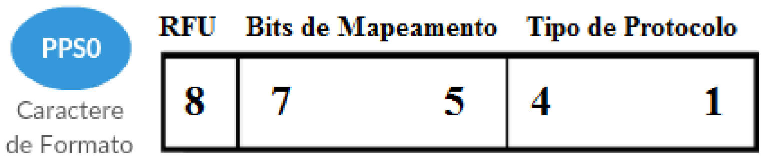

Figura 2.12: Estrutura do Caractere de Formato PPS0.

através do Caractere de Interface TA1, estes valores podem ser usados por forma a alterar o RT entre o CC e o TL, por meio das expressões (2.1) e (2.2).

O procedimento do comando PPS consiste no envio de uma cadeia de caracteres pelo TL para o CC após o procedimento de ATR. Na secção 2.4.3 foi visto que para ambas as ATR devolvidas, o Caractere TA1 indica como fatores candidatos $F = 512$ e $D = 16$, tendo sido aplicada um sinal de relógio de frequência $F_{CLOCK} = 4$ MHz. Foi igualmente visto que o CC apenas admite o protocolo de comunicação $T = 0$.

Quando admitido, a sequência de comandos referente ao comando PPS deve ser assegurada imediatamente após ser devolvida a sequência ATR. O pedido PPS consiste no envio sequencial de um Caractere Inicial PPSS codificado com valor hexadecimal 0xFF, seguido de um Caractere de Formato PPS0, três Caracteres de Parâmetros PPS1, PPS2 e PPS3 (opcionais), e finalizado por um Caractere de Verificação PCK [Iso.org, 2006]. A sequência do comando PPS é ilustrada na Figura 2.11.

O Caractere de Formato PPS0 possui a configuração ilustrada na Figura 2.12. Os bits de mapeamento são usados para indicar a presença dos caracteres opcionais, pelo que a presença do nível lógico '1' nos bits 5, 6 e 7 indicará a presença dos caracteres PPS1, PPS2 e PPS3, respetivamente. No caso do CC, apenas é necessário transmitir o caractere opcional PPS1, pelo que os caracteres PPS2 e PPS3 são descartados. Desta forma, os bits de mapeamento terão a codificação binária '001'. O tipo de protocolo admitido é codificado pelos bits 4 a 1, onde o protocolo $T = 0$ corresponde à codificação binária '0000'. Conjugando estes valores

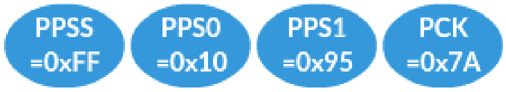

Figura 2.13: Comando PPS enviado ao CC.

tem-se PPS0=0x10.

O Caractere Opcional PPS1 indica os novos valores de F e D a serem usados na comunicação e têm a mesma codificação que o Caractere de Interface TA1, logo PPS1=TA1=0x95 (ver Subsecção 2.4.3).

Por fim, o Caractere de Verificação é processado por forma que, o resultado de um OU-Exclusivo entre todos os caracteres emitidos desde o Caractere PPSS até ao Caractere PCK inclusive, seja igual a zero, tendo-se portanto PCK=0x7A.

A Figura 2.13 resume o valor dos caracteres enviados na transmissão do comando PPS. No caso da operação ser bem sucedida, o CC deverá responder com a mesma sequência de caracteres emitida no envio do comando PPS [Iso.org, 2006] e, por aplicação da equação 2.2 com o ajustamento dos Fatores F e D, RT = 125 Kbit/s para a transmissão de dados precedentes.

Capítulo 3

Operação do Cartão de Cidadão

O elemento mais importante de um SC é o seu SO. É este que transforma um vulgar cartão dotado de um microprocessador num sistema complexo capaz de guardar e gerir dados. Como referido na Secção 1.2.3 o CC opera sob o Sistema Operativo OS ICitizen V2 64K da Axalto. O SO da Axalto possibilita não só o acesso e escrita em ficheiros, ou a criação e eliminação de dados, mas também protege dados sob privilégios de segurança.

Muitos SOs no mercado recorrem a estruturas orientadas a registos onde fazem uso do protocolo de comunicação $T = 1$, o que não é caso do CC. O SO que equipa o CC recorre a uma estrutura transparente de dados, isto é, os dados são armazenados em cadeias de bytes consecutivas de forma diferenciada, recorrendo ao protocolo T=0 para comunicação com dispositivos externos.

3.1 Estrutura de Ficheiros

A estrutura de ficheiros no CC encontra-se armazenada sob a forma de "árvore de ficheiros", onde existe um ficheiro raiz de onde derivam todos os restantes ficheiros. A Figura 3.1 ilustra este tipo de estrutura de ficheiro onde se reconhecem os seguintes quatro tipos de ficheiros [Rankl, 2007]:

- Master File (MF)

  O ficheiro MF é único no cartão e demarca a raiz de toda a estrutura de ficheiros, onde todos os outros ficheiros derivam deste [Iso.org, 2013b]. O ficheiro MF possui propriedades que o assemelham a uma Diretoria, sendo que este ficheiro não armazena dados diretamente e apenas contem outros ficheiros ou Diretorias. O ficheiro MF pode ser comparado com a diretoria 'C:' na estrutura de ficheiros de um PC com o sistema operativo Windows.

  Todos os ficheiros presentes no CC são invocados por um conjunto variável de bytes denominado por "*File ID*" (FID). O FID do ficheiro MF é constituído por 2 bytes onde

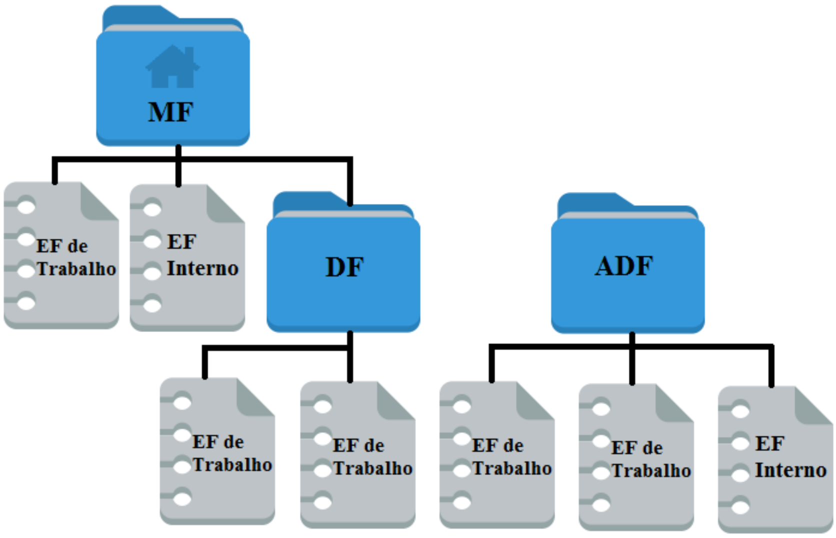

Figura 3.1: Estrutura de Ficheiros do CC.

FID=0x3F00. Após o procedimento de ATR este ficheiro é selecionado por defeito [Iso.org, 2013b].

- Dedicated File (DF)

  Um ficheiro DF é uma Diretoria que armazena outros ficheiros e/ou Diretorias. À semelhança do ficheiro MF, um ficheiro DF não armazena dados diretamente.

  O FID de um ficheiro DF é constituído por 2 bytes de valor variável sem restrições de valor, ou seja, o identificador FID pode tomar valores de 0x0000 a 0xFFFF (excepto 0x3F00).

- Application Dedicated File (ADF)

  Um ficheiro ADF é um tipo de ficheiro especial de ficheiro DF. Um ficheiro ADF assegura todos os ficheiros de uma aplicação específica e possui a particularidade de poder ser invocado sem qualquer relação ao ficheiro MF [Rankl, 2007].

  O FID de um ficheiro ADF possui tamanho variável e pode ser constituído por 1 a 16 bytes, sem restrições de valor.

- Elementary File (EF)

  Os ficheiros EF alojam informações de dados derivadas das aplicações contidas no CC e por isso são conhecidos por "Ficheiros de Dados" [Rankl, 2007]. Estes ficheiros são os ficheiros terminais da estrutura de dados do CC. Por outras palavras, os ficheiros EF têm a particularidade de não poder alojar outros ficheiros e localizam-se sempre na camada acima de outras diretorias.

  Existem dois tipos de ficheiros EF distintos: "Ficheiros EF de Trabalho" e "Ficheiros EF Internos" [Iso.org, 2013b]. Os Ficheiros EF de Trabalho são Ficheiros de Dados onde se encontram armazenados os dados utilizados por uma aplicação e que são disponibilizados externamente a uma interface de leitura, através da invocação de comandos apropriados (exemplo de ficheiro que contem armazenada a morada do cidadão). Por outro lado, os Ficheiros EF Internos são Ficheiros de Dados usados pelo SO do cartão para armazenar dados privados da aplicação e nunca são disponibilizados externamente (exemplo de ficheiro onde se encontra armazenado o PIN de Morada que permite acesso à morada do cidadão).

À semelhança de um ficheiro DF, o FID de um ficheiro EF é constituído por 2 bytes de valor variável sem restrições de valor, ou seja, o identificador FID pode tomar valores de 0x0000 a 0xFFFF (excepto 0x3F00).

3.2 Protocolo de Comunicação T=0

O protocolo de comunicação T=0 define o processo de troca de informação entre o CC e o exterior, assim como os mecanismos de deteção de erros na transmissão de dados. Este protocolo é orientado à comunicação de bytes de dados[1] e foi concebido por forma a maximizar a sua eficiência e simplicidade de utilização, e minimizar os recursos do micro-controlador de um SC[Rankl and Effing, 2010]. O protocolo de comunicação T=0 encontra-se especificado na norma ISO/IEC 7816 parte 3 [Iso.org, 2006].

A deteção de erros na transmissão de dados no protocolo T=0 é feita através do envio de um bit de paridade (ver Figura 2.3). Sempre que seja detetado um erro no bit de paridade, o recetor da comunicação deverá forçar a linha de dados I/O ao nível lógico ativo-baixo durante a transmissão do primeiro período do bit de paragem (recorde-se que o bit de paragem é caraterizado pela linha I/O em estado ativo alto durante 2 ETUs). A retransmissão do byte malfeito por parte do transmissor deverá ser assegurada imediatamente após a sua deteção. Este mecanismo de deteção de erros permite detetar e retransmitir bytes malfeitos, contudo, não permite detetar a perda de um byte durante a transmissão. Esta situação pode levar à ocorrência de um *deadlock* se um dos lados da transmissão estiver à escuta de um número específico de bytes e esse número nunca for atingido, o que poderá levar ao reset do micro-controlador do cartão após ultrapassado o tempo máximo entre bytes consecutivos definido (WT) e iniciar todo o processo de comunicação novamente [Rankl and Effing, 2010].

Quando se aborda um protocolo de comunicação, geralmente é possível expor a temática em termos do conhecido modelo de camadas OSI ("*Open Systems Interconnection*"). Este modelo de camadas carateriza e normaliza as funções de comunicação de um sistema de comunicação, sem ter em consideração a sua estrutura interna e tecnologias subjacentes. O objetivo deste modelo é promover a descrição da interoperabilidade dos diversos sistemas de comunicação com protocolos padrão. Assim, o modelo de camadas descreve a comunicação

[1] Um protocolo orientado à comunicação de bytes significa que um byte é a unidade mínima de informação transmitida.

Figura 3.2: Modelo de camadas protocolar OSI.

entre duas entidades em termos de sete camadas protocolares distintas, onde cada camada se encontra estritamente separada e servida apenas pelas suas camadas adjacentes. Cada camada define um conjunto de serviços para a pilha protocolar, tal como ilustrado na Figura 3.2.

3.3 Estrutura de Mensagens

Os pressupostos de comunicação entre o TL e o CC são sempre baseados no princípio de mestre-escravo. Isto significa que o TL, que atua como mestre na comunicação, envia um comando ao CC e este gera uma resposta referente ao comando recebido, enviando a resposta correspondente ao terminal através da linha de comunicação. Como escravo na comunicação de dados, o CC nunca envia uma resposta sem primeiro receber o comando correspondente do TL.

A troca de comandos e respostas é assegurada na Camada de Ligação pela estrutura de mensagens TPDU (*"Transmission Protocol Data Units"*), definidas no protocolo de comunicação T=0. Por sua vez, estas mensagens são sustentadas a nível de *Software* pela estrutura de mensagens APDU na Camada de Aplicação da pilha protocolar [Jurgensen and Guthery, 2002], tal como ilustrado na Figura 3.3. Existe uma pobre separação das camadas protocolares no protocolo de comunicação T=0, que advém da tentativa de tornar o protocolo o mais responsivo e eficiente possível [Jurgensen and Guthery, 2002].

A estrutura de mensagens APDU encontra-se completamente estandardizada e especificada na norma ISO/IEC 7816, parte 4 [Iso.org, 2013b]. Existe um grande número de comandos sustentados nesta norma, porém todos os comandos APDU não necessários numa dada aplicação são removidos durante a programação do SC por forma a otimizar o SO e a

Figura 3.3: Arquitetura de Mensagens entre o TL e a aplicação do CC.

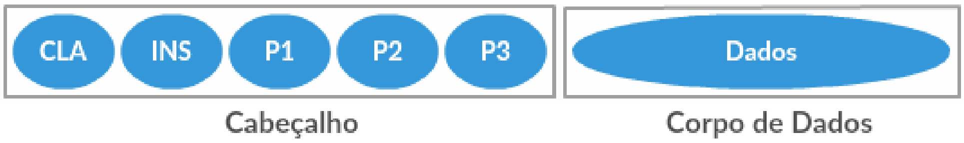

Figura 3.4: Estrutura de um comando TPDU.

sua alocação de memória [Rankl and Effing, 2010]. Nesse sentido, apenas serão relatados os comandos TPDU/APDU utilizados na implementação deste projeto.

3.3.1 Comando TPDU

Os comandos TPDU consistem em um Cabeçalho e no Corpo de Dados. A Figura 3.4 ilustra a estrutura de um comando TPDU, que é decomposto em [Jurgensen and Guthery, 2002]:

- Classe de Comando (CLA)

  A Classe de Comando é usada para identificar aplicações e os seus respetivos conjuntos de instruções. Os possíveis valores de CLA encontram-se definidos no Anexo A.2.1. No caso em estudo, a conceção do CC é baseada na estandardização ISO/IEC 7816 parte 4, e por isso este comando terá codificação binária 0x0X (mais especificamente CLA=0x00).

- Byte de Instrução (INS)

  Dentro de um dado valor de Classe de Comando, o Byte de Instrução é usado para identificar a instrução enviada no comando TPDU. A norma ISO/IEC 7816 parte 4 identifica uma série de instruções praticadas no acesso de ficheiros e em funções de segurança do CC. As instruções contidas na classe CLA=0x00 (instruções definidas na norma ISO/IEC 7816-4) para o protocolo de transmissão T=0 empregues neste projeto são listadas e especificadas na Tabela 3.1. Uma listagem completa das instruções especificadas na norma ISO/IEC 7816 parte 4 pode ser encontrada em Anexos A.2.2.

- Bytes de Parâmetro P1 e P2

  Os Bytes de Parâmetro P1 e P2 são dependentes do Byte de Instrução (INS) especificado, e promovem parâmetros de endereçamento ou controlo à instrução especificada.

Tabela 3.1: Conjunto de valores possíveis para o Byte de Instrução (INS), para o protocolo de transmissão T=0, empregues neste projeto.

INS	Nome do Comando
0xA4	Select File
0xC0	Get Response
0xB0	Read Binary
0xD6	Update Binary
0x0E	Erase Binary
0x20	Verify

Tome-se por exemplo o caso da instrução *Select File* (INS=0xA4). Como será descrito na Secção 3.4.1, esta instrução envolve a seleção de um ficheiro existente na memória do cartão. Neste caso particular o parâmetro P1 é usado para determinar como o ficheiro será referenciado (referência por um identificador de ficheiro ou identificador de aplicação). Por sua vez, o parâmetro P2 será usado para clarificar a instrução do comando ou distinguir qual o ficheiro selecionado.

- Byte de Parâmetro P3

  Não obstante dos restantes Bytes de Parâmetros, o valor do byte P3 também é dependente do Byte de Instrução especificado. Sempre que o comando enviado ao CC pertença a uma instrução que envie dados anexos ao comando TPDU (Figura 3.5), este campo especifica o número de bytes de dados remetidos pelo comando. Por outro lado, sempre que o comando enviado ao CC pertença a uma instrução que envolva a devolução de bytes de dados na resposta (Figura 3.6) este campo especifica o número de bytes de dados esperados. Caso a instrução enviada não implique o envio nem a receção de dados, então o seu valor será P3=0x00.

- Corpo de Dados

  Este campo destina-se ao envio de dados entre os dois pontos de comunicação. Este campo apenas está disponível caso a instrução especificada no comando TPDU admita o envio de dados. O número de bytes de dados enviados é especificado pelo byte P3, que pode ter um mínimo de 1 byte de dados (caso P3=0x01) e um máximo de 256 bytes de dados (caso P3=0x00).

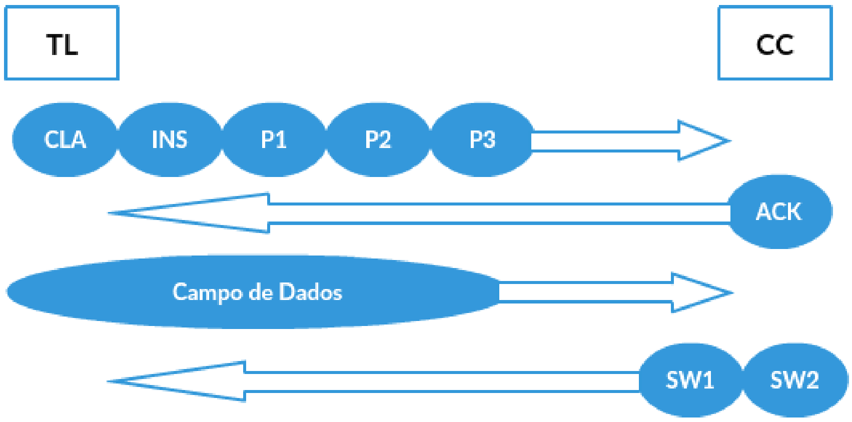

Figura 3.5: Representação do par comando-resposta no envio de dados do TL ao CC.

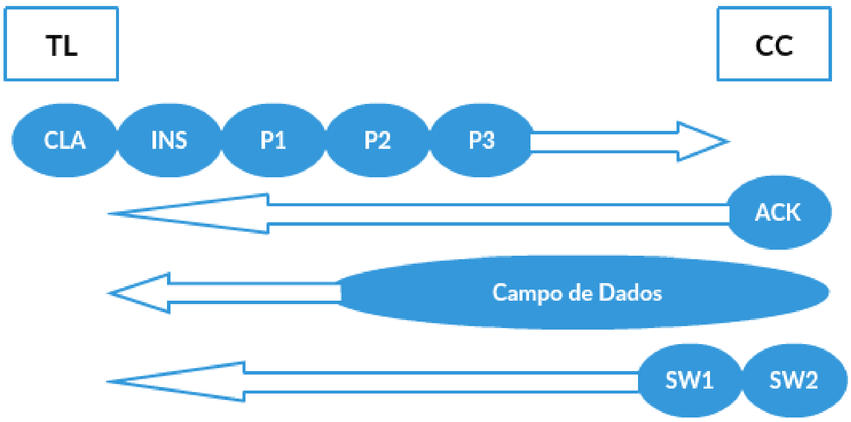

Figura 3.6: Representação do par comando-resposta no envio de dados do CC ao TL.

3.3.2 Resposta TPDU

Sempre que é enviado um comando TPDU do TL ao CC, este responde ao TL com o envio de Bytes de Precedência [Jurgensen and Guthery, 2002], tal como ilustrado nas Figuras 3.5 e 3.6. São conhecidos três tipos de Bytes de Precedência:

- Byte ACK

  Byte que indica a receção de um comando por parte do cartão. Este byte destina-se a notificar o TL da receção do comando TPDU precedente e trata-se simplesmente de uma repetição do Byte de Instrução (ACK=INS).

