



**IPG** Politécnico  
|da|Guarda  
Polytechnic  
of Guarda

# RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Pedro Miguel Amaral dos Reis

dezembro | 2015



Instituto Politécnico da Guarda  
Escola Superior de Tecnologia e Gestão

## **iEcoSys**

**Sistema Inteligente de Gestão de Resíduos Sólidos Urbanos**

Pedro Miguel Amaral dos Reis nº1008705

Projeto de Informática do curso de Licenciatura em Engenharia Informática

4 de Dezembro de 2015



Instituto Politécnico da Guarda  
Escola Superior de Tecnologia e Gestão

## **iEcoSys**

**Sistema Inteligente de Gestão de Resíduos Sólidos Urbanos**

Pedro Miguel Amaral dos Reis n.º1008705

Projeto de Informática do curso de Licenciatura em Engenharia Informática

Orientador: Professor Celestino Gonçalves

Coorientador: Mestre Filipe Caetano, Engenheiro na empresa  
MagicKey/IPG

4 Dezembro de 2015

# Ficha Técnica

**Aluno:** Pedro Miguel Amaral dos Reis

**Nº de Aluno:** 1008705

**Estabelecimento de Ensino:** Escola Superior de Tecnologia e Gestão  
Instituto Politécnico da Guarda

**Curso:** Licenciatura em Engenharia Informática

**Docente Orientador:** Professor Celestino Gonçalves

**Coorientador Externo:** Engenheiro Filipe Caetano

**Local de realização do Projeto:** Instituto Politécnico da Guarda

**Endereço:** Rua Dr. Francisco Sá Carneiro, nº 50 6300-559 Guarda

**Telefone:** 271 220 112

# Agradecimentos

## **Agradecimentos Acadêmicos**

Com o finalizar de mais uma etapa na construção do meu conhecimento, que apenas foi possível com a ajuda de todos os docentes e não docentes que ao longo destes anos me acompanharam nas várias componentes letivas do curso ou extra curriculares.

A todos eles um muito obrigado pelo espírito e dedicação que encarnaram no desempenho das suas funções e que me fizeram crescer como pessoa e aluno.

Um Agradecimento especial e em particular aos Orientadores Professor Celestino e Eng.<sup>o</sup> Filipe pela dedicação, atenção e paciência que tiveram ao longo deste trabalho.

Também um agradecimento especial ao MagicKey, na pessoa do Professor Luís Figueiredo onde os conhecimentos adquiridos, em muito foram a base de suporte, essenciais para o desenvolvimento deste projeto.

## **Agradecimentos Pessoais**

A todos aquelas Pessoas Especiais, Amigos ou Familiares que desde sempre estiveram ao meu lado, me incentivaram a lutar, que sempre acreditaram em mim e nos momentos mais difíceis foram o meu suporte, um Grande Muito Obrigado e Aquele Abraço!!!

*“O Maior inimigo do conhecimento não é a ignorância, é a ilusão do conhecimento.”*

**Stephen Hawking**

*“A Melhor Forma de Prever o Futuro é Cria-lo!...”*

**Peter F. Drucker**

# Resumo

A gestão dos resíduos sólidos urbanos constitui um problema complexo para a generalidade dos países desenvolvidos. Presentemente, apenas algumas cidades implementaram pequenos processos de recolha de lixo com alguma inovação. Urge, assim, implementar medidas indutoras de comportamentos sustentáveis, com a participação ativa dos cidadãos, que assegurem a conservação de recursos através da redução e valorização dos resíduos.

Este trabalho descreve o sistema *iEcoSys - Intelligent Ecologic System*. Consiste numa ferramenta tecnológica que identifica individualmente o lixo produzido, utilizando *tags RFID* incorporadas nos sacos do lixo. Ao depositar os resíduos, o ecoponto identifica e pesa cada saco e a informação recolhida é enviada para um servidor recorrendo a tecnologia *Zigbee*.

Chegada a informação ao servidor é inserida na base de dados tornando possível consultar no portal *iEcoSys* os resíduos depositados e inclusive encomendar novos sacos.

Promover a sustentabilidade alterando o paradigma ao invés de pagar o lixo produzido, receber pelo lixo reciclado, constitui o contributo deste trabalho centrado no desenvolvimento do protótipo do *iEcoPonto*.

**Palavras-chave:** Ambiente, Sustentabilidade, Gestão de RSU, Rede de sensores, Zigbee, RFID, Smart Cities.

# Abstract

The management of municipal solid waste is a complex problem for most developed countries. At present, only a few small towns have implemented procedures for collecting garbage in an innovative way. Thus, it is urgent to implement measures that initiate sustainable behavior, with the active participation of citizens, ensuring the conservation of resources through the reduction and recovery of waste.

This paper describes the *iEcoSys* System - Intelligent Ecologic System developed by the authors. It is a technological tool that identifies the waste produced individually, using *RFID* tags embedded in rubbish bags. When depositing waste, the recycling center identifies and weighs each bag and the collected data is sent to a server using *Zigbee technology*.

When the information reaches the server, it is inserted into the database making it possible to see the deposited waste in the *iEcoSys* internet portal, and even order new bags.

Promoting sustainability by changing the paradigm of receiving the recycled rubbish instead of paying for the waste produced, is the contribution of this work centered in development the prototype of *iEcoPonto*.