- Byte NULL

  Para manter o fluxo de dados na comunicação, por vezes o cartão responde com um byte de valor específico (0x60) por forma a indicar ao TL que o CC se encontra a processar o comando recebido. Desta forma o TL toma conhecimento que o comando enviado ainda se encontra em processamento e não envia novos comandos, nem desabilita o cartão, enquanto não surgir uma resposta por parte do CC.

- Trailer SW1 e SW2

  O Byte de Estado SW1 é usado para especificar o estado da resposta enquanto que o byte SW2 assiste o byte SW1 ao descrever inequivocamente o resultado do processo. O Byte de Estado SW1 poderá ter valores de SW1=0x9X, ou SW1=0x6X exceto SW1=0x60 (Byte NULL).

  Atualmente existem mais de 50 combinações de Bytes de Estado. A listagem completa de configurações especificadas pela norma ISO/IEC 7816 parte 4 dos Bytes de Estado e seus respetivos significados pode ser encontrada em Anexos A.2.3.

Dada a fraca separação das Camadas Protocolares no protocolo de comunicação T=0, os Bytes de Precedência ACK e NULL são proprietários da Camada de Ligação, ao passo que, os Bytes de Estado SW1 e SW2 são resultado da Camada de Aplicação [Rankl and Effing, 2010].

3.3.3 Comando e Resposta APDU

O SO do CC faz uso do protocolo definido na Camada de Aplicação da pilha protocolar para trocar informação entre uma aplicação do cartão e as camadas protocolares acima, numa estrutura de mensagens chamada APDU. A estrutura de mensagens APDU não difere muito da estrutura de mensagens TPDU já relatada, devido à pobre separação de camadas implementada no protocolo de comunicação T=0. A estrutura de mensagens APDU encontra-se convenientemente elucidada na norma ISO/IEC 7816 parte 4 [Iso.org, 2013b].

Das diferenças estruturais entre os comandos TPDU e os comandos APDU, é de salientar a ausência do parâmetro P3 no Cabeçalho do comando APDU. Porém, este byte é transcrito no Corpo de Dados sob a identificação de Lc (se o Byte de Instrução INS for referente a um comando que envie dados ao cartão) ou Le (se o Byte de Instrução INS especificar um comando que receba dados provenientes do cartão). Caso o Byte de Instrução INS não envolva um cenário de envio ou receção de dados ao cartão, o Byte de Parâmetro P3 na estrutura de comando TPDU é simplesmente descartado na estrutura de comando APDU. Os restantes bytes são reproduzidos do comando TPDU para o comando APDU precisamente com o mesmo significado.

Sempre que é processado um comando APDU pelo CC é mandatório o envio de um *trailer* de bytes em função do resultado da execução do comando recebido (resposta). Este *trailer* de bytes é constituído pelos Bytes de Estado SW1 e SW2 relatados na subsecção anterior.

3.4 Comandos Básicos de Comunicação

Todos os comandos TPDU implementados neste projeto seguem as diretivas na norma ISO/IEC 7816 parte 4 [Iso.org, 2013b]. Embora esta norma relate a implementação de cerca de 40 comandos, apenas os comandos descritos na Tabela 3.1 são utilizados com a finalidade

Tabela 3.2: Parâmetros do comando *Select File*.

CLA	'0x00' (como referido na Secção 3.3.1)
INS	'0xA4'
P1	'0x00' ou '0x04'
P2	'0x00'
P3	Tamanho do identificador do ficheiro a selecionar (número de bytes do Corpo de Dados)

de alcançar os objetivos propostos nesta dissertação, e por esse motivo são merecedores de uma descrição detalhada. A listagem completa dos comandos APDU especificadas na norma ISO/IEC 7816 parte 4 pode ser encontrada em Anexos A.2.2.

3.4.1 Comando Select File

Como já abordado, nesta dissertação a disposição de ficheiros no CC encontra-se organizada por uma hierarquia de ficheiros. Após completo o mecanismo ATR, o ficheiro raiz da estrutura, denominado por ficheiro MF, é automaticamente selecionado. Deste modo, o comando *Select File* permite a seleção de ficheiros ou diretorias subsequentes, para manipulação. A estrutura do comando *Select File* é dada pela Figura 3.4, onde se enquadram os parâmetros da Tabela 3.2.

A estrutura de ficheiros do CC compromete ficheiros e aplicações. A seleção do parâmetro P1 no comando *Select File* dependerá do tipo a selecionar, permitindo a seleção de ficheiros pelo seu identificador (caso P1=0x00) ou pelo identificador da aplicação correspondente (P1=0x04). O parâmetro P2 é dedicado a estruturas lineares ou cíclicas de ficheiros onde a informação é disposta numa sequência de registos de dados [Iso.org, 2013b], o que não é o caso da estrutura de ficheiros do CC. O CC tem uma estrutura transparente de ficheiros, onde a informação é disposta numa sequência bytes consecutivos, e por esta razão o parâmetro P2 terá valor 0x00.

Após enviado o comando, o CC deverá responder com um byte denominado ACK. Este byte tem a finalidade de indicar que o cartão reconheceu o comando recebido e encontra-se agora a aguardar dados. Após a receção deste byte, deverá ser enviado o Corpo de Dados que contém o identificador do ficheiro ou da aplicação a selecionar, com tamanho definido pelo parâmetro P3. Depois de remetido o Corpo de Dados, o cartão deverá enviar uma resposta contendo dois Bytes de Estado SW1 e SW2 (tal como ilustrado na Figura 3.5).

Tabela 3.3: Parâmetros do comando *Get Response*.

CLA	'0x00' (como referido na Secção 3.3.1)
INS	'0xC0'
P1	'0x00'
P2	'0x00'
P3	Número de bytes a serem obtidos (P3=SW2)

3.4.2 Comando Get Response

Tal como relatado ao longo da descrição do par comando-resposta, o CC jamais envia ao TL dados que não tenham sido solicitados através do envio de um comando. No entanto, a execução de certos comandos por parte do cartão torna disponível uma série de bytes não solicitados. Nestes casos, o cartão envia uma resposta onde informa o terminal que existe um número específico de bytes que podem ser obtidos.

Um exemplo desta descrição é a seleção de um ficheiro através do comando *Select File*. Por vezes, após um dado ficheiro ser selecionado com sucesso, são disponíveis uma série de bytes cujo conteúdo revela as propriedades do ficheiro selecionado, tais como o número de bytes contidos no ficheiro. Neste caso, após a execução bem-sucedida do comando *Select File*, o cartão responde com os Bytes de Estado SW1=0x61 e SW2=0xXX, onde SW2 indica o número de bytes disponíveis para leitura através do comando *Get Response*.

A estrutura do comando *Get Response* é dada pela Figura 3.4, onde se enquadram os parâmetros da Tabela 3.3.

No caso da execução do comando ser bem sucedida, a resposta ao comando deverá conter os bytes solicitados pelo TL, após o envio do Byte de Precedência ACK. Após a execução do comando pelo CC, dois Bytes de Estado deverão ser enviados ao TL, transmitindo o estado da execução do comando (tal como ilustrado na Figura 3.6).

3.4.3 Comando Read Binary

O comando *Read Binary*, tal como o nome indica, permite ler o conteúdo de um ficheiro EF. Para utilização bem-sucedida deste comando é necessária a seleção prévia do ficheiro desejado através do comando *Select File* (Subsecção 3.4.1).

A estrutura do comando *Read Binary* é dada pela Figura 3.4, onde se enquadram os parâmetros da Tabela 3.4. Os parâmetros P1 e P2, em conjunto (16 bytes), formalizam a posição do ficheiro a iniciar a leitura, onde P1=0x00 e P2=0x00 expõe o primeiro caractere contido no ficheiro. Por sua vez, o parâmetro P3 representa o número de bytes esperados na resposta do cartão após a execução do comando, ou seja, o número de bytes a serem lidos no ficheiro a partir da posição dada por P1 e P2.

Tabela 3.4: Parâmetros do comando *Read Binary*.

CLA	'0x00' (como referido na Secção 3.3.1)
INS	'0xB0'
P1 **P2**	Posição de leitura de 0 a 32767 bytes
P3	Número de bytes a serem lidos do ficheiro

Tabela 3.5: Parâmetros do comando *Update Binary*.

CLA	'0x00' (como referido na Secção 3.3.1)
INS	'0xD6'
P1 **P2**	Posição de leitura de 0 a 32767 bytes
P3	Número de bytes a serem atualizados no ficheiro

Em caso de sucesso, deverá ser devolvida pelo cartão a cadeia de caracteres especificada pelos parâmetros do comando enviado, após o envio do Byte de Precedência ACK. Depois da execução do comando pelo CC, o cartão responde com dois Bytes de Estado notificando o estado da execução do comando, tal como exibido na Figura 3.6.

3.4.4 Comando Update Binary

O comando *Update Binary* possibilita a atualização de caracteres existentes num ficheiro EF. Para utilização bem-sucedida deste comando é necessária a seleção previa do ficheiro desejado através do comando *Select File* (Subsecção 3.4.1).

A estrutura do comando *Update Binary* é dada pela Figura 3.4, onde se enquadram os parâmetros da Tabela 3.5. Os parâmetros P1 e P2, em conjunto (16 bytes), formalizam a posição do ficheiro para iniciar a escrita, onde P1=0x00 e P2=0x00 expõe o primeiro caractere contido no ficheiro. Por sua vez, o parâmetro P3 representa o tamanho do Corpo de Dados enviado pelo comando, ou seja, o número de caracteres a serem escritos no ficheiro a partir da posição dada por P1 e P2. O Corpo de Dados do comando deverá conter os caracteres a serem escritos na memória do cartão e deverá ser enviado após a receção do Byte de Precedência ACK, enviado após a receção e identificação do comando por parte do CC, tal como ilustrado na Figura 3.5.

Depois da execução do comando, o cartão remete dois Bytes de Estado notificando o estado da execução do comando. Em caso de sucesso, cada byte especificado no Corpo de Dados deverá substituir os bytes contidos no ficheiro especificado, nas posições seguintes à

Tabela 3.6: Parâmetros do comando *Erase Binary*.

CLA	'0x00' (como referido na Secção 3.3.1)
INS	'0x0E'
P1 **P2**	Posição de primeiro byte a ser limpo de 0 a 32767 bytes
P3	Número de bytes a serem limpos no ficheiro (P3=0x00 para limpar todo o ficheiro)

discriminada por P1 e P2.

3.4.5 Comando Erase Binary

O comando *Erase Binary* possibilita limpar os bytes já existentes num ficheiro EF de Trabalho (colocar no estado 0x00). Uma vez mais, para utilização bem-sucedida deste comando, é imperativo a seleção previa do ficheiro desejado através do comando *Select File* (Subsecção 3.4.1).

A estrutura do comando *Erase Binary* é dada pela Figura 3.4, onde se enquadram os parâmetros da Tabela 3.6. Este comando permite limpar todo o ficheiro (caso P1=0x00, P2=0x00 e P3=0x00) ou limpar apenas uma seleção de caracteres onde os parâmetros P1 e P2, em conjunto (16 bytes), formalizam a posição do primeiro caractere a limpar e P3 o número de caracteres seguintes.

De grosso modo, este comando não representa mais que a execução do comando *Update Binary* com todo o Corpo de Dados de valor 0x00 (ver Subsecção 3.4.4). Por este motivo o *Trailer* de Bytes de Estado devolvidos pelo comando *Erase Binary* é muito semelhante ao *Trailer* devolvido na resposta ao comando *Update Binary*.

Em caso de sucesso, todos os bytes especificados pelo comando *Erase Binary* deverão encontrar-se no estado 0x00 após a execução do comando.

3.4.6 Comando Verify

Sempre que a manipulação de um ficheiro ou aplicação exija a posse de privilégios de segurança (resposta SW1=0x69 e SW2=0x82) é necessário satisfazer essas mesmas condições. O comando *Verify* inicia um mecanismo de comparação entre um dado de referência armazenado no cartão (em ficheiro EF interno) e um dado enviado pelo utilizador, por forma a adquirir privilégios de segurança. O cartão poderá gravar o número de comparações falhadas por forma a bloquear o acesso a mecanismos de segurança caso um determinado número de tentativas incorretas consecutivas seja excedido. Este comando é utilizado, por exemplo, na validação da introdução do PIN de Morada por forma a aceder ao conteúdo da morada.

A estrutura do comando *Verify* é dada pela Figura 3.4, onde se enquadram os parâmetros da Tabela 3.7. Enquanto o parâmetro P1 terá valor '0x00', o parâmetro P2 permite especificar o identificador do dado de referência (PIN de Morada ou PIN de Autenticação).

Tabela 3.7: Parâmetros do comando *Verify*.

CLA	'0x00' (como referido na Secção 3.3.1)
INS	'0x20'
P1	'0x00'
P2	'0x81' (PIN de Autenticação) ou '0x83' (PIN de Morada)
P3	Tamanho do identificador do ficheiro a selecionar (número de bytes do Corpo de Dados)

Após enviado o comando, o CC responde com um byte denominado ACK, que não é mais que a repetição do Byte de Instrução contido no comando (ACK=INS). Após a receção deste byte, deverá ser enviado o Corpo de Dados que contém o dado de comparação, com tamanho definido pelo parâmetro P3. Depois de remetido o dado de comparação, o cartão deverá enviar uma resposta contendo dois Bytes de Estado SW1 e SW2, tal como se descreve na Figura 3.5.

Capítulo 4

Arquitetura do Leitor de CC

Para a utilização do CC neste projeto foi indispensável o conhecimento da organização de ficheiros e aplicações de interesse no SO do CC. Obviamente, este processo seria facilitado pela presença de documentação oficial que elucidasse de forma clara e acessível a organização de ficheiros e aplicações assim como a troca de comandos e respostas essenciais para a troca de informação, o que não é o caso[1].

Tendo como base a interpretação das ATRs conhecidas para o CC (Secção 2.4), sabe-se que o CC faz recurso do protocolo de comunicação T=0, tornando compreensível a utilização de comandos TPDU e significados das respostas envolvidas. Tendo como base os mecanismos de comunicação descritos nos capítulos anteriores nesta dissertação, foi necessário investigar os processos envolvidos na extração da informação no CC. Para o efeito empregou-se um leitor comercial, neste caso o SCR335 da SCM Microsystems©[Microsystems, 2011], por forma a estudar a informação trocada entre o CC e o leitor, num processo denominado por *Sniffing*. Recorde-se que este leitor é um TL Online, de acordo com a classificação atribuída na Secção 1.2.4. Neste sentido foi utilizado um analisador lógico - o AX da USBee©, fornecido pela *Acronym*. Este analisador permitiu capturar os pacotes de dados trocados entre o chip de contacto do CC e o leitor da SCM Microsystems©, enquanto os dados guardados no CC eram requeridos pela aplicação oficial do CC, disponível no site oficial. A análise morosa de todos os comandos TPDU trocados entre o leitor utilizado e o CC permitiu não só retirar conclusões importantes quanto à estrutura e organização dos dados guardados na memória do CC mas também simplificar o processo de comunicação entre os dois pontos de comunicação, eliminando um considerável número de comandos desnecessários.

Após a compreensão da organização dos dados de interesse no CC, procedeu-se à implementação de comandos TPDU por forma a atingir os objetivos propostos nesta dissertação. Os dados extraídos do cartão são enviados a um PC via interface USB-HID. Esta interface foi produzida com base na *framework* LUFA, tendo sido produzido um protocolo de comunicação para troca de informação entre o TL e o PC. Por fim, tendo em vista alcançar um

[1] A presente documentação oficial do CC pode ser consultada no site oficial http://www.cartaodecidadao.pt.

Figura 4.1: Seleção do Ficheiro ADF 0x604632FF000002.

produto final, foi produzido *Hardware* com base nos pressupostos deste projeto.

4.1 Seleção, Leitura e Gravação de Dados

Os dados de interesse do CC encontram-se armazenados num ficheiro do tipo ADF. Como já descrito na Subsecção 3.1, um ficheiro ADF possui independência em relação ao ficheiro MF, e por isso pode ser invocado a partir de uma qualquer diretoria do sistema de ficheiros. Este ficheiro ADF possui um FID constituído por 7 bytes de valor 0x604632FF000002. Este ficheiro é invocado através da execução do comando TPDU *Select File* com os parâmetros ilustrados na Figura 4.1. Repare-se que o Campo de Dados, com o FID do ficheiro desejado, é enviado após a resposta do CC com o Byte de Precedência ACK.

No âmbito desta dissertação, a aplicação selecionada contem 3 ficheiros de interesse. O Ficheiro 0xEF02 contem os dados pessoais do cidadão e dados intrínsecos ao cartão. Por sua vez, o Ficheiro 0xEF05 armazena os dados referentes à morada do cidadão e por fim, o Ficheiro 0xEF07 é o ficheiro destinado à escrita livre de dados até 1000 caracteres. Os dados armazenados e sua respetiva posição no ficheiro podem ser consultados na tabela disposta no Anexo A.3. Para tratamento dos dados contidos em cada ficheiro é obrigatória a seleção prévia do ficheiro respetivo através do comando *Select File*.

A Figura 4.2 traça os procedimentos necessários para o acesso ao conteúdo dos ficheiros contidos no CC, os quais se descrevem nas subsecções seguintes.

4.1.1 Dados do Cidadão e do Documento (0xEF02)

O Ficheiro 0xEF02 contém dados inerentes ao cidadão titular, tais como, Nome(s), Apelido(s), Nacionalidade, Data de Nascimento, Fotografia, etc. Este ficheiro conta ainda com informações sobre o próprio documento nomeadamente o Número de Documento (onde consta o Número de Identificação Civil), Entidade Emissora, Validade, Número de Série, Versão, etc.

De destacar que neste ficheiro encontra-se armazenada a fotografia do cidadão. A fotografia encontra-se arquivada em formato JPEG2000, sem qualquer mecanismo de segurança anexado. A escolha deste formato prendeu-se essencialmente por questões da gestão do espaço disponível na memória do cartão [Crocker *et al.*, 2010].

A seleção deste ficheiro para manipulação é realizada através do comando *Select File*. Na Figura 4.3 são exibidos os parâmetros adequados.

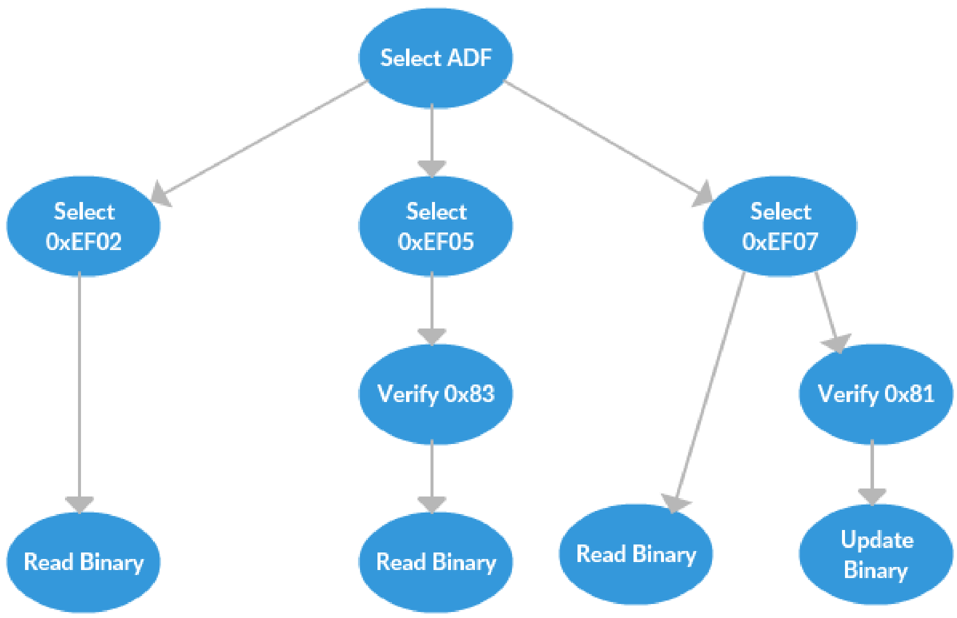

Figura 4.2: Estrutura de comandos TPDU necessários para acesso às funcionalidades dos ficheiros abordados.