**Keywords:** Environment, Sustainability, MSW Management, Network Sensors, Zigbee, RFID, Smart Cities.

# ÍNDICE

<b>Capítulo 1 - Introdução</b> .....	1
1.1 Motivação.....	2
1.2 Objetivos .....	2
1.3 Contribuição .....	4
1.4 Estrutura do documento .....	4
<b>Capítulo 2 - Estado da arte</b> .....	5
2.1 Introdução.....	5
2.2 Exemplos existentes .....	5
2.3 Análise Crítica do Estado da Arte .....	6
<b>Capítulo 3 - Ambiente e Smart Cities</b> .....	8
3.1 Introdução.....	8
3.2 Estratégia da União Europeia para a Redução da Pegada Ecológica .....	9
3.3 Portugal – Plano Nacional de Gestão de Resíduos (PNGR) .....	11
3.5 Enquadramento e Relevância do iEcoSys no PNGR – 2011/2020.....	17
3.6 Smart Cities e Ambiente.....	20
<b>Capítulo 4 - Tecnologia Arduino</b> .....	25
4.1 Arduino .....	25
4.1.1 Porquê o Arduino? .....	26
4.1.2 O Arduino Atmega256 ADK.....	26
4.1.3 O Microcontrolador Atmega256 .....	27
4.1.4 Memória.....	27
4.1.5 Entradas e Saídas do Microcontrolador .....	27
4.1.6 Alimentação.....	28
4.1.7 Bootloader .....	29
4.1.8 In-Circuit System Programming .....	29
4.1.9 Clock, Oscilador de Cristal.....	30
4.2 Software .....	30
4.2 Programação e IDE.....	31

<b>Capítulo 5 - Sistema IEcoSys</b> .....	33
5. Descrição do Sistema .....	33
5.1 Descrição do Processo.....	34
5.2 O iEcoPonto .....	39
5.3 Rede Mesh .....	42
5.4 Servidor de Receção de Dados .....	42
5.5 Servidor de Base de Dados .....	43
5.6 Definição de Rotas de Recolha.....	44
5.7 Benefício do cidadão.....	45
<b>Capítulo 6 - Implementação do iEcoPonto</b> .....	46
6.1 Introdução.....	46
6.2 Estrutura do iEcoPonto.....	47
6.2.1 Sinalizadores Luminosos .....	48
6.2.2 iSacos e as Tags RFID .....	49
6.2.3 Identificação dos iSacos .....	50
6.2.4 Pesagem dos iSacos.....	51
6.2.5 Abertura das Portas Frontais.....	52
6.2.7 Módulo de Comunicação XBee.....	56
6.2.8 Painel Solar - Acumuladores.....	60
6.3 Comunicação do iEcoPonto para a Base Station .....	61
6.4 Fluxograma do Algoritmo do iEcoPonto .....	63
<b>Capitulo 7 - Conclusões</b> .....	65
7.1 Conclusão.....	65
7.2 Trabalho Futuro .....	66
<b>Bibliografia</b> .....	67
<b>ANEXOS</b> .....	69
<b>Anexo I</b> .....	69
<b>Anexo II</b> .....	70
<b>Anexo III</b> .....	77
<b>Anexo IV</b> .....	78
<b>Anexo V</b> .....	79

<b>Anexo VI</b> .....	81
-----------------------	----

## Lista de Figuras

Figura 1 - As Várias dimensões da gestão de resíduos.....	11
Figura 2 - Hierarquia para a gestão dos resíduos.....	12
Figura 3 - Níveis de planeamento de gestão de resíduos .....	13
Figura 4 - Relação dos objetivos estratégicos. ....	17
Figura 5 - Dinâmica das smart cities e o ambiente.....	22
Figura 6 - Arduino Mega ADK. ....	25
Figura 7 - Exemplo de uma janela do IDE Arduino. ....	30
Figura 8 - Exemplo de um programa que recorre a uma librarie.....	31
Figura 9 - Algoritmo geral da arquitetura do Sistema iEcoSys. ....	33
Figura 10 - Registo do cidadão no Sistema.....	35
Figura 11 - Compra, venda e encomenda de iBags. ....	36
Figura 12 – O iBag.....	36
Figura 13 - Depósito no iEcoponto. ....	38
Figura 14 - Sistema informático / eletrónico de leitura processamento e envio de dados do iEcoPonto.....	39
Figura 15 – Obtenção do iEcoPonto. ....	40
Figura 16 -O iEcoPonto. ....	41
Figura 17 - Rede Mesh.....	42
Figura 18 - Servidor de Receção de Dados. ....	43
Figura 19 – Servidor de Base de Dados. ....	44
Figura 20 - Benefício do Sistema iEcoSys / cidadão. ....	45
Figura 21 - Arquitetura do iEcoSys implementada inicialmente.....	46
Figura 22 - iEcoPonto em fase de construção, vista frontal e posterior. ....	47
Figura 23 - Exemplo de ligação de Leds ao Arduino Mega.....	48
Figura 24 - Exemplo de programação das portas digitais .....	48
Figura 25 - Tag RFID.....	49

Figura 26 - Leitor RFID adaptado à estrutura da Balança.....	50
Figura 27 - Ligação do leitor RFID ao Arduino. ....	50
Figura 28 - Balança. ....	51
Figura 29 - Conversor RS232 - TTL de cor verde, soldado a uma placa de circuito impresso. ....	51
Figura 30 - Exemplos de servomotores. ....	52
Figura 31 - Exemplo de uma Ligação de um Servo Motor ao Arduino.....	53
Figura 32 - Circuitos Integrados L293D (Ponte H) utilizados no iEcoPonto. ....	54
Figura 33 - Esquema elétrico de ligação do CI L293D e os motores. ....	54
Figura 34 - Exemplo de ligação de um motor CC a um CI L293D controlado por um microcontrolador Arduino.....	55
Figura 35 - Módulo XBee. ....	56
Figura 36 - Topologias de redes ZigBee.....	57
Figura 37 - Uma janela do software do X-CTU. ....	58
Figura 38 - Modulo XBee de cor Azul, montado no circuito impresso desenhado para o efeito.....	59
Figura 39 - Circuito impresso com o regulador de tensão LM 1117 para alimentar o módulo XBee à esquerda e pinout do regulador, à direita. ....	60
Figura 40 - Circuito de alimentação do microcontrolador. ....	60
Figura 41 - Ligações da Base Station. ....	61
Figura 42 – Fórmula de cálculo do divisor de tensão.....	62
Figura 43 - Fluxograma do funcionamento do iEcoPonto.....	63
Figura 44 - Protótipo do iEcoPonto. ....	64

## Lista de Tabelas

Tabela 1 - Evolução das políticas ambientais Europeias .....	9
Tabela 2 - Metas a atingir 2011 - 2020.....	16
Tabela 3 - Objetivos operacionais e respetivas ações.....	18
Tabela 4 - Características Atmega 256 ADK. ....	27

# Glossário

AAE Avaliação Ambiental Estratégica  
AEA Agência Europeia do Ambiente  
AMP OP Amplificador Operacional  
ANR Autoridade Nacional de Resíduos  
APA Agência Portuguesa do Ambiente  
AREF *Analog Reference* - Pino de referência de tensão para entradas analógicas  
AVR *Alf (Egil Bogen) and Vegard (Wollan)'s RISC processor*  
C Linguagem de programação estruturada;  
CAD *Computer-aided Design*  
CE Comissão Europeia  
CEE Comunidade Económica Europeia  
CI Circuito Integrado  
COM Porta de comunicação responsável pela entrada e saída de dados  
DC *Direct Current*  
DGAE Direção-Geral das Atividades Económicas  
DTR *Data Terminal Ready*  
E/S Entrada e Saída de dados  
EEA European Environment Agency  
EEPROM *Electrically-Erasable Programmable Read-Only Memory*  
FFD *Full Function Device*  
FTDI *Future Technology Devices International*  
GAYT *Get As You Throw*  
GND *Ground*  
GPL *General Public License*  
I&D Investigação e Desenvolvimento  
ICSP *In-Circuit System Programming*  
IDE *Integrated Development Environment*  
iEcoSys – Sistema Inteligente de Gestão de Resíduos Sólidos Urbanos  
INE Instituto Nacional de Estatística  
INR Instituto dos Resíduos  
IoE *Internet of Everything*  
IoT *Internet of Things*  
iSaco – Saco Inteligente com identificador RFID  
ISP *In-System Programmer*  
IWM Integrated Waste Management  
LED *Light Emitting Diode*

*LGPL Library General Public License*  
MAOT Ministério do Ambiente e do Ordenamento do Território  
ME Ministério da Educação  
MOR Mercado Organizado de Resíduos  
NPN Ligação lógica negativa  
OCDE Organização para a Cooperação e Desenvolvimento Económico  
ONU Organização das Nações Unidas  
OP Objetivos Operacionais  
PAA Programa de Ação em matéria de Ambiente  
PAYT Pay As You Throw  
*PERH Plano Estratégico Resíduos Hospitalares*  
PERSU Plano Estratégico de Resíduos Sólidos Urbanos  
PESGRI Plano Estratégico de Gestão dos Resíduos Industriais  
PIRSUE Plano de Intervenção de Resíduos Sólidos Urbanos e Equiparados  
PNGR Plano Nacional de Gestão de Resíduos  
PNP Ligação lógica positiva  
*PWM Pulse with Modulation*  
RAP Responsabilidade Alargada do Produtor  
REEE Resíduos de Equipamentos Elétricos e Eletrónicos  
*RFD Reduced Function Device*  
RFID Radio Frequency Identification  
RH Resíduos Hospitalares  
RI Resíduos Industriais  
RSU Resíduos Sólidos Urbanos  
SIRAPA Sistema Integrado de Registo da Agência Portuguesa do Ambiente  
*SPI Serial Peripheral Interface*  
SPV Sociedade Ponto Verde  
*SRAM Static Random Access Memory*  
TTL *Time To Live*  
*TWI Two Wire Interface*  
UART *Universal Asynchronous Receiver/Transmitter*  
UE União Europeia  
USB *Universal Serial Bus*  
*WPAN Wireless Personal Area Network*  
*ZC ZigBee Coordinator*  
*ZED ZigBee End Device*  
*ZR ZigBee Router*



# Capítulo 1 - Introdução

Do ponto de vista ambiental é urgente encontrar soluções pedagógicas para que os cidadãos participem de forma ativa e sejam consciencializados a reciclar exaustivamente o lixo produzido, de forma a minimizar a pegada ecológica da nossa civilização que se torna cada vez mais pesada para o planeta Terra.

A não reciclagem dos resíduos, para além da questão ambiental, implica um encargo financeiro para as entidades envolvidas no processo, nomeadamente para o cidadão que muitas vezes de forma injusta paga o lixo que não produziu.

Neste momento os processos de reciclagem implementados para a recolha seletiva e reciclagem de resíduos não sofreram qualquer inovação tecnológica praticamente desde a sua conceção [1].

Após ter sido feito o levantamento das evoluções tecnológicas no processo de recolha seletiva de resíduos verifica-se que apenas pontualmente em alguns países, algumas cidades, implementaram pequenos processos de recolha com alguma inovação tecnológica que de alguma forma se tornam pedagogicamente mais atrativos e justos para o cidadão, mas ainda muito pouco rigorosa, tendo todos eles como objetivo final o conceito *PAYT (Pay As You Throw)* [2].

Entretanto, têm surgido em comunidades científicas de investigação para o desenvolvimento de alguns sistemas com o objetivo de colmatar a falha existente dos sistemas já implementados, mas com um custo de manutenção considerável em termos de consumos energéticos e comunicações [3] [4].

Para além dos factos observados, os Países pertencentes à União Europeia têm metas a atingir de acordo com os protocolos firmados a nível Europeu relativamente à percentagem de lixo reciclado, não obstante, o acumular da crise económica levou à quebra do consumo e conseqüente redução significativa dos resíduos a reciclar. Segundo as diretivas da União Europeia, os governos nacionais terão de tomar medidas para que sejam reciclados até 2020, 50% do papel, plástico e vidro dos resíduos domésticos [8].

Embora o número de ecopontos tenha crescido de uma forma quase exponencial nos últimos anos, sobretudo em ambiente urbano, verificou-se que as taxas de reciclagem não acompanharam esta subida. Em Portugal, de acordo com

os dados da EGF – Empresa Geral de Fomento, gestora do processo, houve um decréscimo de quase 12% na reciclagem de RSU – Resíduos sólidos Urbanos, de tal forma que no ano de 2012 foram recicladas 3.197.000 Ton, enquanto que 3 anos antes, no ano de 2009, foram reciclados 3.623.000 Ton, ano em que foi atingido o pico máximo de reciclagem em Portugal nos últimos 10 anos [9].

Desta forma, existe a necessidade de tomar medidas de educação ambiental no sentido de tornar a sociedade portuguesa mais ecológica, e de desenvolver técnicas pedagogicamente atrativas de forma a nos aproximarmos de países como a Suíça, onde no ano de 2011 35% dos resíduos produzidos foram reciclados, contra os 12% reciclados em Portugal [10].