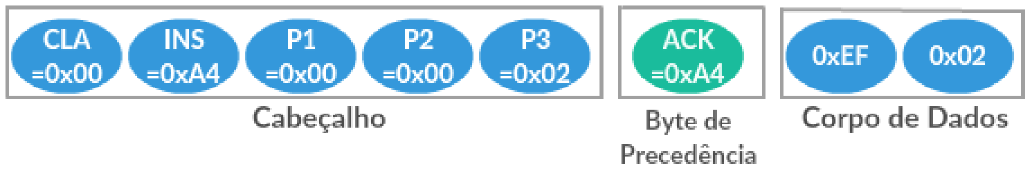

Figura 4.3: Seleção do Ficheiro 0xEF02.

4.1.1.1 Leitura de Dados

Sendo a estrutura de dados do CC uma estrutura transparente, onde a informação é armazenada numa série de bytes consecutivos, a leitura de dados do ficheiro 0xEF02 é realizada através do envio do comando TPDU *Read binary*, não sendo necessário ultrapassar nenhuma barreira de segurança adicional. Os Bytes de Parâmetros P1, P2 e P3 irão variar de acordo com a leitura do campo desejado[2]. Tal como descrito na Subsecção 3.4.3 para o comando Read Binary, os Bytes de Parâmetros P1 e P2 definem a posição de leitura no ficheiro, enquanto o Byte de Parâmetro P3 define o número de bytes a serem lidos.

Sabendo a posição no ficheiro do campo de interesse e o tamanho dos dados a serem obtidos seria trivial a leitura do conjunto de dados do cartão. No entanto, muitos dos dados solicitados neste projeto possuem tamanho variável e desconhecido antes da sua aquisição, tais como, Nome(s), Apelido(s), Filiação, etc. Porém, a análise aos bytes contidos no ficheiro permite concluir que os campos de dados de tamanho variável se encontram separados por bytes nulos (de valor 0x00). Tirando proveito deste facto, é lido um bloco de dados maior

[2]Pode ser consultada a posição de cada campo de dados contido no ficheiro no Anexo A.3.

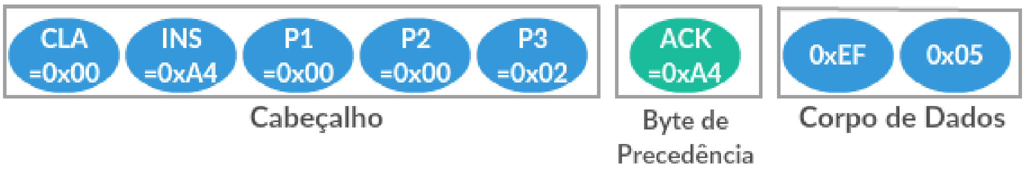

Figura 4.4: Seleção do Ficheiro 0xEF05.

que o tamanho dos dados esperados, sendo posteriormente aplicada uma filtragem dos dados extraídos por parte do TL antes de os enviar ao PC (extração de dados até encontrar um byte de valor 0x00).

Uma abordagem mais interessante, sobretudo caso se pretenda a extração de mais que um dado do cartão, é a solicitação de respostas em blocos de 256 bytes de dados até perfazer a secção de leitura de interesse ou fim do ficheiro. Isto é, envio do comando *Read Binary* com o Byte de Parâmetro P3=0x00 (256 bytes) e os bytes P1 e P2 com valores múltiplos de P3, por forma a extrair uma secção de dados de interesse ou a totalidade do ficheiro. Desta forma, o CC devolve os dados de interesse de uma só vez, reduzindo o número de interações trocadas entre o CC e o TL. Visto que é pretendida a leitura de múltiplos dados do cartão, esta foi a metodologia seguida por forma a não comprometer o desempenho da aplicação, visto que a extração de cada dado individualmente seria um processo moroso.

Após a extração do ND, NIF e NSS por parte do TL, é verificada a sua correta leitura segundo o método de [Teixeira, 2015]. Este método permite verificar não só a correta leitura dos dados pelo TL, como também rejeitar um CC que não contenha os campos ND, NSS e NIF válidos. Estes algoritmos de verificação encontram-se descritos no Anexo A.4.

4.1.2 Morada do Cidadão (0xEF05)

O Ficheiro 0xEF05 contém os dados referentes à morada do cidadão titular. O comando TPDU *Select File* para a seleção deste ficheiro é mostrado na Figura 4.4.

A leitura de dados deste ficheiro realiza-se recorrendo ao comando *Read Binary* do mesmo modo ao descrito na Subsecção 4.1.1.1 para o ficheiro 0xEF02. No entanto, para tornar possível o acesso à informação contida neste ficheiro é necessário transpor os mecanismos de proteção do ficheiro através do envio do PIN de Morada. Por defeito, o valor do PIN de Morada será '0000', caso não tenha sido posteriormente alterado pelo titular.

4.1.2.1 Verificação do PIN de Morada

O processo de envio do PIN de Morada do TL ao CC é assegurado pelo comando TPDU *Verify*, descrito na Subsecção 3.4.6 com os parâmetros ilustrados na Figura 4.5. De notar nos parâmetros da Figura 4.5 que o Byte de Parâmetro P2 possui o identificador do PIN de Morada P2=0x83 e o Byte de Parâmetro P3 possui a dimensão do Corpo de Dados do comando, P3=0x08. Embora o PIN de Morada seja apenas constituído por 4 dígitos, a

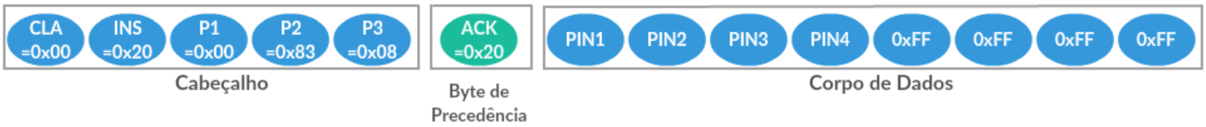

Figura 4.5: Verificação do PIN de Morada através do comando *Verify*.

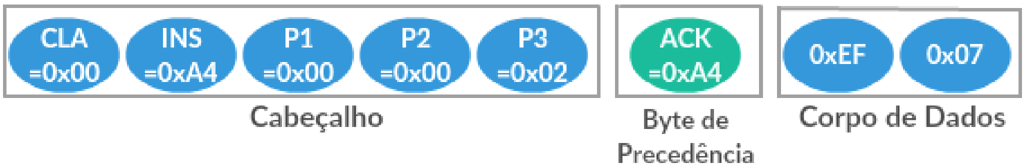

Figura 4.6: Seleção do Ficheiro 0xEF07.

aplicação de validação do PIN alojada no CC espera a receção de 8 caracteres e por isso torna-se mandatório o envio do mesmo número de caracteres no Campo de Dados. O PIN corresponde aos quatro primeiros caracteres enviados, e apenas é enviado após a resposta do CC com o Byte de Precedência (ver Figura 3.5). Os restantes bytes do Campo de Dados deverão possuir valor 0xFF.

Em caso de sucesso, o cartão remeterá uma resposta onde são enviados os Bytes de Estado SW1=0x90 e SW2=0x00.

4.1.3 Memória Livre (0xEF07)

Quando o CC é entregue ao seu titular, o ficheiro 0xEF07 não contém dados gravados. Em vez disso, este ficheiro destina-se à introdução de texto. A seleção deste ficheiro para manipulação é feita pelo Comando *Select File* com os parâmetros declarados na Figura 4.6.

Ao contrário do Ficheiro 0xEF05, o Ficheiro 0xEF07 não contém nenhum mecanismo de proteção contra a leitura não autorizada de dados. Deste modo, para ler o conteúdo deste ficheiro apenas é necessário aplicar o comando *Read Binary* após a seleção do ficheiro, do mesmo modo ao descrito na Subsecção 4.1.1.1 para o ficheiro 0xEF02.

Apesar da leitura do ficheiro ser livre, o mesmo não acontece com escrita de dados. O ficheiro 0xEF07 encontra-se protegido contra a escrita não autorizada de dados. Para escrever neste ficheiro é necessária autenticação do titular do cartão com o PIN de Autenticação fornecido na entrega do CC.

4.1.3.1 Verificação do PIN de Autenticação

O processo de verificação do PIN de Autenticação é assegurado pelo comando TPDU *Verify*, descrito na Subsecção 3.4.6, com os parâmetros exibidos na Figura 4.7. De notar nos parâmetros desta figura que o Byte de Parâmetro P2 possui o identificador do PIN de Autenticação P2=0x81 e o Byte de Parâmetro possui o tamanho do PIN, P3=0x08. O envio do PIN corresponde aos quatro primeiros caracteres enviados no Campo de Dados e apenas

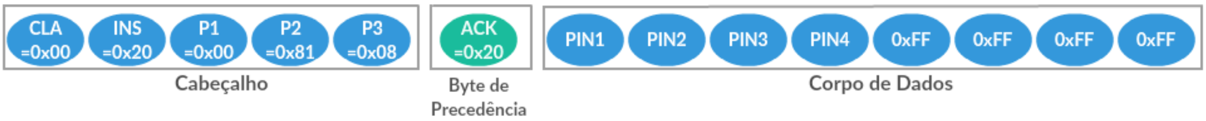

Figura 4.7: Verificação do PIN de Autenticação através do comando *Verify*.

é enviado após a resposta do CC com o Byte de Precedência (ver Figura 3.5). Os restantes bytes do Campo de Dados deverão possuir valor 0xFF.

Em caso de sucesso, o cartão remeterá uma resposta onde são enviados os Bytes de Estado SW1=0x90 e SW2=0x00.

4.1.3.2 Escrita de Dados

Validada a autenticação através da verificação do PIN de Autenticação, a escrita de dados realiza-se recorrendo ao comando TPDU *Update Binary* descrito na Subsecção 3.4.4. De notar que, cada comando *Update Binary* enviado ao cartão apenas permite a escrita até 256 caracteres (P3=0x00), sendo necessária a repetição do comando *Update Binary* até perfazer o número de caracteres desejados para escrita, onde a posição de escrita dada pelos bytes P1 e P2 deverá ser ajustada adequadamente. Em caso de sucesso, o cartão remeterá uma resposta onde são enviados os Bytes de Estado SW1=0x90 e SW2=0x00.

4.2 Comunicação USB

A informação extraída do CC só toma significado quando visualizada pelo utilizador. Nesse sentido foi necessário implementar um mecanismo de transferência de dados entre o TL projetado e um PC. A troca de informação entre os dois pontos de comunicação é realizada através de uma comunicação série USB. Para evitar a instalação de eventuais drivers adicionais no PC do utilizador (conceito "*Plug and Play*") foi utilizada a classe HID na implementação desta comunicação. A classe USB-HID [Bergman *et al.*, 2001] promove mecanismos de configuração e gerenciamento de um dispositivo genérico e é compatível com a generalidade de SOs atuais. No caso desta dissertação, os mecanismos de comunicação foram testados em ambiente Windows©, que recorre a uma API embutida no SO para assegurar as solicitações USB, evitando assim a instalação de drivers adicionais. No entanto, a largura de banda de uma implementação HID é limitada a 64 KB/sec, porém suficiente para o propósito deste projeto.

Todo o dispositivo USB é constituído por um ID de Vendedor (VID) e um ID de Produto (PID). O SO de um PC faz uso desta combinação VID e PID para identificar o dispositivo. Para o propósito experimental deste projeto, foi utilizado o VID=0x03EB e PID=0x204F, disponibilizado pela *Atmel*. Realizada a identificação do dispositivo, a configuração e estabelecimento da comunicação é realizado através de um processo denominado por enumeração,

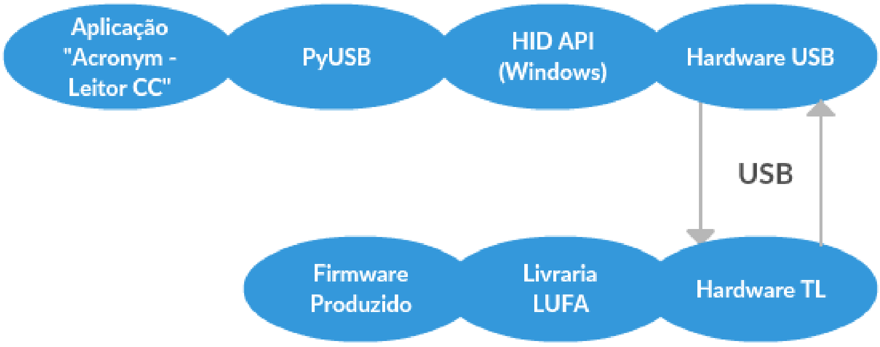

Figura 4.8: Diagrama de mecanismos envolvidos na comunicação USB-HID.

no qual é solicitado um conjunto de descritores do dispositivo que permitem reconhecer e caraterizar a comunicação entre os dois pontos. Este conjunto de descritores é composto por um Descritor de Dispositivo, um Descritor de Configuração, um Descritor de Interface, um Descritor do Ponto Terminal (*endpoint*), um Descritor HID e um Descritor do Relatório HID. É com base nestes descritores que o SO de um PC atribui uma configuração ao dispositivo por forma a habilitá-lo no sistema e são configurados os relatórios de dados trocados entre os pontos terminais.

O trabalho mais exaustivo na implementação da comunicação USB passou por determinar a configuração correta de todos os parâmetros dos descritores envolvidos. A implementação da classe USB-HID foi realizada com auxílio da biblioteca *open source* LUFA [Camera, 2013] e a documentação existente relativa à classe HID [Bergman *et al.*, 2001]. A biblioteca LUFA[3] é uma livraria para a implementação de mecanismos de comunicação série em micro-controladores AVR que possuam módulo USB. Esta biblioteca também assegura um conjunto de funções para suportar solicitações USB. A configuração utilizada em cada descritor especificado encontra-se disponível para consulta no Anexo A.5.

Com o dispositivo reconhecido e devidamente enumerado, uma aplicação no sistema *host* (PC) deverá aceder ao dispositivo através do seu par VID/PID por forma a interagir com o TL criado. Esta aplicação encontra-se descrita na subsecção seguinte. A Figura 4.8 retrata o posicionamento das partes envolvidas no processo de comunicação USB.

Estabelecidos os parâmetros básicos da comunicação USB, o próximo passo passou pela especificação de um protocolo de comunicação para a transferência de dados entre os dois pontos de comunicação. O protocolo estabelecido define um conjunto de mensagens de 8 bytes que governam pedidos e/ou informações sobre o estado atual do processo. Esta troca de mensagens, implementadas no TL e na aplicação de interface com o utilizador, permitem o desencadeamento de ações a elas associadas, nomeadamente a solicitação de dados ao cartão, execução de comandos e processos, o envio e receção de dados específicos, informação de erros no processo, sincronização da comunicação, etc. As mensagens essenciais ao processo

[3] A documentação existente relativa à biblioteca LUFA está disponível em http://www.fourwalledcubicle.com/files/LUFA/Doc/130901/html/ .

de comunicação e seu significado correspondente encontram-se listadas no Anexo A.6.

4.3 Aplicação Produzida

A aplicação denominada por "Acronym - Leitor CC" foi produzida em ambiente Windows©e tem como objetivo a visualização do resultado das interações com o CC. O conceito da aplicação assenta no princípio da comunicação USB-HID, no qual é responsável pela troca de informação entre um PC e o TL.

A aplicação recorre a funções DLL (*Dynamic-Link Library*) em Windows, responsáveis pelo acesso e comunicação com dispositivos USB-HID. Com recurso a estas funções, a aplicação produzida começa por procurar o TL (conectado ao PC via USB) com par VID/PID especifico, e caso o encontre, cria um porto de comunicação com o dispositivo discriminado. Após este ponto, a aplicação troca informação com o TL através de relatórios USB definidos pelo descritor de relatórios HID delineado para o dispositivo (ver Anexo A.5.6). Por forma a manter a aplicação responsiva, a interface da aplicação e os mecanismos de comunicação USB são executadas em *threads* distintas.

Ao introduzir corretamente um cartão válido na interface de leitura do TL, a aplicação produzida é informada da introdução de um cartão no TL e solícita-lhe o envio de comandos TPDU com vista a obtenção de dados provenientes do CC. Deste ponto adiante, é possível a interação com o CC através dos mecanismos implementados.

A aplicação concebida foi trabalhada em linguagem *Python*™, onde foi utilizado o módulo *PyUSB* para acesso às funcionalidades USB em Windows. O ambiente gráfico foi criado com recurso à ferramenta informática *Qt Designer*©. Por forma a simplificar o processo de distribuição da aplicação produzida foram criados ficheiros de instalação executáveis para máquinas de 32 bits e 64 bits.

A Figura 4.9 ilustra a interface inicial da aplicação produzida. Uma descrição mais detalhada sobre a interface da aplicação é presente no Anexo A.7.

4.4 Protótipo Criado

Como prova de conceito desta dissertação, foi produzido *Hardware* tendo em vista a implementação dos pressupostos descritos anteriormente. O esquema de ligações formulado para este fim pode ser consultado no Anexo A.8.1, onde se expõe uma breve descrição.

Dos mecanismos principais do *Hardware* produzido destacam-se a Slot de Leitura (SL) e o micro-controlador responsável pela comunicação entre o CC e o PC. O TL criado neste projeto teve por base o micro-controlador ATMEGA16U2 da *Atmel* [ATMEL, 2010]. A escolha deste micro-controlador de 8 bits prendeu-se essencialmente pelo seu baixo custo de mercado e pelo módulo de transferência de dados USB 2.0 integrado, que facilitou a implementação da troca de informação entre o terminal e o PC. Também não foi alheio o

Figura 4.9: Interface inicial da aplicação produzida.

facto do micro-controlador escolhido ser utilizado na plataforma *open source Arduino™Uno Rev3*. Na verdade, esta plataforma auxiliou o estudo do protótipo e implementações iniciais até ser concebido um produto final.

Para estabelecer comunicação com o CC é necessário a introdução do mesmo numa SL. Esta SL deverá seguir as especificações da norma ISO 7816 parte 2 [Iso.org, 2013a], referente às dimensões e localização dos contactos elétricos. A SL escolhida para este projeto foi o modelo 7431E0225S01 do fabricante FCI©[FCI, 2005]. Para além do correto alinhamento com os pins de contacto do CC, esta SL faculta um mecanismo de deteção de inserção de um cartão na slot através de um *switch* mecânico.

Estabelecido o esquema de ligações entre os componentes fundamentais do TL, foi desenhada uma Placa de Circuito Impresso (PCB) com o auxilio da ferramenta informática *Eagle*©. As dimensões e formato da PCB foi restrita por forma a ser embutida numa caixa plástica fornecida pela *Acronym*. As dimensões e formato da placa PCB produzida encontra-se ilustrada na Figura A.16, no Anexo A.8.2. A PCB concebida possui duas camadas em cobre separadas por isolante FR-4 com espessura de 0,6 mm, onde a espessura das ligações é de 0,5 mm e a espessura das vias de 0,7 mm. Os ficheiros *Gerber* enviados para produção desta placa podem ser consultados no Anexo A.8.3.

Figura 4.10: TL concebido nesta dissertação.

A Figura 4.10 exibe o resultado do produto final no âmbito desta dissertação.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Sobre as Impressões Digitais

Um dos objetivos complementares definidos no início deste projeto era a extração da imagem biométrica de uma impressão digital armazenada no CC. No entanto, ao contrário do que seria expetável, o mecanismo de validação de impressões digitais contido no CC - *Match On Card* - recolhe uma impressão digital fornecida ao CC e através do resultado da comparação entre a impressão digital fornecida e as impressões digitais armazenadas em memória, valida ou não, uma dada impressão digital. Deste modo, em ponto algum do processo as impressões digitais coletadas no CC entram em contacto com o mundo exterior, não sendo assim possível a extração deste conteúdo do cartão. Este facto resulta da salvaguarda da privacidade do cidadão, tal como é latente no artigo 14.º da Lei n.º 7/2007 de 5 de Fevereiro parte 4: "A funcionalidade das impressões digitais contida no circuito integrado do cartão de cidadão só pode ser usada por vontade do respetivo titular" e parte 5: "As autoridades judiciárias e as entidades policiais são as únicas entidades que podem obrigar o cidadão, no âmbito das competências que lhes estejam atribuídas, a provar a sua identidade através da funcionalidade das impressões digitais contidas no circuito integrado do cartão de cidadão de que é portador".