O trabalho desenvolvido a seguir apresentado, o *iEcoSys – Sistema Inteligente de Gestão de Resíduos Sólidos Urbanos*, visa criar um circuito fechado de fluxo de informação para dar resposta a um processo de gestão de resíduos urbanos com vista a que o cidadão receba pelo lixo corretamente reciclado – conceito *GAYT (Get As You Throw)*.

O Sistema *iEcoSys* já resultou na publicação de um pedido de patente por parte do Instituto Nacional da Propriedade Industrial (INPI) [24] e na participação em quatro conferências internacionais, com a respetiva publicação de artigo científico [23][25][26][27].

## 1.1 Motivação

A motivação para a elaboração deste trabalho vem no seguimento da necessidade de desenvolver um sistema que permitisse retribuir um incentivo aos cidadãos, em função dos resíduos produzidos e reciclados, de forma justa e coerente, pondo assim os conhecimentos adquiridos ao longo do curso em prática e desta forma responder a uma problemática da nossa sociedade, tanto de ordem nacional como de ordem Global.

## 1.2 Objetivos

A solução encontrada para responder ao problema foi o *iEcoSys – Sistema Inteligente de Gestão de Resíduos Sólidos Urbanos*, apresentada e descrita neste trabalho.

O *iEcoSys* recorre a tecnologia amplamente divulgada no mercado, de baixo custo e com alta fiabilidade, tais como as tecnologias *RFID*, *ZigBee*, *Arduino*, entre outras.

Este sistema permite identificar o tipo de resíduos produzidos individualmente por cada cidadão e a respetiva quantidade. Desta forma é possível atribuir um incentivo ao cidadão em função da quantidade e do tipo de resíduos reciclados. Para que o projeto satisfaça a finalidade a que se destina é necessário:

- Desenvolver uma solução que permite identificar o tipo de resíduos produzidos individualmente por cada cidadão e a respetiva quantidade.
- Permitir que o *cidadão* se registe para aderir ao projeto. O registo pode ser *On-line* ou através de uma aplicação.
- Permitir que os sacos do lixo tenham um identificador único que os permita associar ao *cidadão* e ao *Tipo de Lixo* que acondicionam.
- Permitir ao *cidadão* adquirir, *iSacos* - sacos com identificador, *on-line* ou presencialmente e associar esses *iSacos* ao mesmo.
- Permitir identificar os *iSacos* no ato de depósito, pesar o mesmo e enviar essa informação através de uma rede de dados.
- Permitir receber a informação através da rede de dados e inserir essa informação numa *Base de Dados*.
- Ter uma *Base de Dados* por forma a associar ao *cidadão* os *iSacos* adquiridos pelo mesmo e o respetivo *tipo de lixo*. Consultar a quantidade, em quilos, de lixo depositada por *tipo de lixo*. Consultar os *iSacos* comprados e os depositados. Consultar a data e hora de depósito do lixo.

O principal objetivo que se pretende atingir neste trabalho é desenvolver um protótipo do *iEcoPonto*. O protótipo deve permitir identificar o depósito de um *iSaco* pelo cidadão, o tipo de lixo e o seu peso comunicando essa informação à *Base Station*, ou seja o local onde essa informação é recebida. O *hardware* principal e de maior relevo é o leitor *RFID*, o *Arduino Mega ADK*, a balança com *RS232* e o *XBee*. O *iEcoPonto* deve ser alimentado por uma bateria e um painel solar.

O segundo objetivo é o desenvolvimento de uma *Base Station* para a receção de dados relativos aos depósitos vindos dos *iEcoPontos* sendo constituída

por um *XBee*, um *FTDI Chip* e um *Servidor*, como principais elementos de *hardware*.

### 1.3 Contribuição

Sendo o ambiente um problema que diz respeito a todos, como sociedade globalizada que somos, pretende-se desta forma contribuir para a redução da pegada ecológica no planeta, pelas razões apresentadas.

### 1.4 Estrutura do documento

O documento compreende 7 capítulos, incluindo a presente introdução.

No segundo capítulo é apresentado o estado da arte, onde se faz referência a algumas das soluções parciais já existentes no mercado e é apresentada a opinião em relação ao *iEcoSys*.

No terceiro capítulo é apresentado o enquadramento do projeto *iEcoSys* em relação às estratégias e políticas ambientais atuais, quer ao nível de Portugal, quer a nível da Comunidade Europeia.

No quarto capítulo é descrita a plataforma Arduino, o elemento central do projeto em termos de *hardware* e *software*.

No quinto capítulo é descrita a análise pormenorizada dos requisitos necessários da solução.

No sexto capítulo descreve-se a implementação da solução proposta, apresentando o protótipo implementado em laboratório.

Finalmente, no sétimo capítulo, são apresentadas as conclusões mais relevantes do trabalho e as perspetivas de desenvolvimento que se pretendem efetuar no futuro.

# Capítulo 2 - Estado da arte

## 2.1 Introdução

O estudo realizado para obter o estado da arte permite concluir que não existem sistemas com as mesmas características do sistema apresentado. Contudo, existem sistemas que em parte executam algumas das funcionalidades do sistema. Assim, será feita uma análise geral dos sistemas existentes em comparação com o iEcoSys. Estes sistemas já existentes, para além de somente implementarem em parte as funcionalidades do iEcoSys, são direcionados para a aplicação do conceito PAYT [7]. O iEcoSys implementa o conceito GAYT, ou seja, Get As You Throw, [25] sendo que apenas dois dos sistemas encontrados têm como finalidade a implementação deste conceito.

## 2.2 Exemplos existentes

Os sistemas existentes que de alguma forma executam algumas das funcionalidades do sistema iEcoSys e implementam o conceito GAYT são:

- Sistema PFAND Station:

O sistema que está implementado na Holanda, Finlândia e Alemanha conhecido como PFAND Station, está presente em alguns supermercados onde os cidadãos trocam garrafas de plástico PET por dinheiro ou talões de desconto nos supermercados [19].

Em Pequim, na China, desenvolvido pela Incom, estes equipamentos trocam as garrafas de plástico PET por bilhetes de metro. Os equipamentos estão disponíveis nas várias estações espalhadas na capital chinesa [18].

- Sistema BigBelly Station:

Outro sistema semelhante foi desenvolvido por uma empresa norte americana nos EUA, a BigBelly, denominados de Estações BigBelly e implementado em New York. Estas estações disponíveis em locais públicos, trocam latas por sinal WiFi. Grande parte da alimentação energética das Estações BigBelly é providenciada por painéis solares e com uma estética visualmente agradável [20].

Os sistemas que implementam o conceito PAYT são genericamente sistemas de acesso a contentores por chave eletrónica. A chave eletrónica consiste num

cartão com banda magnética ou com tag RFID que identifica o utilizador pelos depósitos que faz. Consequentemente os cidadãos pagam uma taxa em função dos depósitos que efetuam.

O sistema mais próximo do iEcoSys mas ainda assim com característica PAYT, foi um sistema de recolha de lixo implementado na Alemanha. O sistema permite a identificação de contentores de lixo por RFID à passagem dos camiões de recolha de lixo. Os mesmos camiões também permitem a pesagem do lixo, sendo a informação recolhida num computador de bordo existente na viatura. À posteriori os dados são inseridos no sistema através de um cartão de dados à chegada dos respetivos camiões de recolha do lixo à central. Por fim é enviada uma conta individual para cada cidadão para pagamento do lixo, produzido num determinado período de tempo e em função do lixo que a comunidade ou bairro deposita num determinado contentor. O sistema foi implementado em várias cidades alemãs, nomeadamente Bremen, Cologne e Dresden [12].

### 2.3 Análise Crítica do Estado da Arte

O iEcoSys é um sistema inteligente de gestão de resíduos sólidos que permite a gestão completa e integrada do fluxo de informação que vai desde a compra de iBags que são associados ao cidadão, passando pelos depósitos, permitindo monitorizar os iEcopontos, entre outros.

Os sistemas similares de recolha de resíduos são limitados a vários níveis. Os sistemas PFAND Station, Estações BigBelly e o sistema da Incom, RVM – reverse vending machine, estão limitadas ao tipo de resíduo e não associam o tipo de resíduo ao cidadão.

O sistema de gestão de resíduos que mais se assemelha ao iEcoSys é o sistema implementado em algumas cidades alemãs, anteriormente referido. O sistema, em comparação como o iEcoSys, não permite associar o tipo de resíduos e a quantidade de resíduos depositados ao cidadão, apenas associa o lixo a um bairro ou condomínio de cidadãos que à posteriori recebem a fatura relativa ao pagamento do mesmo.

A comunicação de dados também é feita de forma diferente. O iEcoSys comunica os dados em tempo real, através de uma rede ZigBee implementada pelos próprios iEcoPontos que enviam os dados para uma base station. Relativamente aos

depósitos efetuados pelos cidadãos, o sistema anteriormente referido apenas recolhe dados de pesagem e identificação por RFID no momento da recolha do lixo por uma viatura equipada com uma balança e um leitor RFID que lê os ID das tags existentes nos contentores.

Face ao exposto, faz todo o sentido o desenvolvimento do sistema iEcoSys, que torna possível a identificação do tipo de resíduos e a quantidade de resíduos que cada cidadão deposita, permitindo dessa forma, dependendo sempre do modelo de negócio a implementar, a compensação dos cidadãos pela reciclagem que fazem de acordo com o conceito GAYT [23].

## Capítulo 3 - Ambiente e Smart Cities

### 3.1 Introdução

A evolução das civilizações ao longo dos séculos tem tido como pressuposto do seu desenvolvimento um modelo assente nos recursos ilimitados, tanto a nível dos recursos naturais, como da capacidade da natureza para absorver e transformar os resíduos efluentes gerados pela atividade humana.

Esta forma de estar tem estado na génese de alguns problemas ambientais no planeta como por exemplo as alterações climáticas.

O início da revolução industrial no século XVIII, pioneira na absorção desmesurada dos recursos naturais e o culminar da globalização no século XX responsável pela intensificação das atividades económicas e as suas consequências, tendo como ponto alto a década de 80, contribuíram para o esgotamento dos recursos naturais e da capacidade natural para a absorção e processamento dos resíduos.

A consciência ambiental tem vindo a ganhar força. Com a mudança de milénio, a questão ambiental passou a fazer parte das agendas políticas, das empresas, da sociedade em geral e dos cidadãos em particular. Desta forma as políticas ambientais têm vindo a ganhar relevância nas últimas décadas, sendo hoje em dia indissociáveis de um desenvolvimento social, económico e sustentável equilibrados.

Nas políticas ambientais em particular, tem sido dada especial atenção aos resíduos de forma progressiva, juntamente com os recursos naturais uma das áreas prioritárias da União Europeia (decisão n.º 1600 / 2002 CE) [17].

### 3.2 Estratégia da União Europeia para a Redução da Pegada Ecológica

Evolução científica e técnica na área do ambiente, em particular no que toca à relação do homem com o ambiente, levou à alteração das políticas ambientais, com uma evolução acentuada nas últimas décadas em todo o mundo e em particular na Europa (Tabela 1).