Posto este cenário, foi proposto aos responsáveis da *Acronym* a inclusão de um leitor biométrico no *Hardware* produzido, por forma a possibilitar a comparação da impressão digital do cidadão titular com a impressão digital armazenada no cartão, e caso a comparação fosse efetuada com sucesso, seria apresentada a impressão digital sujeita a validação. No entanto, para além da solução citada levantar várias questões do âmbito legal, a mesma foi descartada pela *Acronym* fundamentando com aumento substancial nos custos de projeto que a inclusão de um leitor biométrico acarretaria.

5.2 Conclusão

O leitor de CC, sendo este um documento único, pessoal e intransmissível, vem adicionar uma mais-valia aos projetos já existentes na *Acronym*, nomeadamente em mecanismos de controlo de acessos e validação de introdução de dados. Os estudos aqui apresentados veem habilitar a empresa na integração de TLs adaptados ao CC nos seus projetos, substituindo os vulgares cartões RFID usados para o efeito. Apesar dos projetos existentes com cartões RFID serem mais práticos para o utilizador pela sua capacidade de comunicação sem contacto, estes são menos seguros em comparação com o documento de cidadania português e encarecem o projeto na sua implementação (pois é necessário atribuir um cartão RFID a cada utilizador).

Este projeto foi extremamente aliciante pela possibilidade de superar desafios, em particular a falta de documentação técnica relativa ao CC. A documentação específica existente refere-se à operação em aplicações num leitor tipo *Online* e não ao nível do par comando-resposta TPDU discutido nesta dissertação. Deste modo, devido à complexidade e dimensão das funcionalidades abordadas ao longo da dissertação, foi dedicado grande parte do tempo à pesquisa das funcionalidades de baixo nível de um SC, por forma a particularizar o resultado deste estudo ao CC.

Como prova de conceito, foram realizados vários testes em CCs distintos ao longo da implementação dos vários conceitos expostos. Ao longo deste processo, foram notórias as limitações de memória que o micro-controlador empregue apresentou, limitando o desenvolvimento de eventuais módulos adicionais. Adicionalmente, a biblioteca LUFA utilizada na implementação da comunicação USB com o PC, revelou-se pesada computacionalmente para o micro-controlador utilizado no TL, sendo notório o atraso na comunicação quando enviados grandes blocos de dados, tais como a fotografia do cidadão. No entanto, este facto não se revelou um entrave às exigências da *Acronym* que aceitou a solução desenvolvida, pois este não era um objetivo substancial ao comprimento deste projeto.

De salientar que foram superadas com sucesso todas as metas básicas e suplementares delineadas, com exceção da aquisição de uma impressão digital armazenada no cartão (ver justificação na Subsecção 5.1). Com o tempo disponível, foram também adicionados novos objetivos, os quais foram superados com êxito.

Em resumo, foi confrontada a norma ISO/IEC 7816 com o documento de cidadania português por forma a superar os mecanismos de conexão, ativação e transmissão de dados do CC. O processo de *Sniffing* efetuado ao CC, permitiu o estudo e implementação de mecanismos de comunicação ao nível do par comando-resposta TPDU, assim como tomar conhecimento da estrutura de armazenamento de dados do CC. A investigação realizada permitiu a implementação de mecanismos de leitura e escrita de dados, tendo sido superadas barreias de segurança inerentes aos objetivos propostos. Como prova de conceito da dissertação, foram estudados e implementados mecanismos de comunicação USB, com vista a troca de informação com uma aplicação produzida, instalada num PC. Por fim, foi projetado e concebido um TL do tipo *Offline*, tendo sido produzida a respetiva PCB.

5.3 Trabalhos Futuros

O trabalho aqui descrito não descarta a utilidade de um TL do tipo *Online*. Este tipo de terminais continuam a ser essenciais em aplicações que interajam diretamente com as APIs num SO de um PC que aceda a informações e mecanismos do documento de cidadania português. São exemplos os métodos de introdução de dados automática ou mecanismos de autenticação *online*.

Neste sentido, seria interessante produzir um TL capaz de operar nos dois modos de funcionamento, tipo *Online* e tipo *Offline*. Ou seja, o TL a desenvolver teria um descritor USB duplo que permitiria operar em modo *Offline*, conforme a dissertação desenvolvida, e em modo *Online* dependendo de uma seleção do utilizador. Assim, seria necessário desenvolver mecanismos de comunicação entre o TL produzido e os comandos reconhecidos por uma API disponível (sugere-se PC/SC versão 1.0), e elaborar um descritor USB com classe CCID para que o dispositivo seja corretamente identificado e enumerado pela mesma API.

Superado o objetivo descrito acima, seria interessante a interligação entre o TL produzido e a aplicação oficial do CC. Sabendo que a aplicação recorre à API PC/SC versão 1.0, seria necessário respeitar as normas requeridas pela aplicação, sendo que a norma ISO/IEC 7816 já se encontra implementada.

Apêndice A

Anexos

A.1 Fatores de Ajustamento da Transmissão

As tabelas que se seguem permitem descodificar os valores candidatos para o ajustamento da transmissão de dados, nomeadamente o Fator F (Tabela A.1) e o Fator D (Tabela A.2). Os valores de F e D são obtidos no Caractere de Interface TA1 devolvido na sequência ATR (Secção 2.4.3) e podem ser empregues no comando PPS (Secção 2.5).

Tabela A.1: Fator de Conversão de Rácio de Relógio (F). Fonte: [Iso.org, 2006].

Bits 8 a 5	**F**	F_{CLOCK} **(MHz)**
0000	372	4.0
0001	372	5.0
0010	558	6.0
0011	744	8.0
0100	1116	12.0
0101	1488	16.0
0110	1860	20.0
0111	RFU	-
1000	RFU	-
1001	512	5.0
1010	768	7.5
1011	1024	10.0
1100	1536	15.0
1101	2048	20.0
111X	RFU	-

Tabela A.2: Fator de Ajustamento Bit-Rácio (D). Fonte: [Iso.org, 2006].

Bits 4 a 1	**D**
0000	RFU
0001	1
0010	2
0011	4
0100	8
0101	16
0110	32
0111	64
1000	12
1001	20
101X	RFU
11XX	RFU

A.2 Campos do Comando/Resposta APDU

A.2.1 Lista de Classes de Comando APDU

A Tabela A.3 enumera todas as Classes de Comando APDU definidas na norma ISO/IEC 7816 parte 4. O byte Classe de Comando é utilizado para identificar aplicações e as suas coleções respetivas, em cabeçalho de comando TPDU (ou comando APDU).

Tabela A.3: Conjunto de valores possíveis da Classe de Comando (CLA) e suas aplicações.

CLA	Aplicação
0x0X	Instruções definidas na norma ISO/IEC 7816-4 (ficheiros e aplicações)
0x10 a 0x7F	RFU
0x8X ou 0x9X	Instruções definidas na norma ISO/IEC 7816-4
0xAX	Aplicações proprietárias do fabricante
0xB0 a 0xCF	Instruções definidas na norma ISO/IEC 7816-4
0xD0 a 0xFE	Aplicações proprietárias do fabricante
0xFF	Reservado ao comando PPS

A.2.2 Lista Completa de Instruções APDU

A Tabela A.4 enumera todas as instruções APDU definidas na norma ISO/IEC 7816 parte 4, independentemente do protocolo de comunicação utilizado.

Note-se que os valores de Byte de Instrução '0x6X', '0x9X' e '0xFF' são inválidos. Estes valores de Byte de Instrução são descartados em comprimento do especificado na norma ISO/IEC 7816 parte 3. Segundo esta norma, o recurso a estes valores poderia originar inconsistências na utilização do protocolo de comunicação T=0, visto que poderiam ser confundidos por Bytes de Precedência ACK. Recorde-se que na definição do protocolo T=0 têm-se NULL=0x60, SW1=0x9X ou SW1=0x6X (excepto SW1=0x60), e PPSS=0xFF (ver Subsecção 3.3.2).

Tabela A.4: Lista completa de instruções APDU.

INS	Nome do Comando
0x04	Deactivate File
0x0C	Erase Record(s))
0x0E | 0x0F	Erase Binary
0x10	Perform SCQL operation
0x12	Perform Transaction Operation

0x14	Perform User Operation
0x20 | 0x21	Verify
0x22	Manage Security Environment
0x24	Change Reference Data
0x26	Disable Verification Requirement
0x28	Enable Verification Requirement
0x2A	Perform Security Operation
0x2C	Reset Retry Counter
0x44	Activate File
0x46	Generate Asymmetric Key Pair
0x70	Manage Channel
0x82	External(Mutual) Authenticate
0x84	Get Challenge
0x86 | 0x87	General Authenticate
0x88	Internal Authenticate
0xA0 | 0xA1	Search Binary
0xA2	Search Record
0xA4	Select File
0xB0 | 0xB1	Read Binary
0xB2 | 0xB3	Read Record(s)
0xC0	Get Response
0xC2 | 0xC3	Envelope
0xCA | 0xCB	Get Data
0xD0 | 0xD1	Write Binary
0xD2	Write Record
0xD6 | 0xD7	Update Binary
0xDA | 0xDB	Put Data
0xDC | 0xDD	Update Record
0xE0	Create File
0xE2	Append Record
0xE4	Delete File
0xE6	Terminate DF
0xE8	Terminate EF
0xFE	Terminate Card Usage

A.2.3 Lista Completa de Respostas APDU

A Tabela A.5 enumera todos os Bytes de Estado devolvidos em respostas a comandos APDU para os comandos e protocolos de comunicação definidos na norma ISO/IEC 7816 parte 4. De notar que não são listadas respostas a comandos proprietários.

Tabela A.5: Lista completa de Respostas APDU.

SW1	SW2	Significado	Descrição
0x61	–	**Informação**	**Bytes de resposta disponíveis**
	0xXX	Informação	Comando executado com sucesso; 'XX' bytes de dados disponíveis através do comando "Obter Resposta"
0x62	–	**Aviso**	**Estado da memória não-volátil inalterada**
	0x00	Aviso	Sem informação
	0x01	Aviso	Memória não-volátil inalterada
	0x81	Aviso	Parte dos dados devolvidos podem estar corrompidos
	0x82	Aviso	Fim do ficheiro/registo antes de ler Le/P3 bytes
	0x83	Aviso	Ficheiro selecionado inválido
	0x84	Aviso	Formato de ficheiro não suportado
	0x85	Aviso	Sem motor de bolsa escravizado por R3bc
	0xA2	Aviso	R-MAC incorreto
	0xA4	Aviso	Cartão bloqueado (durante *reset*)
	0xCX	Aviso	Contador com valor 'X'
	0xF1	Aviso	C-MAC incorreto
	0xF3	Aviso	*Reset* interno
	0xF5	Aviso	Agente por defeito bloqueado
	0xF7	Aviso	Titular do cartão bloqueado
	0xF8	Aviso	Porão como agente atual
	0xF9	Aviso	Configuração de Chave CALC não desbloqueada
	0xXX	Aviso	RFU
0x63	–	**Aviso**	**Estado da memória não-volátil alterada**
	0x00	Aviso	Sem informação
	0x81	Aviso	Ficheiro preenchido até ao ultimo byte; Opções de carregar/atualizar não permitidas
	0x82	Aviso	Chave de cartão não suportada
	0x83	Aviso	Chave de terminal não suportada
	0x84	Aviso	Transmissão de texto simples não suportada
	0x85	Aviso	Transmissão segura não suportada
	0x86	Aviso	Memória volátil indisponível
	0x87	Aviso	Memória não-volátil indisponível
	0x88	Aviso	Número chave inválido

	0x89	Aviso	Comprimento da chave incorreto
	0xC0	Aviso	Verificação sem sucesso; Sem tentativas restantes
	0xC1	Aviso	Verificação sem sucesso; Uma tentativa restante
	0xC2	Aviso	Verificação sem sucesso; Duas tentativas restantes
	0xC3	Aviso	Verificação sem sucesso; Três tentativas restantes
	0xXX	Aviso	RFU
0x64	–	**Erro**	**Estado da memória não-volátil inalterada**
	0x00	Erro	Sem informação.
	0x01	Erro	Tempo de execução esgotado
	0xXX	Erro	RFU
0x65	–	**Erro**	**Estado da memória não-volátil alterada**
	0x00	Erro	Sem informação
	0x01	Erro	Erro ao ler ou escrever na memória EEPROM
	0x81	Erro	Falha de memória
	0xXX	Erro	RFU
0x66	–	**Segurança**	
	0x69	Segurança	Encriptação/Desencriptação incorreta
0x67	–	**Erro**	
	0x00	Erro	Comprimento incorreto
0x68	–	**Erro**	**Funções não suportadas na classe especificada**
	0x00	Erro	Sem informação
	0x81	Erro	Canal lógico não suportado
	0x82	Erro	Mensagens seguras não suportadas
	0x83	Erro	Último comando do encadeamento esperado
	0x84	Erro	Comando por encadeamento não suportado
	0xXX	Erro	RFU
0x69	–	**Erro**	**Comando não permitido**
	0x00	Erro	Sem informação
	0x81	Erro	Comando incompatível com a estrutura de ficheiros existente
	0x82	Erro	Condições de segurança não satisfeitas
	0x83	Erro	Método de autenticação bloqueado
	0x84	Erro	Dados referenciados bloqueados
	0x85	Erro	Condições de utilização não satisfeitas
	0x86	Erro	Comando não permitido (nenhum ficheiro EF selecionado)
	0x87	Erro	Objeto esperado de mensagem segura ausente.
	0x88	Erro	Objeto de mensagem segura incorreto
	0x96	Erro	Os dados têm de ser atualizados novamente

	0xF0	Erro	Permissão negada
	0xF1	Erro	Permissão negada (falta de privilégios)
	0xXX	Erro	RFU
0x6A	–	**Erro**	**Parâmetros P1-P2 incorretos**
	0x00	Erro	Sem informação
	0x80	Erro	Parâmetros no Campo de Dados incorretos
	0x81	Erro	Função não suportada
	0x82	Erro	Nenhum ficheiro selecionado
	0x83	Erro	Registo não encontrado
	0x84	Erro	Memória insuficiente no ficheiro ou registo
	0x85	Erro	Parâmetro Lc inconsistente com a estrutura TLV
	0x86	Erro	Parâmetro P1 ou P2 incorreto
	0x87	Erro	Parâmetro Lc inconsistente com P1-P2
	0x88	Erro	Dados referenciados não encontrados
	0x89	Erro	Ficheiro já existente
	0x8A	Erro	Nome de ficheiro DF já existente
	0xF0	Erro	Parâmetro incorreto
	0xXX	Erro	RFU
0x6B	–	**Erro**	
	0x00	Erro	Parâmetro(s) P1-P2 incorreto(s)
	0xXX	Erro	Referência incorreta
0x6C	–	**Erro**	**Parâmetro Le incorreto**
	0x00	Erro	Parâmetro P3 incorreto
	0xXX	Erro	Parâmetro P3 incorreto; Parâmetro correto dado por SW2
0x6D	–	**Erro**	
	0x00	Erro	Instrução inválida ou não suportada
0x6E	–	**Erro**	
	0x00	Erro	Classe inválida ou não suportada
0x6F	–	**Erro**	**Exceção interna**
	0x00	Erro	Execução do comando abortada
	0xFF	Erro	Cartão morto
0x9-	–		
0x90	0x00	Informação	Comando executado com sucesso
	0x04	Aviso	PIN não verificado, 3 ou mais tentativas restantes
	0x08		Chave/ficheiro não encontrado
	0x80	Aviso	Contador de tentativas de desbloqueio atingiu valor nulo.
0x91	0x01		Estado bloqueado

	0x02		Número de transição atingiu o seu limite
0x92	0x0X	Informação	Escrita na memória EEPROM com sucesso após 'X' tentativas
	0x10	Erro	Memória insuficiente; Sem espaço de armazenamento disponível
	0x40	Erro	Escrita na memória EEPROM sem sucesso.
0x93	0x01		Erro de integridade
	0x02		S2 candidato inválido
0x94	0x00	Erro	Nenhum ficheiro EF selecionado
	0x01		Código de intercâmbio candidato não corresponde à moeda de transação
	0x02		Quantia candidata demasiado elevada
	0x02	Erro	Gama de endereços excedida
	0x03		Quantia candidata demasiado baixa
	0x04	Erro	FID, registo ou padrão de comparação não encontrado
	0x05		Problemas encontrados no campo de dados
	0x07		Mau intercâmbio: motor de bolsa não possui *slot* com intercâmbio R3bc
	0x08		Intercâmbio R3bc não suportado no motor de bolsa
	0x08	Erro	Ficheiro selecionado não tem correspondência com o comando
0x95	0x80		Má sequência
0x96	0x81		Escravo não encontrado
0x97	0x00		PIN bloqueado e Contador de tentativas de desbloqueio tem valor 1 ou 2
	0x02		Chaves principais bloqueadas
	0x04		PIN não verificado, 3 ou mais tentativas restantes
	0x84		Chave base
	0x85		Limite excedido - Chave C-MAC
	0x86		Erro SM - Limite excedido - Chave R-MAC
	0x87		Limite excedido - Contador de sequência
	0x88		Limite excedido - Comprimento R-MAC
	0x89		Serviço não disponível
0x98	0x02	Erro	PIN não definido
	0x04	Erro	Condições de acesso não satisfeitas. Falha na autenticação
	0x35	Erro	"Pedido aleatório"ou "Oferta aleatória"não executada
	0x50	Erro	"Incremento"ou "Decremento"não pode ser executado devido ao limite ter sido atingido

0x99	0x00		Uma tentativa de PIN restante
	0x04		PIN não verificado, uma tentativa restante
	0x85		Estado incorreto - Titular do cartão bloqueado
	0x86	Erro	Falta de privilégios
	0x87		PIN não instalado
	0x88		Estado incorreto - Estado R-MAC
0x9A	0x00		Duas tentativas de PIN restantes
	0x04		PIN não verificado, duas tentativas restantes
	0x71		Parâmetro incorreto - Agente AID duplicado
	0x72		Parâmetro incorreto - Tipo de agente duplicado
0x9D	0x05	Erro	Tipo de certificado incorreto
	0x07	Erro	Tamanho da sessão de dados incorreto
	0x08	Erro	Tamanho de ficheiro de registo DIR incorreto
	0x09	Erro	Tamanho de registo FCI incorreto
	0x10	Erro	Memória insuficiente para carregar a aplicação
	0x11	Erro	AID inválido
	0x12	Erro	AID duplicado
	0x13	Erro	Aplicação previamente carregada
	0x14	Erro	Lista do historial da aplicação cheio
	0x15	Erro	Aplicação não aberta
	0x17	Erro	*Offset* inválido
	0x18	Erro	Aplicação já carregada
	0x19	Erro	Certificado inválido
	0x1A	Erro	Assinatura inválida
	0x1B	Erro	KTU inválido
	0x1D	Erro	Controlos MSM não configurados
	0x1E	Erro	Assinatura da aplicação inexistente
	0x1F	Erro	KTU não existe
	0x20	Erro	Aplicação não carregada
	0x21	Erro	Comprimento dos dados do comando de abertura inválido
	0x30	Erro	Parâmetro de verificação de dados incorreto (endereço inicial incorreto)
	0x31	Erro	Parâmetro de verificação de dados incorreto (comprimento incorreto)
	0x32	Erro	Parâmetro de verificação de dados incorreto (verificação de área de memória ilegal)
	0x40	Erro	Controladores MSM inválidos
	0x41	Erro	Controladores MSM já configurados

	0x42	Erro	Controladores MSM configurados com comprimento de dados inferior a 2 bytes
	0x43	Erro	Comprimento de dados do controlador MSM inválido
	0x44	Erro	Controlador MSM excessivo
	0x45	Erro	Verificação dos dados do controlador MSM inválido
	0x50	Erro	Identificador de produção MCD inválido
	0x51	Erro	Identificador de emissor MCD inválido
	0x52	Erro	Data da configuração do controlador MSM inválida
	0x53	Erro	Número MCD inválido
	0x54	Erro	Campo de erro reservado
	0x55	Erro	Campo de erro reservado
	0x56	Erro	Campo de erro reservado
	0x57	Erro	Campo de erro reservado
	0x61	Erro	Número máximo de desbloqueios atingido
	0x62	Erro	Cartão não foi bloqueado
	0x63	Erro	Funções de encriptação não disponíveis
	0x64	Erro	Aplicação não carregada
0x9E	0x00		PIN não instalado
	0x04		Não foi possível verificar o PIN com sucesso, PIN não instalado
0x9F	0x00		PIN bloqueado e contador de tentativas de desbloqueio igual 3
	0x04		Não foi possível verificar o PIN, contador de tentativas de desbloqueio igual 3
	0xXX		Comando executado com sucesso; 'XX' bytes disponíveis através de comando "Obter Resposta"

A.3 Organização e Posição de Dados

A Tabela A.6 enumera a lista de ficheiros contidos no ficheiro ADF 0x604632FF000002 e seu respetivo conteúdo. Sendo a informação alojada em cada ficheiro armazenada numa série de bytes consecutivos, é listada a posição de cada campo de dados no ficheiro.

Tabela A.6: Lista de Ficheiros e dados extraídos da aplicação 0x604632FF000002.

Ficheiro EF	Campo	Posição
0xEF02	Entidade Emissora	0x001
	Número de Documento (ND)	0x009B
	Número de Série	0x00B8
	Data de Emissão	0x00E7
	Versão	0x00D7
	Local de Pedido	0x00FB
	Data de Validade	0x0137
	Apelido(s)	0x014B
	Nome(s)	0x01C3
	Sexo	0x023B
	Nacionalidade	0x023D
	Data de Nascimento	0x0243
	Altura	0x0257
	Filiação	0x02E9, 0x03D9
	NIF	0x0451
	NSS	0x0463
	NUS	0x0479
	MRZ	0x0503
	Fotografia	0x062F
0xEF05	Distrito	0x000B
	Concelho	0x0077
	Freguesia	0x00E7
	Tipo de Via	0x015F
	Designação da via	0x01C3
	Número de porta	0x0303
	Localidade	0x03CB
	ZIP4	0x042F
	ZIP3	0x0437
	Localidade postal	0x043D
0xEF07	Memória Livre	0x0001

A.4 Mecanismos de Validação

A.4.1 Número de Documento (ND)

O CC veio reforçar o controlo na leitura e escrita do seu número de identificação. O antigo número de BI é agora integrado no ND, que integra também, nos três últimos digitos, um Número de Controlo composto por duas letras e um algarismo, que representam o número de emissão do cartão.

Para verificar a validade do ND empregou-se o algoritmo descrito por Ricardo Teixeira em [Teixeira, 2015]:

1. Inversão da ordem do ND (leitura dos caracteres da direita para a esquerda);
2. Atribuição de pesos às letras do Número de Controlo, onde valor A=10, B=11, C=12, ... ,Y=34 e Z=35;
3. Soma de todos os algarismos e pesos contidos nas posições ímpares;
4. Multiplicação por 2 de todos os algarismos e pesos contidos nas posições pares;
5. Subtração de 9 unidades aos valores obtidos no passo 4 superiores a 10;
6. Soma dos valores resultantes do passo 5;
7. Soma dos valores obtidos nos passos 3 e 6;
8. O valor obtido no passo 7 deverá ser valor múltiplo de 10;

   - Caso se verifique esta condição o ND é válido;

   - Caso não se verifique esta condição o ND é inválido.

A.4.2 Número da Segurança Social (NSS)

O NSS é constituído por 11 algarismos. Para validação do número de NSS realiza-se a leitura do número da direita para a esquerda e multiplicam-se os algarismos a partir da segunda posição desta inversão pelos primeiros dez números primos (2, 3, 5, 7, 11, 13, 17, 18, 23 e 29) e adicionam-se os valores obtidos. O resultado desta soma deverá possui o algarismo das unidades igual a '9', caso contrário o NSS não foi corretamente lido ou o cartão apresentado é inválido [Teixeira, 2015]. Deste modo, o algoritmo de validação do NSS é construído pelos seguintes passos:

1. Inversão da ordem do NSS (leitura dos caracteres da direita para a esquerda);
2. Multiplicação dos algarismos desde a segunda posição do resultado do passo 1 pelos primeiros dez números primos consecutivos (2, 3, 5, 7, 11, 13, 17, 18, 23 e 29);

3. Soma dos resultados obtidos no passo 2;

4. O valor obtido no passo 3 deverá possuir o algarismo das unidades igual a 9;

   - Caso se verifique esta condição o NSS é válido;

   - Caso não se verifique esta condição o NSS é inválido.

A.4.3 Número de Identificação Fiscal (NIF)

Para a efetuar verificação o NIF, toma-se o número da direita para à esquerda e ao resultado desta inversão multiplicam-se os algarismos sucessivamente por 1, 2, 3, 4, 5, 6, 7, 8 e 9, adicionando-se os valores obtidos. O resultado desta soma deverá ser um múltiplo de 11.

No entanto, este sistema de verificação pode falhar no caso do último número do NIF original (Algarismo de Controlo) for igual a '0'. O Algarismo de Controlo (AC) é o último digito do NIF e varia de maneira que a aplicação do algoritmo de verificação resulte em um número múltiplo de 11. No entanto, como o resto da divisão de um número por 11 pode variar entre 0 e 10, também o AC deveria tomar estes valores, em vez de limitado entre 0 e 9. Assim, sempre que o último algarismo do NIF for '0' e o resultado da validação não for múltiplo de 11, deverá calcular-se novamente o algoritmo de validação do NIF, mas desta feita com AC com valor 10. Neste procedimento, a nova soma de teste deverá resultar num número múltiplo de '11', caso contrario o NIF não foi corretamente lido ou o cartão apresentado é inválido. Esta fragilidade foi detetada por Jorge Picado, professor da Universidade de Coimbra aquando na verificação do antigo número do BI, que utilizava um algoritmo semelhante [Teixeira, 2015].

Os passos para implementação do algoritmo de validação do NIF são a seguir apresentados:

1. Inversão da ordem do NIF (leitura dos caracteres da direita para a esquerda);

2. Multiplicação sucessiva dos algarismos por 1, 2, 3, 4, 5, 6, 7, 8 e 9;

3. Soma dos resultados obtidos no passo 2;

4. Dividir o resultado obtido no passo 3 por 11;

   - Caso o resto da divisão anterior seja nula, o NIF é valido;

   - Caso o último algarismo do NIF seja '0' (AC), este algarismo deverá tomar peso igual a 10 e deverão ser repetidos todos os passos anteriores;

   Caso não se verifique as condições anteriores o NIF é inválido.

A.5 Descritores USB-HID

Este anexo expõe os descritores da classe HID usados no TL para reconhecimento e estabelecimento da comunicação série USB com um PC.

A.5.1 Descritor de Dispositivo

Este descritor descreve o TL por inteiro. Revela informações sobre a versão USB utilizada, o tamanho máximo dos pacotes de dados trocados no Ponto Terminal, o identificador do dispositivo através dos parâmetros VID e PID, a versão do TL, o número de configurações possíveis, entre outros. A Tabela A.7 elucida os parâmetros utilizados neste descritor.

Tabela A.7: Parâmetros do Descritor de Dispositivo.

Campo	**Tamanho (bytes)**	**Valor**	**Descrição**
bLenght	1	0x12	Tamanho do Descritor em bytes
bDescriptorType	1	0x01	Descritor de Dispositivo
bcdUSB	2	0x0110	USB v1.1
bDeviceClass	1	0x00	Cada interface possui o seu próprio identificador de classe
bDeviceSubClass	1	0x00	-
bDeviceProtocol	1	0x00	-
bMaxPacketSize0	1	0x08	Tamanho máximo do pacote de dados para o Ponto Terminal 0
idVendor	2	0x03EB	VID
idProduct	2	0x204F	PID
bcdDevice	2	0x0001	Versão do dispositivo v0.01
iManufacturer	1	0x01	"Index" do Descritor de Fabricante (opcional)
iProduct	1	0x02	"Index" do Descritor de Produto (opcional)
iSerialNumber	1	0x00	"Index" do Descritor de Número de Série (opcional)
bNumConfigurations	1	0x01	Número de configurações do Dispositivo

A.5.2 Descritor de Configuração

Um dispositivo USB pode conter várias configurações possíveis. No entanto, no âmbito deste projeto apenas foi utilizada uma configuração, exibida na Tabela A.8. O descritor de configuração específica, entre outros, o número de interfaces, o modo de fornecimento de energia elétrica, a corrente máxima consumida, o número de interfaces pertencentes ao dispositivo e o tamanho total de dados dos descritores envolventes.

Tabela A.8: Parâmetros do Descritor de Configuração.

Campo	**Tamanho (bytes)**	**Valor**	**Descrição**
bLenght	1	0x09	Tamanho do Descritor em bytes
bDescriptorType	1	0x02	Descritor de Configuração
wTotalLenght	2	0x0022	Número de bytes da hierarquia de configuração
bNumInterfaces	1	0x01	Número de Interfaces
bConfiguration	1	0x01	Index do Descritor "String"associado à configuração (opcional)
bmAttributes	1	0xC0	Dispositivo auto-alimentado
bMaxPower	1	0x32	Corrente máxima de 100 mA

A.5.3 Descritor de Interface

Como apenas é a especificada uma interface no âmbito deste projeto, apenas existirá um Descritor de Interface. Da informação relevante especificada neste descritor destaca-se o número de Pontos Terminais associados ao descritor (nesta implementação apenas 1), o número de configurações alternativas e classe associada à interface (HID). A atribuição de valores deste descritor encontra-se elucidada na Tabela A.9.

A.5.4 Descritor do Ponto Terminal

O Descritor do Ponto Terminal é utilizado para descrever os Pontos Terminais associados ao descritor de Interface, exceto o terminal *host*. Especifica os dados específicos à transferência de dados entre os dois pontos de comunicação, entre os quais, a direção do relatório de dados (entrada ou saída) o tipo de transferência suportada, o tamanho máximo de dados que o Ponto Terminal suporta, o intervalo da transferência de dados, etc. A Tabela A.10 especifica os parâmetros utilizados na configuração deste descritor.

Tabela A.9: Parâmetros do Descritor de Interface.

Campo	Tamanho (bytes)	Valor	Descrição
bLenght	1	0x09	Tamanho do Descritor em bytes
bDescriptorType	1	0x04	Descritor de Interface
bInterfaceNumber	1	0x00	Identificador da Interface
bAlternateSetting	1	0x00	Identificador da configuração alternativa
bNumEndpoints	1	0x01	Número de Pontos Terminais associados à interface
bInterfaceClass	1	0x03	Classe HID
bInterfaceSubClass	1	0x00	Sem Classe
bInterfaceProtocol	1	0x00	Sem protocolo especificado
iInterface	1	0x00	Index do Descritor *String* associado à interface (opcional)

Tabela A.10: Parâmetros do Descritor do Ponto Terminal.

Campo	Tamanho (bytes)	Valor	Descrição
bLenght	1	0x07	Tamanho do Descritor em bytes
bDescriptorType	1	0x05	Descritor de Interface
bEndpointAddress	1	0x81	Endereço do Ponto Terminal
bmAttributes	1	0x03	Transferência de dados por interrupção
wMaxPacketSize	2	0x0008	Tamanho máximo do pacote de dados
bInterval	1	0x00	Intervalo para transferências de dados

A.5.5 Descritor HID

O Descritor HID especifica o número, tipo e tamanho do Descritor do Relatório HID associado à classe HID. Os dados associados a este descritor que particulariza as normas que regem a classe, são exibidos na Tabela A.11.

Tabela A.11: Parâmetros do Descritor HID.

Campo	Tamanho (bytes)	Valor	Descrição
bLenght	1	0x09	Tamanho do Descritor em bytes
bDescriptorType	1	0x21	Descritor HID
bcdHID	2	0x0111	v1.11
bCountryCode	1	0x16	Portugal
bNumDescriptors	1	0x01	Número de Descritores HID
bDescriptorType	1	0x22	Relatório associado ao descritor
wDescriptorLenght	2	0x0022	Tamanho do relatório em bytes

A.5.6 Descritor do Relatório HID

Os dados de interesse trocados entre o TL e o CC são transferidos através de relatórios HID. No entanto, o formato dos dados trocados entre os dois pontos de comunicação não são especificados pela classe HID. Posto isto, o Descritor do Relatório HID determina o significado, tamanho e gama de dados que um relatório HID pode conter.

A especificação correta deste descritor é essencial para o SO do dispositivo *host* interpretar corretamente os pacotes de dados trocados. A Tabela A.12 lista os parâmetros de utilização neste descritor.

Tabela A.12: Parâmetros do Descritor do Relatório HID.

Item	Dados	Valor
Usage Page	0x06, 0x00, 0xFF	Vendor-Defined 1
Usage	0x09, 0x01	Vendor-Defined 1
Collection	0xA1, 0x01	Application
- Usage	0x09, 0x02	Vendor-Defined 2
- Logical Minimum	0x15, 0x00	0
- Logical Maximum	0x25, 0xFF	-1
- Report Size	0x75, 0x08	8
- Report Count	0x95, 0x08	8
- Input	0x81, 0x02	(Data, Var, Abs, NWrp, Lin, Pref, NNul, Bit)
- Usage	0x09, 0x03	Vendor-Defined 3
- Logical Minimum	0x15, 0x00	0
- Logical Maximum	0x25, 0xFF	-1
- Report Size	0x75, 0x08	8
- Report Count	0x95, 0x08	8
- Output	0x91, 0x02	(Data, Var, Abs, NWrp, Lin, Pref, NNul, NVol, Bit)
End Collection	0xC0	-

A.6 Mensagens USB

Por forma a sincronizar a comunicação USB entre as ações desencadeadas na aplicação produzida e o fluxo de dados do TL, foi elaborado um protocolo baseado na troca de mensagens. Esta troca de mensagens permite notificar ambos os pontos de comunicação sobre o estado de um processo, desencadear ações, auxiliar a execução de processos, enviar e receber dados, sincronizar a comunicação, etc.

A Tabela A.13 exibe o conjunto de mensagens USB de 8 bytes implementadas para comunicação entre o TL e um PC. Nesta tabela é indicada a direção da mensagem USB, que é definida como "Entrada" ou "Saída". Visto que o PC é ponto central da comunicação USB, "Entrada" denota as mensagens que são remetidas do TL ao PC e "Saída" denota das mensagens que são remetidas do PC ao TL.

Tabela A.13: Lista completa de mensagens USB produzidas para comunicação entre o PC e o TL.

Mensagem USB	Direção	Descrição
CISSUING	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Entidade Emissora
DOCNUMRD	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Número de Documento
CCSERIAL	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Número de Série do CC
EMISDATE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Data de Emissão do documento
CVERSION	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Versão do documento
CCISSUER	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Local de Pedido do documento
CEXPDATE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Data de Validade do documento
SURNAMES	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao(s) Apelido(s) do cidadão
MAINNAME	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao(s) Nome(s) principais do cidadão
CPARENTS	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Filiação do cidadão
CSEXINFO	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Sexo do cidadão
NATIONAL	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Nacionalidade do cidadão
BIRTHDAY	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Data de Nascimento do cidadão
CCHEIGHT	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à altura do cidadão
NIFISCAL	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Número de Identificação Fiscal (NIF)
NSSOCIAL	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Número de Segurança Social (NSS)
NUHEALTH	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Número de Utente de Saúde (NUS)

CCMRZONE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao *Machine Readable Zone* (MRZ)
FOTODATA	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Fotografia do cidadão
DISTRICT	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Distrito da Morada do cidadão
MUNICIPL	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Concelho da Morada do cidadão
CVPARISH	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Freguesia da Morada do cidadão
STRTYPE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Tipo de Via da Morada do cidadão
DOORNUMB	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se ao Número de Porta da Morada do cidadão
LOCALITY	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Localidade da Morada do cidadão
ZIP4CODE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se aos primeiros 4 dígitos do Código Postal da Morada do cidadão
ZIP3CODE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se aos últimos 3 dígitos do Código Postal da Morada do cidadão
POSTALOC	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Localidade Postal da Morada do cidadão
READFREE	Entrada	Informa a aplicação que o próximo bloco de dados enviado refere-se à Memória Livre Armazenada no CC
CARDINTO	Entrada	Informa a aplicação que um cartão foi inserido
CARDEXIT	Entrada	Informa a aplicação que um cartão foi removido
CARDFOOL	Entrada	Informa a aplicação que o cartão inserido é inválido
DFFAILED	Entrada	Informa a aplicação que ocorreu um erro na leitura dos dados do cartão
SAVERROR	Entrada	Informa a aplicação que ocorreu um erro ao guardar os dados na memória livre do CC
ERASINGM	Entrada	Informa a aplicação que o processo de limpeza da Memória Livre do CC foi iniciado
DELETEOK	Entrada	Informa a aplicação que o conteúdo da Memória Livre do CC foi apagado

FIELDEND	Entrada	Informa a aplicação que o envio de determinado campo de dados terminou
USBD1END	Entrada	Informa a aplicação que a transmissão de dados terminou
USBD2END	Entrada	Informa a aplicação que a transmissão de dados referentes à morada terminou
USBD3END	Entrada	Informa a aplicação que a transmissão de dados referentes à fotografia terminou
GIVEMOR	Entrada	Informa a aplicação que o TL aguarda novo bloco de informação para ser gravado na Memória Livre do CC
CREMOVED	Entrada	Informa a aplicação que o processo de desativação do cartão foi concluído e é possível remover o cartão com segurança
PINAOKOK	Entrada	Informa a aplicação que o PIN de Autenticação foi corretamente validado
PNA3LEFT	Entrada	Informa a aplicação que o PIN de Autenticação contem 3 tentativas restantes
PNA2LEFT	Entrada	Informa a aplicação que o PIN de Autenticação contem 2 tentativas restantes
PNA1LEFT	Entrada	Informa a aplicação que o PIN de Autenticação contem 1 tentativa restante
PNA0LEFT	Entrada	Informa a aplicação que o PIN de Autenticação não contem tentativas restantes e encontra-se bloqueado
PINMOKOK	Entrada	Informa a aplicação que o PIN de Morada foi corretamente validado
PNM3LEFT	Entrada	Informa a aplicação que o PIN de Morada contem 3 tentativas restantes
PNM2LEFT	Entrada	Informa a aplicação que o PIN de Morada contem 2 tentativas restantes
PNM1LEFT	Entrada	Informa a aplicação que o PIN de Morada contem 1 tentativa restante
PNM0LEFT	Entrada	Informa a aplicação que o PIN de Autenticação não contem tentativas restantes e encontra-se bloqueado
READDATA	Saída	Solicita ao TL informação armazenada no CC
READFOTO	Saída	Solicita ao TL os dados referentes à Fotografia armazenada no CC
REMOVECC	Saída	Solicita ao TL a desativação dos contactos do CC

PINA'XXXX'	Saída	Envia o PIN de Autenticação introduzido pelo utilizador para verificação por parte do CC, onde 'XXXX' denota o PIN introduzido. Caso o PIN introduzido seja válido, o TL inicia o processo de escrita na Memória Livre do CC
PINE'XXXX'	Saída	Envia o PIN de Autenticação introduzido pelo utilizador para verificação por parte do CC, onde 'XXXX' denota o PIN introduzido. Caso o PIN introduzido seja válido, o TL inicia o processo de limpeza da Memória Livre do CC
PINB'XXXX'	Saída	Envia o PIN de Morada introduzido pelo utilizador para verificação por parte do CC, onde 'XXXX' denota o PIN introduzido. Caso o PIN introduzido seja válido, o TL inicia o processo de leitura da Morada do cidadão
SEND'XXXX'	Saída	Envia uma mensagem onde é informado o tamanho dos dados a armazenar na Memória Livre do CC, onde 'XXXX' denota o tamanho dos dados em formato hexadecimal
BLOCKEND	Saída	Mensagem utilizada para informar o TL que todos os blocos de dados a armazenar na Memória Livre do CC foram enviados

A.7 Interface da Aplicação

Este anexo tem como objetivo descrever de forma explícita a interface da aplicação produzida para o PC e denominada por "*Acronym - Leitor CC*", assim como todas as suas janelas e funcionalidades associadas.

A.7.1 Interfaces Principais

Ao executar a aplicação, o utilizador é presenteado com a interface da Figura A.1. Nesta

Figura A.1: Interface inicial da aplicação.

interface ao clicar sobre os logótipos da *Acronym* e da Universidade de Coimbra, o utilizador será redirecionado para a página web da respetiva entidade.