Período	PAA	Tipo de visão e enfoque	Tipo de Instrumentos	Alguns exemplos a nível nacional
Anos 70	Políticas anteriores e 1º PAA	<i>Business-as-usual</i> <i>Vocacionada para os processos</i> <ul style="list-style-type: none"> <li>• Eliminação da poluição</li> <li>• Comando e controlo</li> <li>• Tecnologias de fim de linha</li> </ul>	Legislativos	
Anos 80	2º e 3º PAAs	<i>Vocacionada para os processos</i> <ul style="list-style-type: none"> <li>• Tecnologias de fim de linha</li> <li>• Controlo da poluição</li> <li>• Introdução de importância da prevenção</li> </ul>	Legislativos	Lei de bases do ambiente
Anos 90	4º e 5º PAAs	<i>Vocacionada para os produtos</i> <ul style="list-style-type: none"> <li>• Princípios de hierarquia de gestão</li> <li>• Responsabilidade alargada do produtor</li> <li>• Introdução dos princípios de desenvolvimento sustentável</li> </ul>	Legislativos Outros instrumentos (ex. económicos)	Criação do Instituto dos Resíduos DL 239/1997 SPV PERSU I
Finais de 90	5º e 6º PAAs	<i>Vocacionada para os produtos e uso de recursos</i> <ul style="list-style-type: none"> <li>• Uso de recursos (saídas ligadas às entradas e vice versa)</li> <li>• Formalização do ambiente como política transversal</li> <li>• Política integrada de produto</li> </ul>	Legislativos Económicos Fiscais Informação	PCSGRI PNAPRI PERH ENDS 2015 DL 178/2006
Princípios Séc. XXI	6º PAA	<i>Maior enfoque nos sistemas</i> <ul style="list-style-type: none"> <li>• Uso de recursos</li> <li>• Maior enfoque na prevenção e estratégias de consumo mais sustentáveis</li> <li>• Sinergias entre diversos ciclos de vida</li> </ul>	Maior integração e sinergias entre vários tipos de instrumentos	Programa de compras sustentáveis Princípio da responsabilidade pela gestão PNGR

Tabela 1 - Evolução das políticas ambientais Europeias  
in PNGR 2011 – 2020 [17]

No início da União Europeia, nos tratados celebrados, não havia referências concretas no que toca à proteção ambiental.

A única referência a questões ambientais foi feita no tratado de Roma para garantir a competição justa entre países, que alerta para se evitar o dumping ambiental. Neste período, no que toca às políticas ambientais, as empresas apenas

melhoram os processos de fabrico e minimização dos danos sem qualquer prevenção. Havia unicamente a regulação do aterro dos resíduos e das tecnologias para o tratamento dos mesmos para as indústrias que mais poluíam. Só na década de 70, na cimeira de Paris, é que a Comunidade Europeia pela primeira vez produziu documentos relativos a políticas ambientais com o primeiro programa de ação em matéria de ambiente, onde referiam o princípio do poluidor pagador e a racionalização dos recursos naturais.

Neste período, no que toca a políticas ambientais, as empresas apenas melhoraram os processos de fabrico nos últimos 10 anos sem qualquer prevenção,

havendo unicamente a regulamentação do aterro dos resíduos e aplicação de tecnologias para o tratamento dos resíduos para as indústrias com mais poluição.

Com o passar dos anos, devido a fatores económicos, sociais e políticos de origem interna e externa, as políticas ambientais foram-se ajustando nos diversos *PAA (Programa de Ação Ambiental)*. Na década de 80 houve uma mudança de paradigma com o segundo e o terceiro *PAA*. Deixou haver o conceito de *controlo de poluição* para passar a haver *prevenção da poluição*. Contudo, isto não se traduziu na implementação de legislação europeia. Apenas existia o controlo da poluição assente na proibição de substâncias e restrição de processos.

A proteção ambiental passou a ser consagrada como objetivo europeu em 1987 com o ato único europeu que alterou o tratado de Roma em 1989. Houve então uma verdadeira alteração de paradigma ambiental com o quarto *PAA*, onde a prevenção se torna a principal prioridade na estratégia comunitária para a gestão de resíduos. Foi durante este período de transição que se começou a explorar a possibilidade de se recorrer a outras formas de incentivos, tanto de ordem económica, como de divulgação de informação relativa ao ambiente, acompanhados de atualização da legislação

Ainda assim, houve uma subversão relativa ao entendimento da produção e gestão de resíduos no que toca às soluções para eliminar os mesmos, levando à adoção de novas políticas, tais como a prevenção, a reutilização e a reciclagem, entre outras, em oposição aos aterros e lixeiras. O desenvolvimento sustentável passou a ser objetivo prioritário da União Europeia com o quinto *PAA (1993/2000)* e o tratado de Maastricht.

O ambiente passou a estar presente nas políticas afetas aos vários setores e as medidas adotadas contemplam a responsabilidade partilhada dos vários intervenientes.

Foi aplicado o conceito *RAP - Responsabilidade Alargada do Produtor*, definindo fluxos de resíduos em função da sua quantidade e perigosidade. A mudança nas políticas ambientais em torno dos resíduos tem sido cada vez maior

atingindo o seu ponto alto com o sexto PAA (2002/2010) sobre o lema “O nosso futuro, a nossa escolha”.

Os resíduos tornam-se parte integrante do sistema económico, permitindo assim uma gestão sustentável dos recursos naturais como uma contribuição efetiva [17].

“Neste contexto, a visão de que os resíduos são apenas substâncias e objetos que urgem serem eliminados (“lixo”) de modo a reduzir a poluição a si associada, encontra-se ultrapassada. Um exemplo paradigmático desta evolução é o que combina fluxos de resíduos, métodos de recolha, tratamento e eliminação, com o objetivo de alcançar benefícios ambientais, otimização económica e aceitabilidade social (Coleman et al., 2003), o conceito de Gestão Integrada dos Resíduos.” (Integrated Waste Management, IWM) [17]

### 3.3 Portugal – Plano Nacional de Gestão de Resíduos (PNGR)

#### Definição de Resíduos

Os resíduos a nível nacional têm como definição qualquer “substância ou objeto de que o detentor se desfaz, tem a intenção ou a obrigação de se desfazer” decreto-lei n.º178/2006. A definição de resíduo está também definida no direito Europeu desde 1975.

Com o objetivo de aproximar a gestão dos materiais dos recursos materiais no sistema económico, foram introduzidos os conceitos de subprodutos e de fim do estatuto de resíduo, “end of wast”, diretiva n.º 2008/98 CE [17].



Figura 1 - As Várias dimensões da gestão de resíduos.  
in PNGR 2011 - 2020 [17]

## Gestão de Resíduos

Um objeto ou substância, apenas se torna resíduo, se o detentor se desfaz do objeto ou substância. Para o detentor se desfazer da mesma requer que este execute um conjunto de ações tais como a recolha, transporte, armazenagem, triagem, tratamento, valorização e eliminação. A este conjunto de atividades dá se o nome de gestão de resíduos, diretiva n.º 2006/12 CE (Figura 1).