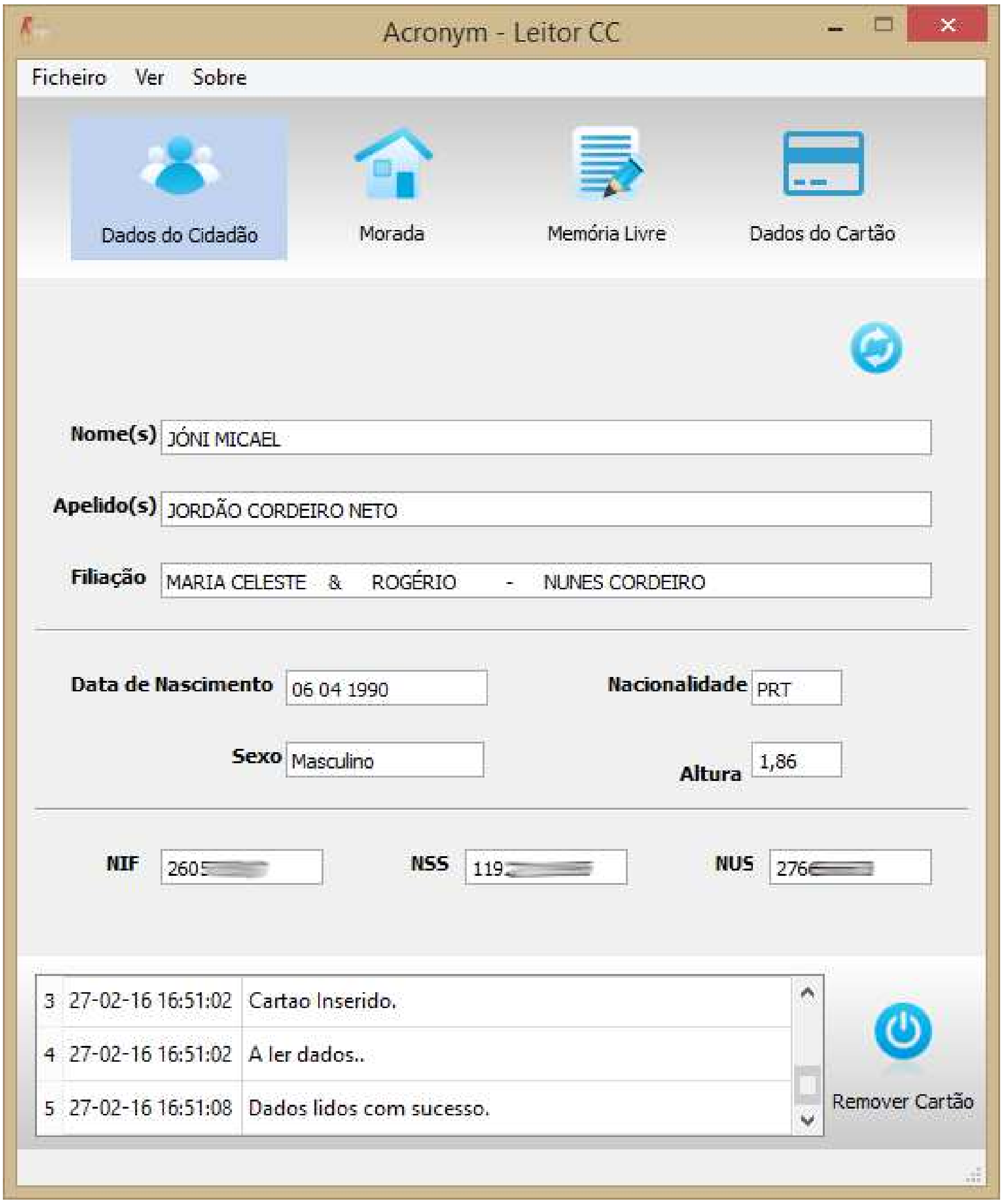

Figura A.2: Interface da janela 'Dados do Cidadão'.

Na Barra de Ferramentas da aplicação são incluídos os botões 'Dados do Cidadão', 'Morada', 'Memória Livre' e 'Dados do Cartão' que permitem alterar a interface apresentada por forma a exibir a informação referente ao cidadão (ver Figura A.2), morada do cidadão (ver Figura A.3), memória livre gravada no cartão (ver Figura A.4) e os dados referentes ao documento (ver Figura A.5), respetivamente. Em cada interface, é apresentado um botão 'Atualizar' com o qual o utilizador pode ordenar uma nova leitura dos dados apresentados e assim renovar a informação apresentada. Note-se que a visualização dos dados de morada do cidadão requer a introdução do PIN de Morada através da janela exibida na Figura A.6.

O menu 'Memória Livre', para além de permitir a visualização do conteúdo de memória livre gravado no CC, permite a edição dos mesmos dados. Para tal, é possível a edição

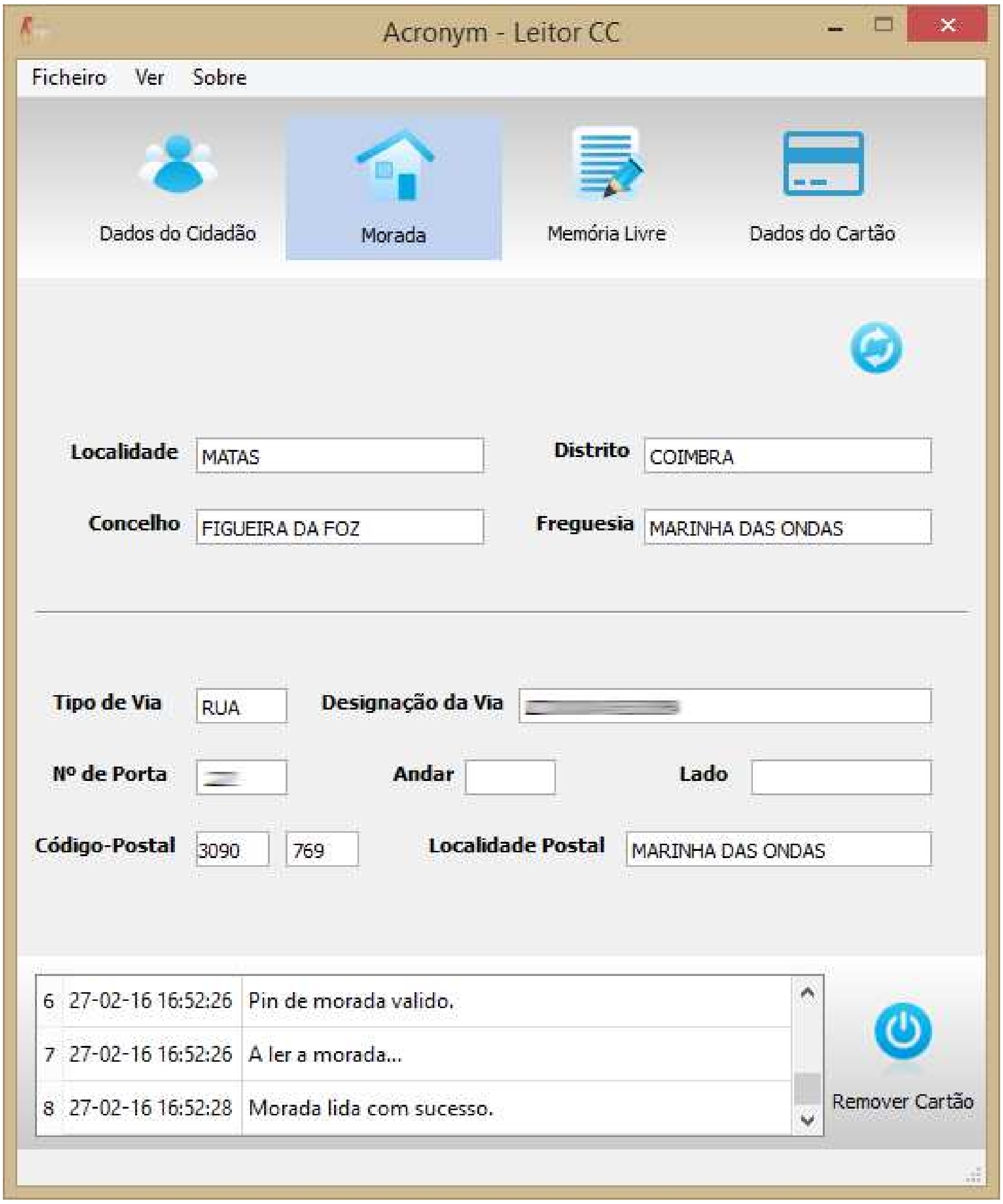

Figura A.3: Interface da janela 'Morada'.

de texto na caixa de texto apresentada na interface, e são dispostos os botões 'Gravar' e 'Apagar', que permitem gravar as alterações efetuadas na memória do CC e apagar o conteúdo da memória livre, respetivamente. Estes processos obrigarão à introdução do PIN de Autenticação através da janela disposta na Figura A.7.

A.7.2 Barra de Menu

A aplicação produzida apresenta ao utilizador uma Barra de Menus no seu topo. Esta barra é constituída pelos menus e submenus exibidos:

- **Ficheiro**

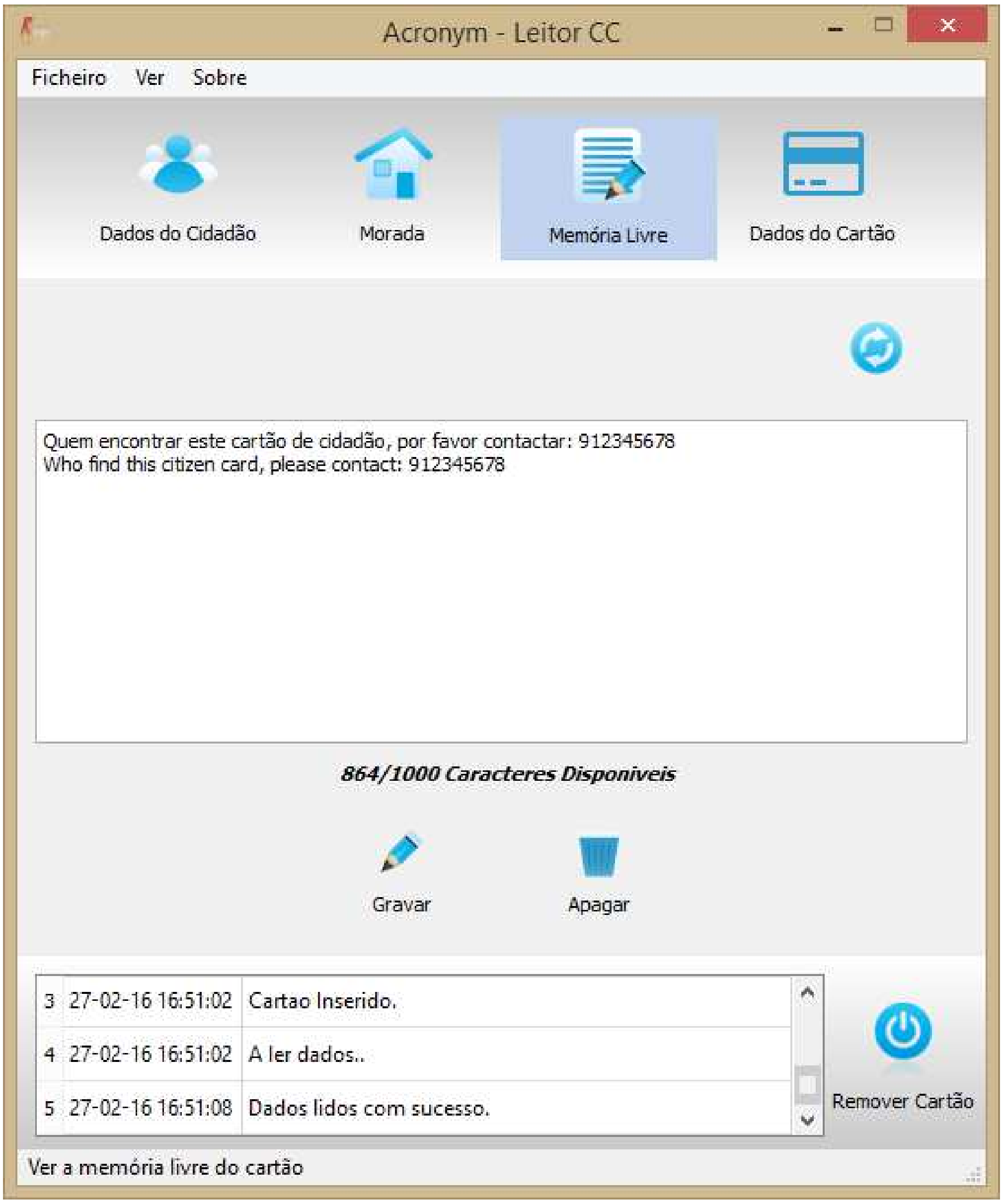

Figura A.4: Interface da janela 'Memória Livre'.

- Atualizar Dados: Solicita ao TL nova leitura dos dados contidos no CC (também disponível com o atalho Ctrl+D);

- Atualizar Morada: Solicita ao TL uma nova leitura dos dados referentes à morada do cidadão. Será solicitado o PIN de Morada (também disponível com o atalho Ctrl+M);

- Gravar Memória: Desencadeia o mecanismo de gravação de dados na Memória Livre do CC (também disponível com o atalho Ctrl+G). Será solicitado o PIN de Autenticação;

- Apagar Memória: Liberta o conteúdo de Memória Livre do CC (também disponível com o atalho Ctrl+E). Será solicitado o PIN de Autenticação;

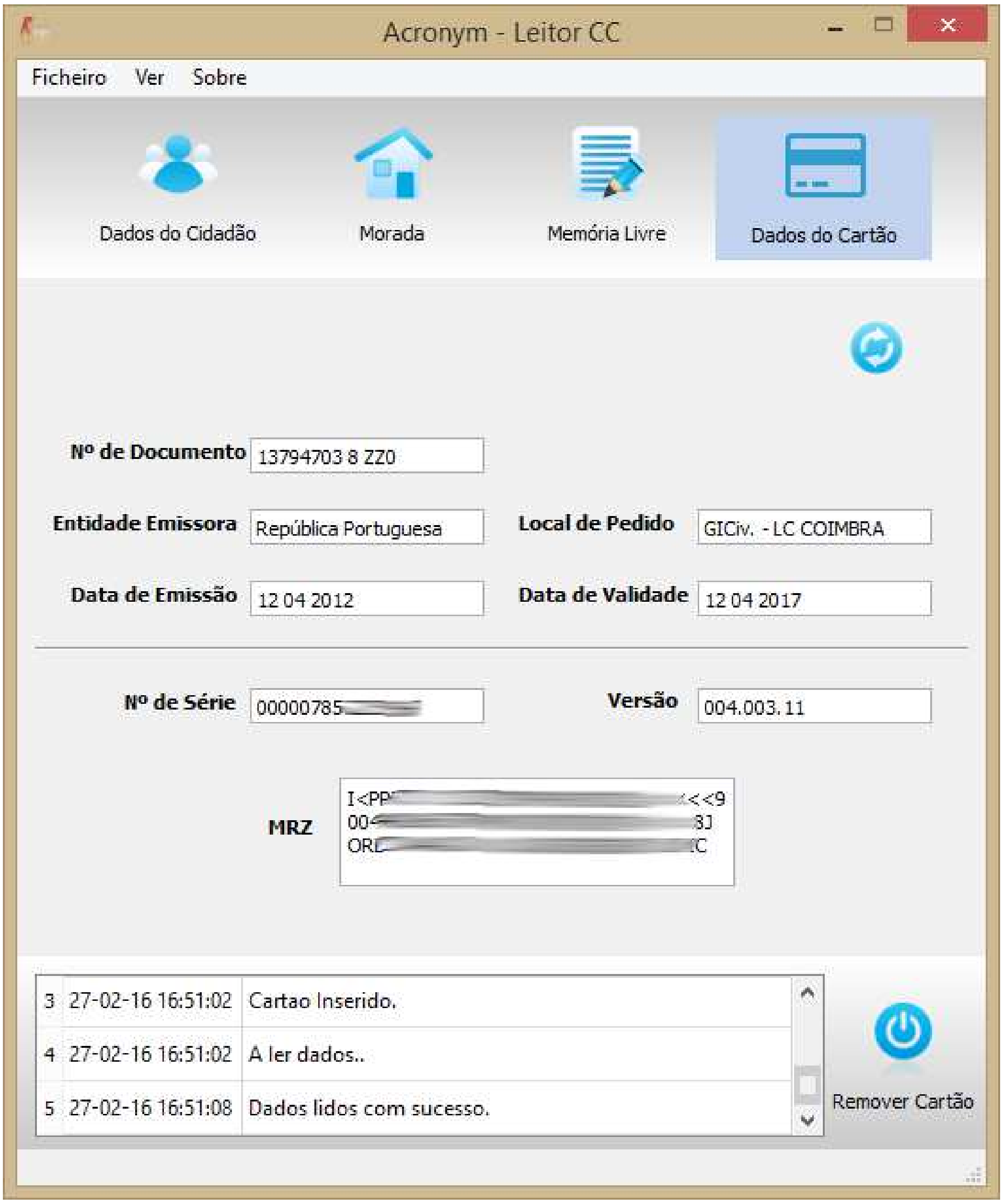

Figura A.5: Interface da janela 'Dados do Cartão'.

- **Remover Cartão:** Inicia o processo de remoção do cartão com segurança (também disponível com o atalho Ctrl+R);

- **Sair:** Permite sair da aplicação (também disponível com o atalho Ctrl+S).

- **Ver**

  - **Dados do Cidadão:** Altera para a interface apresentada na Figura A.2 por forma a exibir a informação referente ao cidadão (também disponível com o atalho Ctrl+1);

  - **Fotografia:** Inicia o processo de leitura da Fotografia do cidadão e exibe o resultado com recurso à aplicação responsável pela visualização de imagens pré-definida

Figura A.6: Janela de introdução do 'PIN de Morada'.

Figura A.7: Janela de introdução do 'PIN de Autenticação'.

no SO (também disponível com o atalho Ctrl+2);

- Morada: Altera para a interface apresentada na Figura A.3 por forma a exibir a informação referente à morada do cidadão (também disponível com o atalho Ctrl+3). Caso o processo de leitura da morada do cidadão não tenha sido previamente desencadeado, serão processados os mecanismos necessários ao processo;

- Memória Livre: Altera para a interface apresentada na Figura A.4 por forma a exibir a informação referente ao conteúdo de Memória Livre do CC (também disponível com o atalho Ctrl+4);

- Dados do Cartão: Altera para a interface apresentada na Figura A.5 por forma a exibir a informação referente ao documento (também disponível com o atalho Ctrl+5).

- **Sobre**

- Aplicação: Exibe as informações gerais sobre a versão e propósito da aplicação exibidas na Figura A.8 (também disponível com o atalho Ctrl+A);

- Acronym: Redireciona o utilizador para a página web da *Acronym* (também disponível com o atalho Ctrl+W);

- FCTUC: Redireciona o utilizador para a página web da FCTUC (também disponível com o atalho Ctrl+U).

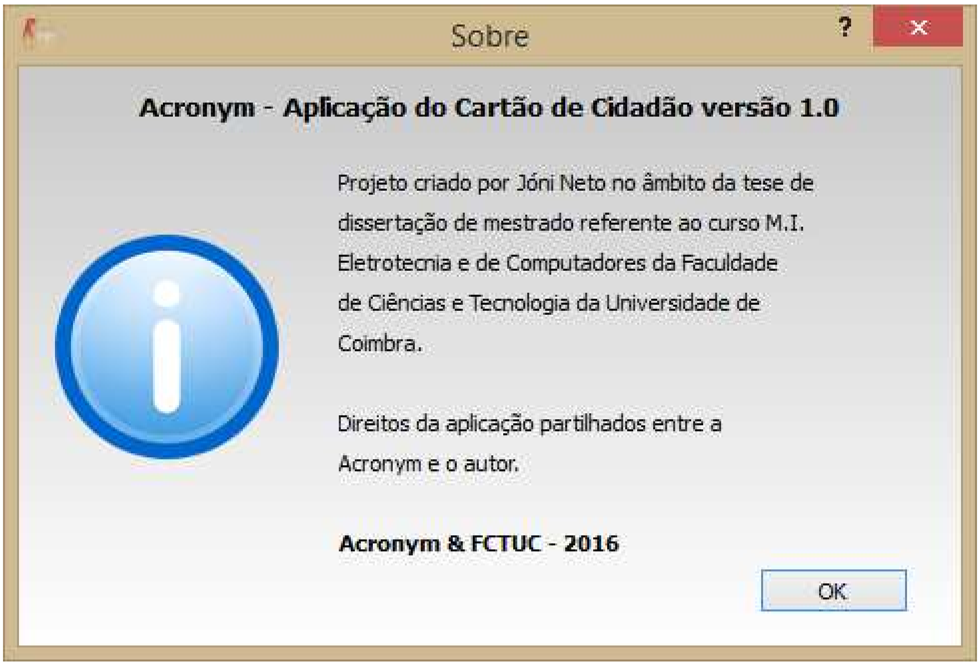

Figura A.8: Janela informativa do âmbito da aplicação.