Para além das atividades anteriormente referidas, existem outras dimensões tais como, regulação, legislação e políticas relativas à prevenção e produção de resíduos. A gestão de resíduos deve obedecer a uma hierarquia de uma pirâmide invertida Figura 2, que ilustra de forma clara as ações a tomar em relação à gestão destes, devendo ser sempre uma referência de forma a se obterem os melhores resultados ambientais. Sendo o princípio da *prevenção e redução* muito importante e o principal, existem outros princípios também eles importantes, tais como, o *princípio da responsabilidade pela gestão*, o *princípio da equivalência* e o *princípio da autossuficiência* [17].

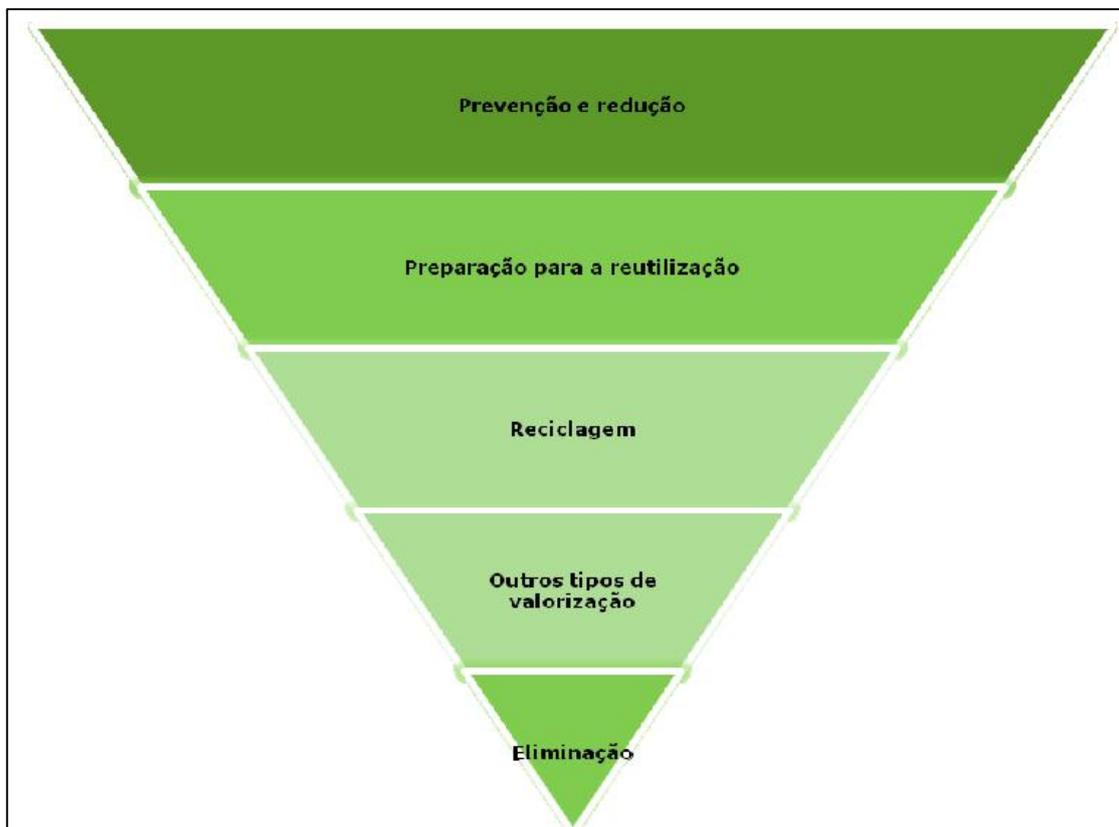


Figura 2 - Hierarquia para a gestão dos resíduos.

in PNGR 2011 - 2020 [17]

## A Caminhada de Portugal

Portugal definiu várias orientações estratégicas para a gestão de resíduos em vários planos, sendo os de maior relevância aqueles que foram definidos a partir dos finais da década de 90.

Dos planos com maior importância podemos destacar o *PERSU* – Plano Estratégico de Resíduos Sólidos Urbanos, o *PERH* – Plano Estratégico de Resíduos Hospitalares e o *PESGRI* – Plano Estratégico de Gestão dos Resíduos Industriais.

No entanto, com a aprovação do Decreto – Lei n.º 178/2006, foi determinado que fosse constituído o Plano Nacional de Gestão de Resíduos (PNGR).

A Agência Portuguesa do Ambiente (APA), como Autoridade Nacional de Resíduos (ANR), recebeu do governo a tarefa de elaborar esse *PNGR*.

No *PNGR* são definidas metas a nível nacional relativas à política de gestão de resíduos e às normas diretoras para que estas sejam igualmente implementadas e de forma coesa (Figura 3).

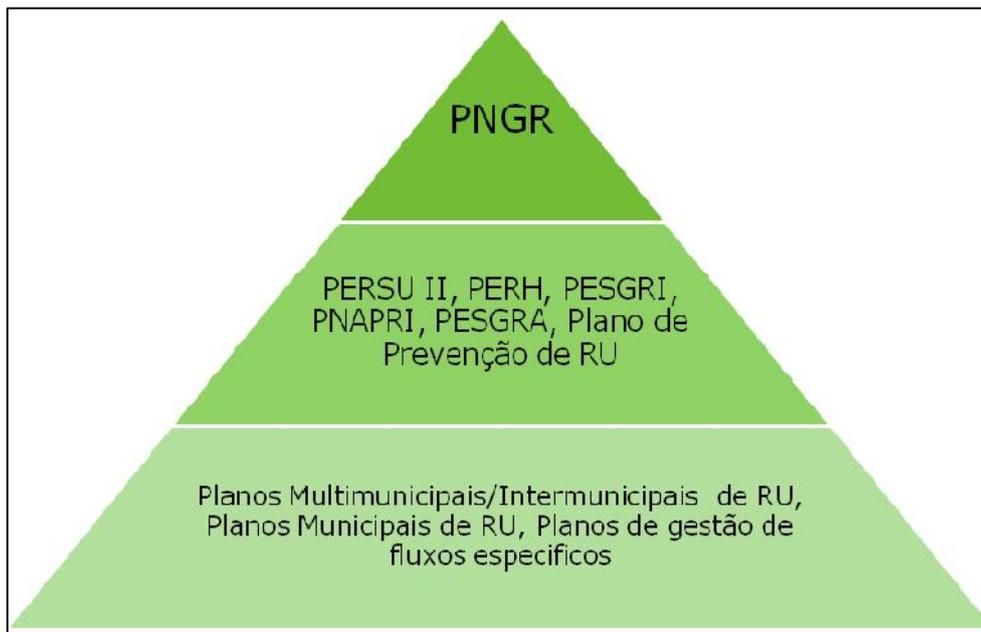


Figura 3 - Níveis de planeamento de gestão de resíduos

in *PNGR 2011 – 2020* [17]