A.7.3 Janelas de Mensagens

Com o TL conectado e a aplicação em execução, deverá ser introduzido um CC na SL. Ao introduzir um cartão, o TL verifica se o cartão introduzido é válido através da sua cadeia de resposta ATR, e em caso afirmativo, informa a aplicação que um cartão válido foi introduzido e inicia o processo de leitura de dados do CC. Caso contrário, o utilizador será alertado com a janela de aviso exibida na Figura A.9. Adicionalmente, caso exista um erro na leitura de dados, o utilizador será alertado com a janela de erro da Figura A.10. Caso seja tentado algum mecanismo de leitura ou escrita de dados através da aplicação sem que tenha sido inserido um CC válido na SL será exibida a mensagem de aviso da Figura A.11.

É de notar que o TL deverá ser previamente conectado ao PC via USB antes da execução da aplicação. Caso não se verifique, será exibida a mensagem de erro da Figura A.12.

A.7.4 Outras Funcionalidades

No fundo da aplicação é exibida uma caixa com informações sobre o estado do processo. Nesta caixa, denominada por 'Caixa de Estado' é apresentado o número da notificação, data e hora do processo, e uma descrição informativa ao utilizador.

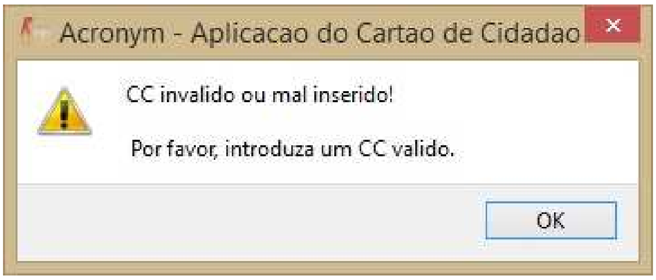

Figura A.9: Janela de aviso de cartão mal inserido ou inválido.

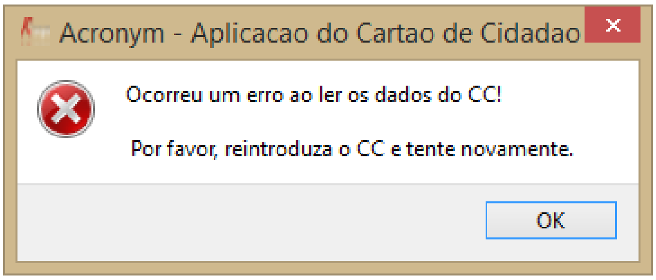

Figura A.10: Janela de erro na leitura dos dados do cartão.

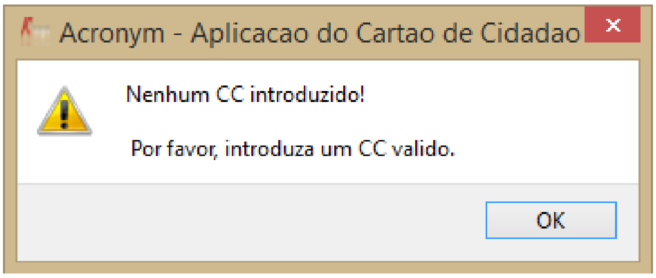

Figura A.11: Mensagem de alerta de nenhum CC introduzido.

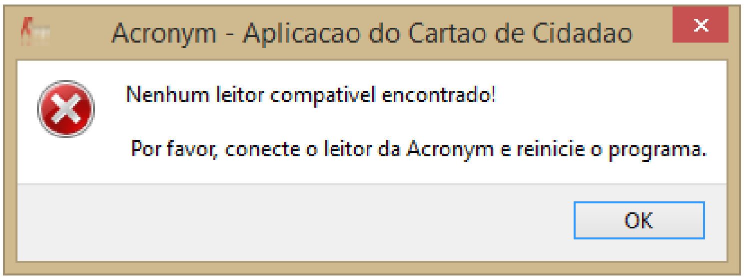

Figura A.12: Mensagem de erro - TL não conectado.

O botão 'Remover Cartão', situado igualmente no fundo da aplicação, desencadeia o mecanismo de desativação do cartão em segurança (ver Secção 2.2). Por forma a proteger eletricamente o CC, este processo ordena ao TL a desativação elétrica dos pinos de contacto do chip do cartão e após a sua conclusão será exibida a mensagem de aviso da Figura A.13. Por forma a assegurar a segurança da informação do utilizador, ao desencadear este processo, todas as informações extraídas do CC na presente sessão serão apagadas da aplicação, incluido a fotografia do cidadão, se extraida.

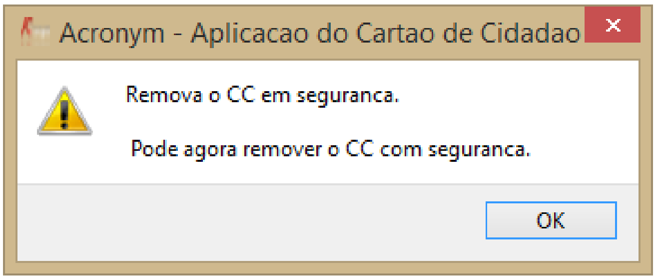

Figura A.13: Janela de aviso que informa que os contactos do cartão foram eletricamente desativados e é possível remover o cartão em segurança.

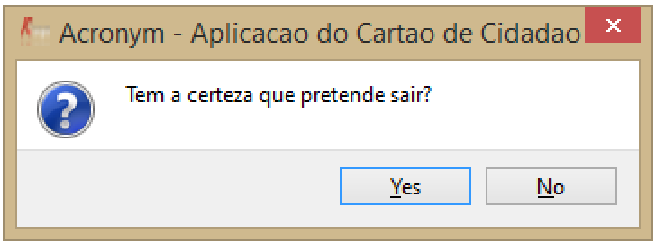

Figura A.14: Janela de confirmação de encerramento da aplicação.

Por fim, caso o utilizador pretenda fechar a aplicação, quer clicando no botão da janela do Windows para o efeito ou através da Barra de Menu 'Ficheiro->Sair', o utilizador será confrontado com a janela de confirmação da Figura A.14.

A.8 Hardware Projetado

A.8.1 Esquema de Ligações

A Figura A.15 exibe o esquema de ligações para o TL concebido para levar a cargo os objetivos deste projeto. Por forma a reduzir a complexidade do projeto, o circuito é alimentado via USB (5V), com uma proteção de corrente de 500 mA.

Este circuito contém o micro-controlador ATMEGA16U2 da *Atmel* [ATMEL, 2010] que assegura os mecanismos de comunicação USB descritos na Subsecção 4.2. O micro-controlador ATMEGA16U2 é ainda responsável por alimentar e estabelecer comunicação com um CC introduzido através da SL 7431E0225S01 do fabricante FCI©[FCI, 2005], onde a ligação com a tensão de alimentação do cartão (VCC) encontra-se estabelecida nos portos PB4 e PB5 do micro-controlador, fornecendo uma tensão de 5V contínuos e uma corrente máxima de 80 mA (40 mA por cada pino). O sinal de relógio CLK é fornecido graças ao módulo *Output Compare* do micro-controlador que é configurado por forma a gerar uma onda quadrada com frequência de 4 MHz. Por sua vez, a troca de dados com a memória do cartão é assegurada através do módulo USART do micro-controlador, nas linhas de comunicação TXD1 e RXD1 (portos PD2 e PD3). De notar que estes pinos encontram-se conectados entre si. Desta forma a comunicação com o CC ocorre em modo *halfduplex* em que apenas uma das linhas de transmissão se encontra ativa em cada instante de tempo, tal como descrito na Subsecção 2.3. Por fim, o sinal de RST é assegurado pelo porto PB6 do micro-controlador.

A SL da FCI possui um *switch* mecânico que é ativo quando um cartão é introduzido. Este mecanismo de deteção é extremamente útil, pois permite a ativação eléctrica dos terminais do cartão apenas quando os seus pinos se encontram corretamente alinhados, evitando eventuais danos no micro-controlador embutido no cartão. O nível lógico deste *switch* mecânico é observado pelo porto PB1 do micro-controlador da *Atmel*.

A placa produzida possui um LED (*Light-Emitting Diode*) bicolor, ligado aos portos PD4 e PD5 do micro-controlador. Este LED deverá ser habilitado quando um CC é introduzido na SL, fornecendo um elemento visual ao estado do processo. O LED deverá possuir cor verde, indicando que o cartão foi corretamente introduzido e que o TL se encontra preparado para enviar e receber comandos do cartão. Por outro lado, quando se encontra a ocorrer um processo de comunicação com o CC, o LED deverá possuir cor vermelha, indicando que o cartão não deverá ser removido até que o processo de comunicação seja concluído.

Por fim, como a placa produzida se trata de uma versão protótipo, foram adicionados Pontos de Teste ao circuito. Estes Pontos de Teste para além de permitirem averiguar o estado do processo em diversas instâncias, são fundamentais para programar o micro-controlador *ATMEGA* após a sua instalação no circuito. Assim, foram adicionados Pontos de Teste às linhas MOSI, MISO, RST, 5V e GND do micro-controlador.

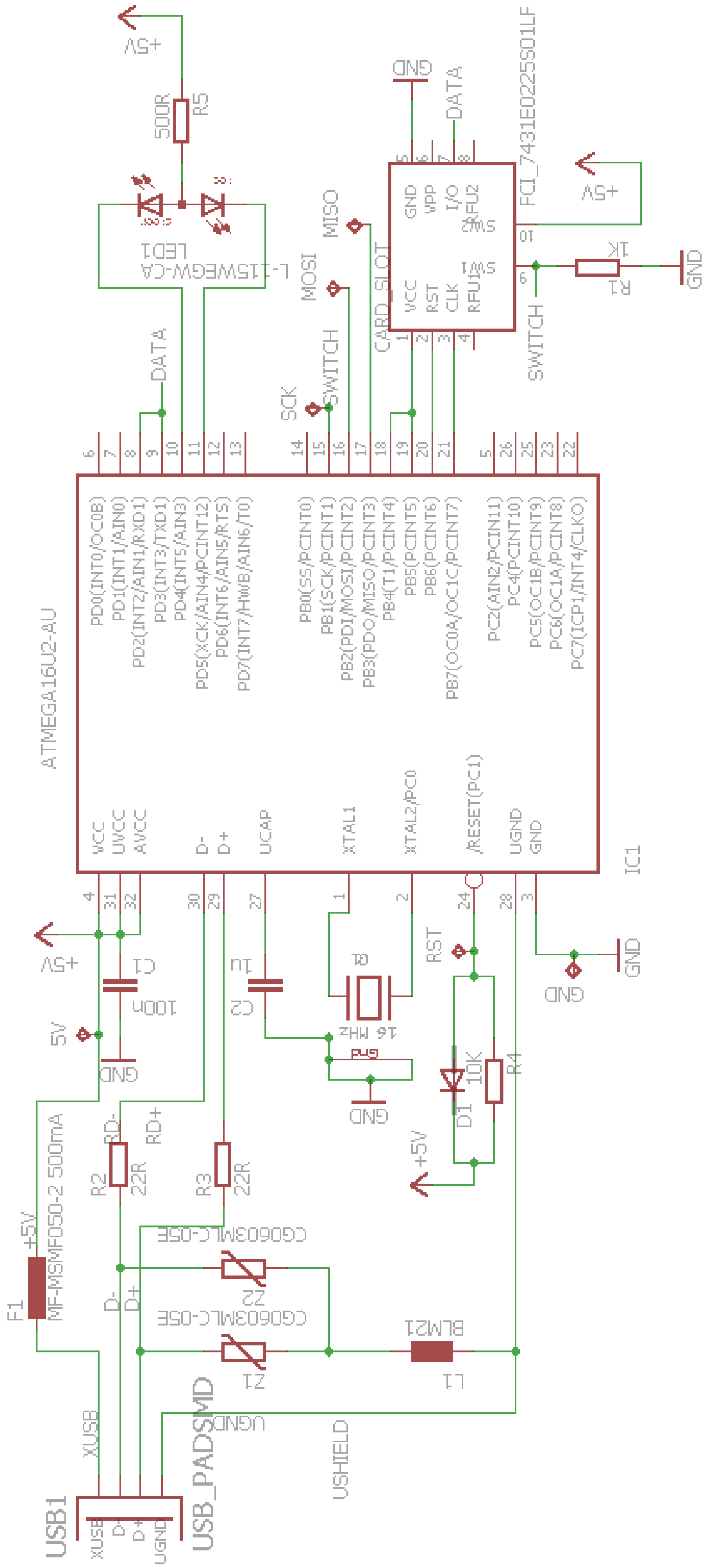

Figura A.15: Esquema de ligações para o TL projetado.

A.8.2 Placa de Circuito Impresso

Estabelecido o esquema de ligações do Anexo A.8.1, foi desenhada a PCB restrita às dimensões da Figura A.16. Esta placa foi criada em ambiente *Eagle*©, do qual resultou do produto da Figura A.17.

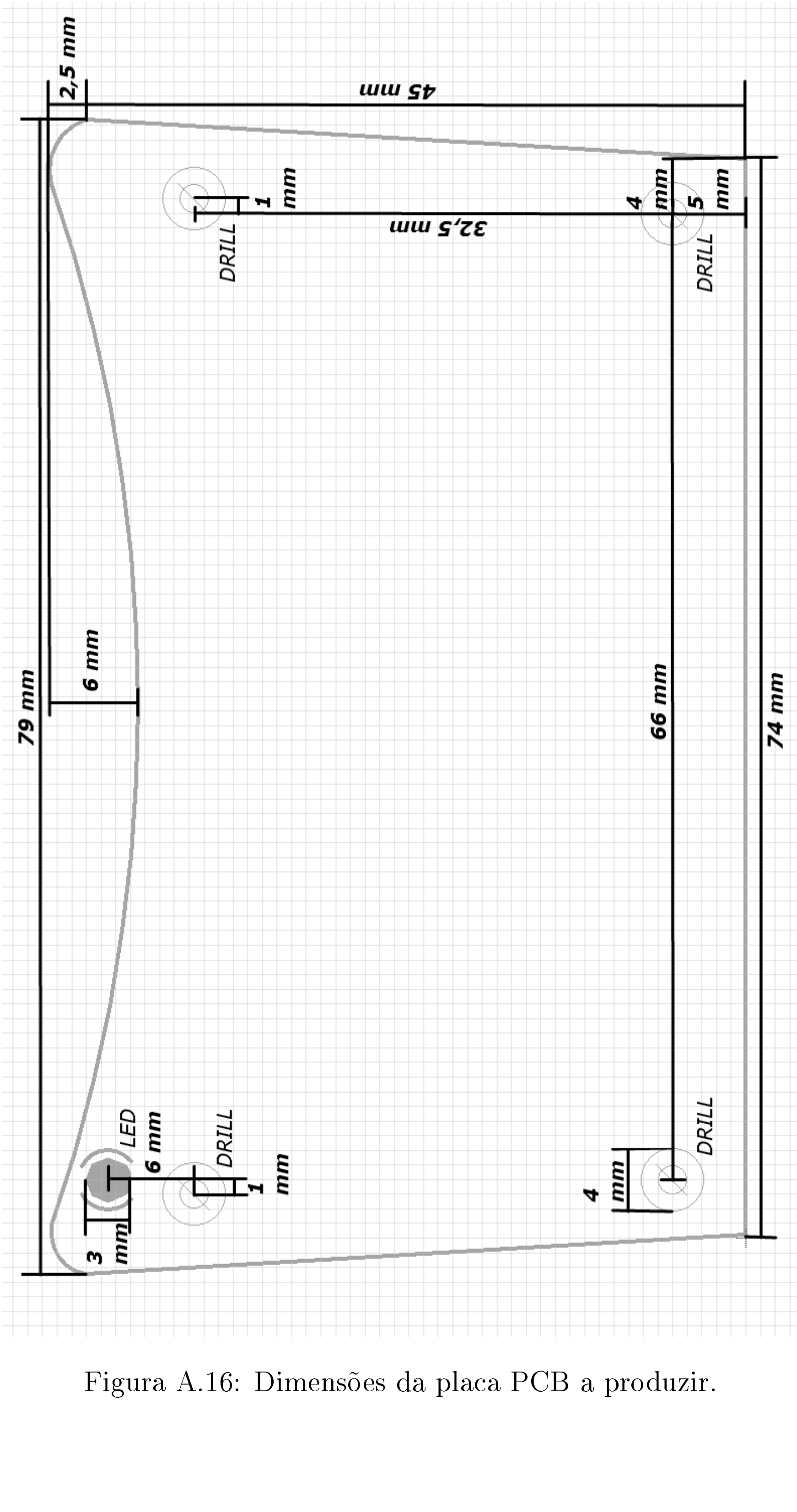

Figura A.16: Dimensões da placa PCB a produzir.

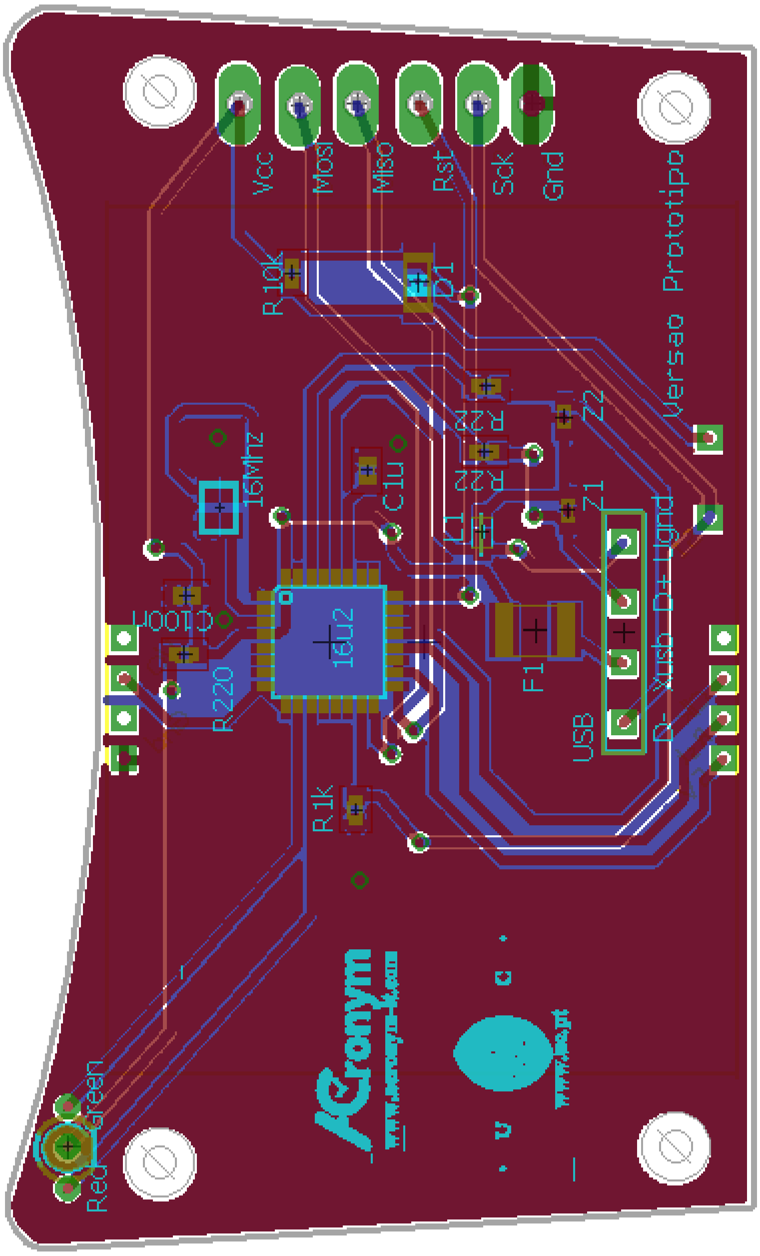

Figura A.17: Pré-visualização da PCB produzida em ambiente *Eagle*©.

A.8.3 Ficheiros Gerber

Delineado o protótipo da PCB do TL analisado nesta dissertação, foram gerados os ficheiros *gerber* necessários para a produção da placa. As figuras seguintes exibem a pré-visualização dos ficheiros .GBL, .GBO, .GBS, .GTL, .GTO, .GTP e .GTS essenciais para a produção da placa projetada.

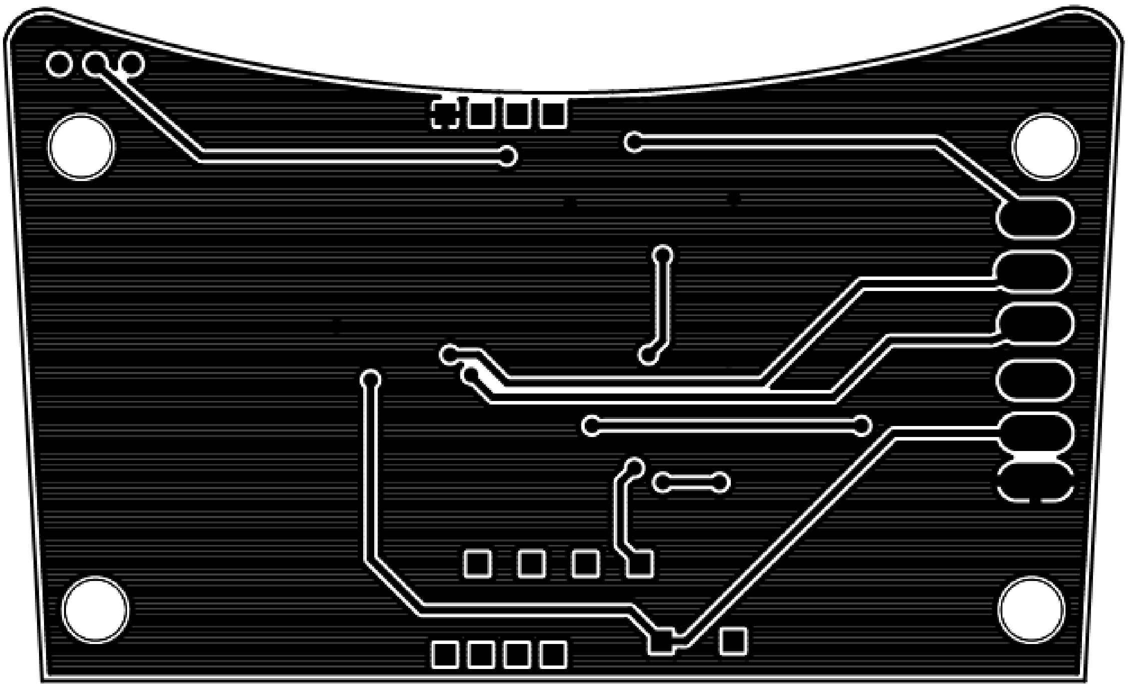

Figura A.18: Ficheiro *gerber* GBL (camada de cobre inferior).

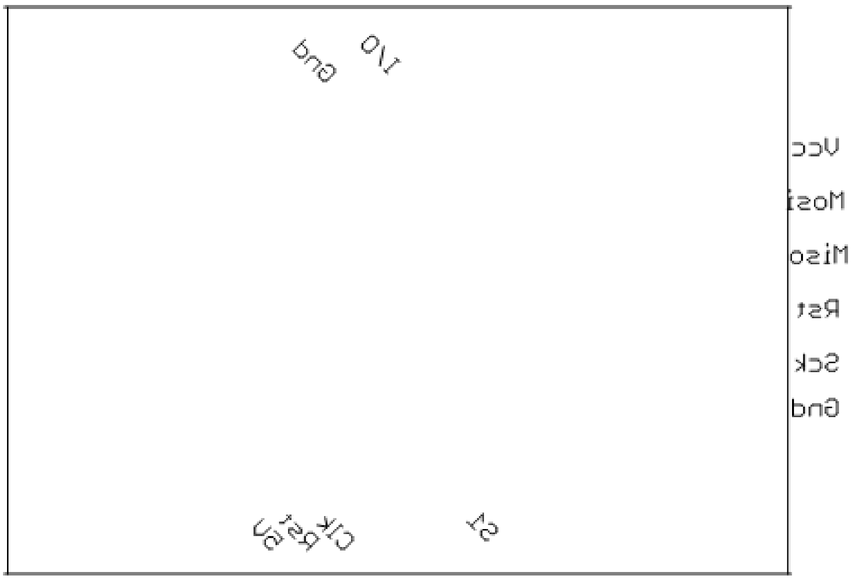

Figura A.19: Ficheiro *gerber* GBO (*silkscreen* inferior).

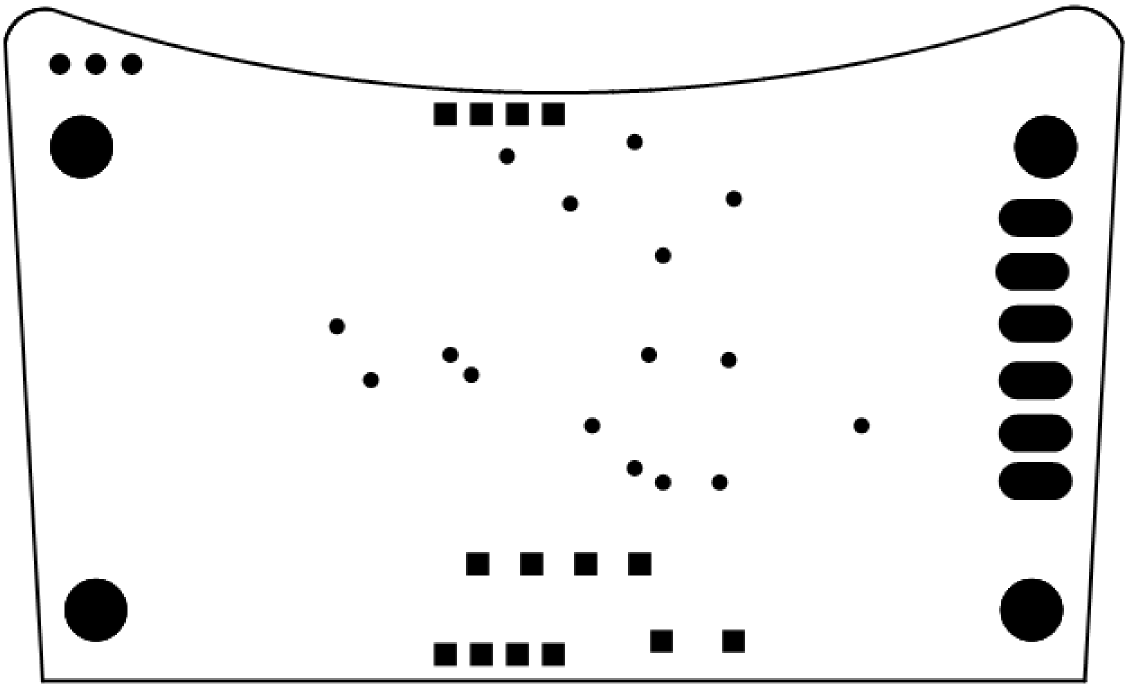

Figura A.20: Ficheiro *gerber* GBS (máscara de solda inferior).

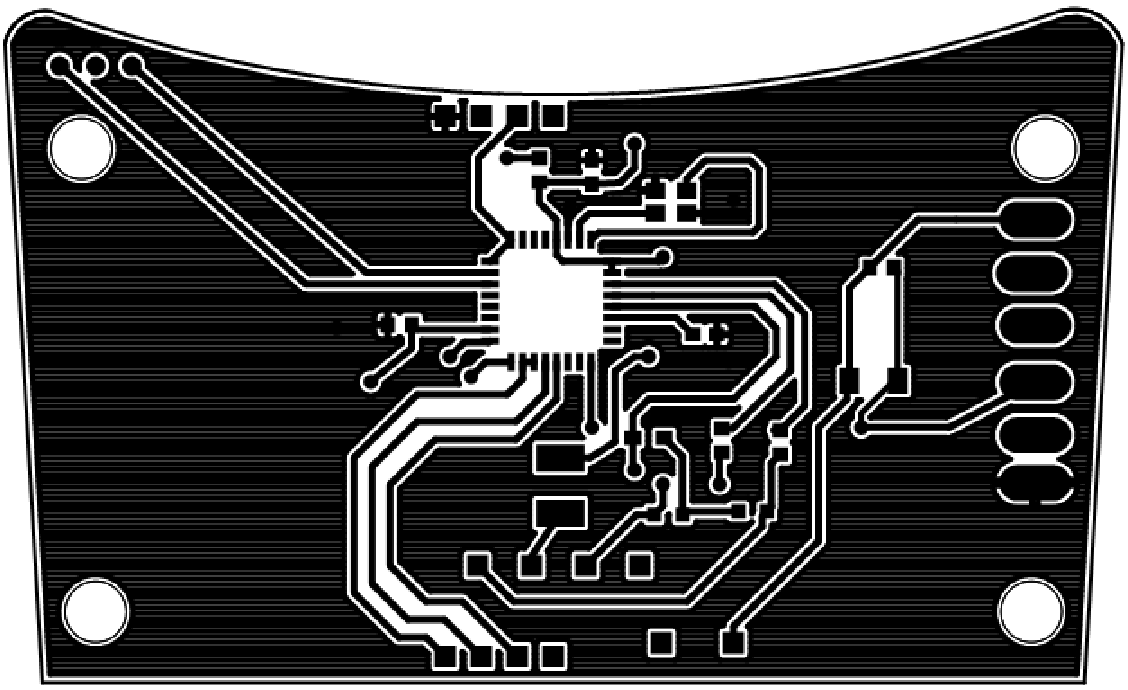

Figura A.21: Ficheiro *gerber* GTL (camada de cobre superior).

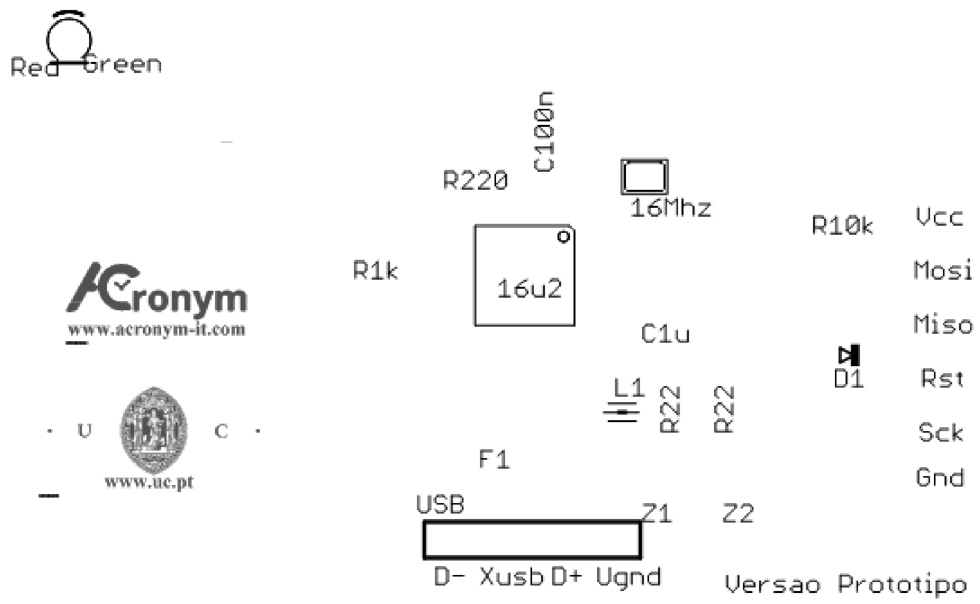

Figura A.22: Ficheiro *gerber* GTO (*silkscreen* superior).

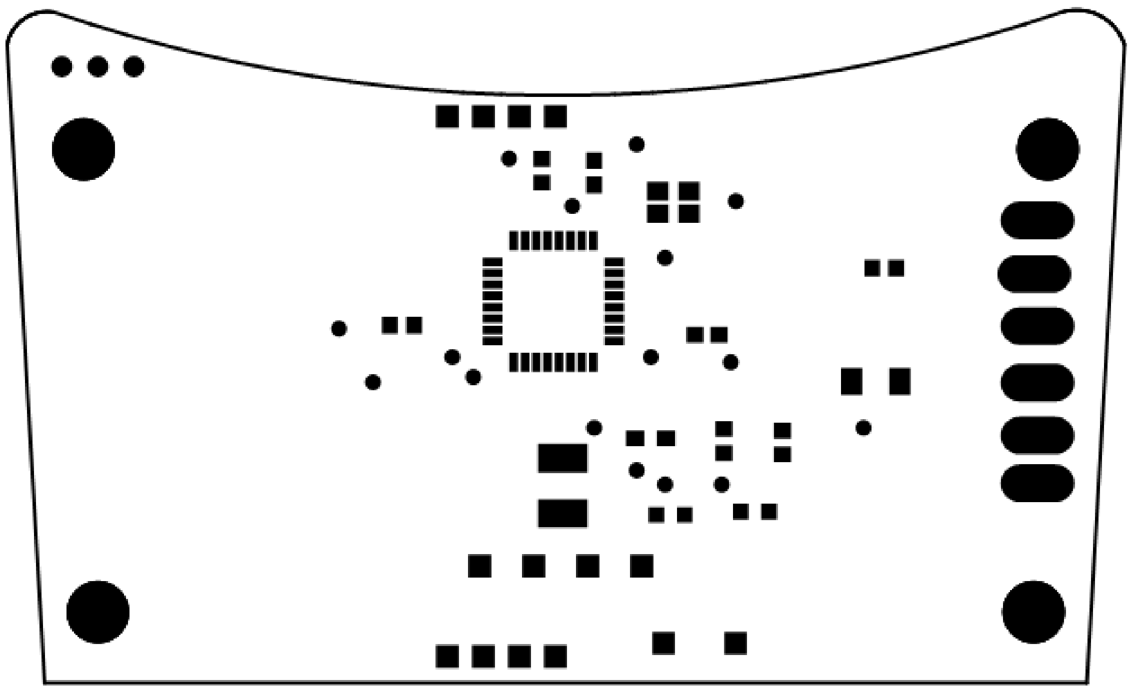

Figura A.23: Ficheiro *gerber* GTS (máscara de solda superior).

Bibliografia

[ATMEL, 2010] ATMEL. *ATmega16U2: 8-bit AVR Microcontroller with 16K Bytes of ISP Flash and USB Controller*, 2010. URL http://www.atmel.com/devices/ATMEGA16U2.aspx. (Citado nas páginas 9, 13, 54, e 97).

[Bergman *et al.*, 2001] Mike Bergman, Tom Peurach, Tom Schmidt, Steve McGowan, Jodi Crowe, Robert Dezmelyk, Remy Zimmwemann, Mike Van Flandern, Bob Nathan, Mike Davis, and Joe Rayhawk. Universal Serial Bus (USB). In: *Device Class Definition for Human Interface Devices (HID).* USB Implementers Forum, June 2001. (Citado nas páginas 52, e 53).

[Camera, 2013] Dean Camera. *Lightweight USB Framework for AVRs*, 2013. URL http://www.lufa-lib.org. (Citado nas páginas 11, 13, e 53).

[Crocker *et al.*, 2010] Paul Crocker, Vasco Nicolau, and Simão Sousa. Sniffing with Portuguese Identify Card for Fun and Profit. In: *Proc. European Conf. on Information Warfare and Security*, ECIWS 2010, pp. 43–56. July 2010. (Citado na página 48).

[CWAV, 2005] CWAV. *USBee AX Test Pod Users Manual*, 2005. URL http://www.usbee.com. (Citado na página 9).

[Everett, 1992] David Everett. *Smart Card Tutorial - Part 1*, September 1992. URL http://www.smartcard.co.uk/tutorials/sct-itsc.pdf. (Citado na página 20).

[FCI, 2005] FCI. *7431E0225S01 Datasheet - FCI connector*, 2005. URL http://pdf1.alldatasheet.com/datasheet-pdf/view/161312/FCI-CONNECTOR/7431E0225S01.html. (Citado nas páginas 10, 55, e 97).

[Ferrari *et al.*, 1998] Jorge Ferrari, Susan Poh, Robert Mackinnon, and Laksman Yatawara. *Smart Cards: A Case Study*, October 1998. URL http://www.redbooks.ibm.com/redbooks/pdfs/sg245239.pdf. (Citado nas páginas 2, e 29).

[Iso.org, 2006] Iso.org. *Part 3: Cards With Contacts - Electrical Interface And Transmission Protocols.* ISO/IEC, 2006. (Citado nas páginas xv, 3, 11, 18, 20, 21, 23, 25, 26, 27, 30, 31, 32, 35, 61, e 62).

[Iso.org, 2011] Iso.org. *Part 1: Cards With Contacts - Physical Characteristics*. ISO/IEC, 2011. (Citado na página 2).

[Iso.org, 2013a] Iso.org. *Part 2: Cards With Contacts - Dimmensions And Location Of The Contacts*. ISO/IEC, 2013a. (Citado nas páginas xv, 3, 10, 17, 20, e 55).

[Iso.org, 2013b] Iso.org. *Part 4: Organization, Security and Commands For Interchange*. ISO/IEC, 2013b. (Citado nas páginas 3, 11, 12, 23, 28, 29, 33, 34, 36, 40, e 41).

[Jurgensen and Guthery, 2002] Timothy Jurgensen and Scott Guthery. *Smart Cards: The Developer's Toolkit*. Prentice Hall PTR, 1st ed., July 2002. ISBN 0130937304. (Citado nas páginas 20, 36, 37, e 39).

[Mesquita, 2010] Tiago Mesquita. *Soluções Single Sign On em Software de Uso Livre*, 2010. URL `https://repositorio-aberto.up.pt/bitstream/10216/59137/1/000142675.pdf`. (Citado na página 5).

[Microsystems, 2011] SCM Microsystems. *SCR335/SCR335v2.0: PC-linked Contact Smart Card Reader*, 2011. URL `http://www.scmmicro.com`. (Citado nas páginas 9, e 47).

[Mota, 2012] Sara Mota. Movensis Cria Solução Que Identifica Clientes Na Venda De Tabaco. *Diário Economico*, p. 26, Julho 2012. (Citado na página 7).

[Qt, 2015] Qt. *Qt Designer*, 2015. URL `http://www.qt.io/developers/`. (Citado na página 12).

[Rankl, 2007] Wolfgang Rankl. *Smart Cards Applications: Design models for using and programming smart cards*. John Wiley & Sons, Ltd, 1st ed., June 2007. ISBN 047005882X. (Citado nas páginas 17, 19, 33, e 34).

[Rankl and Effing, 2010] Wolfgang Rankl and Wolfgang Effing. *Smart Card Handbook*. John Wiley & Sons, Ltd, 4th ed., 2010. ISBN 0470743670. (Citado nas páginas xiii, 2, 6, 25, 26, 28, 35, 37, e 40).

[Teixeira, 2015] Ricardo Teixeira. Matemática No Quotidiano: Do BI Aos Números De Identificação Do Cartão de Cidadão. *Atlântico Expresso*, p. 15, Março 2015. (Citado nas páginas 13, 50, 73, e 74).

[UCMA *et al.*, 2007] UCMA, UMIC, and DGRN. *Cartão de Cidadão: O Novo Documento De Identificação Dos Cidadãos Portugueses [Nota Informativa-1]*, Março 2007. URL `http://www.pofc.qren.pt/ResourcesUser/2013/Concursos_Avisos/AAC01_SAMA_7_GuiaPraticoUtilizacao_CartaoCidadao.pdf`. (Citado nas páginas xiii, 1, 4, e 5).