No seguimento do mesmo plano é definida também a implementação de um conjunto de infraestruturas de eliminação e valorização de todo o tipo de resíduos,

recorrendo a várias tecnologias com custos economicamente viáveis que venham ao encontro do N.º1 do artigo 14 do decreto-lei anteriormente referido.

Assim, o PNGR é proposto pelo ministro responsável pela área do ambiente e aprovado pelo conselho de ministros depois de ouvida a Assembleia Nacional de Municípios Portugueses.

Este documento é orientador das políticas de gestão de resíduos nos anos subsequentes, com a implementação de planos específicos de forma a ir ao encontro do plano na especificidade de cada área produtora de resíduos.

Contudo, o *PNGR*, de acordo com a diretiva quadro número 2008/98 CE do parlamento europeu obriga a que qualquer *PNGR* deve conter:

- A verificação da situação atual da gestão dos resíduos produzidos na área onde se pretende implementar o plano.
- Medidas a implementar para reutilização, reciclagem e formas de valorização e eliminação de resíduos.
- A avaliação da forma como o plano irá dar apoio para se atingirem as metas referidas nas diretivas.

Entretanto, Portugal como estado membro da União Europeia, deve desenvolver programas de forma a prevenir os resíduos e incluir os mesmos nos planos nacionais de gestão de resíduos. Assim, surgiu neste enquadramento o documento: “*Plano Nacional de Gestão de Resíduos 2011-2020*”, onde são apresentados os objetivos, visão e respetiva operacionalização de forma a guiar a gestão dos resíduos em Portugal até 2020 [17].

### **PNGR – 2011/2020, Visão, Estratégia e Objetivos**

Os dados recolhidos ao longo do tempo demonstram que o consumo de recursos naturais tem variado no sentido de aumentar. Então tem existido uma preocupação de combater a ineficiência do aproveitamento de matérias-primas resultantes de produtos úteis. Face ao exposto, existe uma necessidade intrínseca de um contínuo combate de forma a obter uma redução sistemática do impacto ambiental devido às atividades de gestão de resíduos. Neste contexto, existe um conjunto de estratégias ambientais de índole nacional e europeia [17].

## **Estratégia**

A primeira grande estratégia adotada pela União Europeia prende-se com a utilização sustentável dos recursos naturais para os próximos 25 anos. Pretende-se com esta estratégia a utilização dos recursos naturais de forma mais eficaz e sustentável, no respetivo ciclo de vida, diminuindo assim o consumo dos recursos e por forma a não interferir negativamente no desenvolvimento económico [17].

## **Estratégia 2020**

A Estratégia 2020, assente numa economia inteligente, tem como pilares fundamentais a sustentabilidade e a inclusão para que desta forma se atinjam elevados níveis de emprego e coesão social. A estrutura de apoio onde esta economia assenta, é composta com base numa economia de baixo carbono, utilização dos recursos de forma eficiente, a proteção do ambiente e em tecnologias verdes [17].

## **Green Economy**

Esta estratégia tem como principal linha orientadora um conjunto de políticas que promovem a *green economy* assente na sustentabilidade e na política integrada de produtos. Esta estratégia converge por um lado as políticas e por outro os mais variados instrumentos e objetivos comuns na mesma linha. Em particular a redução dos impactos ambientais ao longo do ciclo de vida de produtos e serviços e aplicação do princípio do poluidor pagador de forma a promover a comercialização de produtos mais ecológicos. Pretende-se também a consciencialização dos cidadãos e das empresas e assim mais uma vez, minimizar o impacto ambiental dos produtos nos seus variados ciclos de vida [17].

## **Visão**

Sendo atualmente a gestão de resíduos considerado o fim do ciclo de vida dos mesmos, a visão do PNGR passa por tornar a gestão dos resíduos como continuidade do ciclo de vida dos mesmos ou seja criar um círculo em que quando um produto não seja útil ao produtor/detentor se torne uma matéria-prima.

Ao ser integrado novamente pelo sistema económico sempre que seja viável do ponto de vista ambiental, técnico ou socioeconómico, torna-se desta forma possível a redução ao mínimo do consumo de novas matérias-primas e aliviar o ambiente, sendo esta a visão subjacente ao PNGR [17]:

*“Promover uma gestão de resíduos integrados no ciclo de vida dos produtos, centrada numa economia tendencialmente circular e que garanta uma maior eficiência na utilização dos recursos naturais.”*  
in PNGR 2011 – 2020 [17]

## Objetivos

No seguimento das normas, conceitos, diretivas e visão, referidas anteriormente, o PNGR 2011-2020 tem dois pilares principais como objetivos estratégicos [17]:

- 1 - Promover a eficiência da utilização de recursos naturais na economia,
- 2 - Prevenir ou reduzir os impactos adversos decorrentes da produção e gestão de resíduos.

in PNGR 2011 – 2020 [17]

Objectivo estratégico 1 - Promover a eficiência da utilização de recursos naturais na economia					
Meta estratégica	Indicador de realização	Valor de referência	Meta 2013	Meta 2016	Meta 2020
1. Dissociar o crescimento económico do consumo de materiais	PIB a preços constantes <sup>20</sup> /CIM (k€/t)	0,75 (ano 2007)	0,83	0,87	0,93
2. Dissociar o crescimento económico da produção de resíduos	Produção de resíduos/ PIB a preços constantes <sup>21</sup> (t/k€)	0,18	0,17	0,15	0,13
3. Aumentar a integração de resíduos na economia	(Preparação para reutilização + Reciclagem)/(Produção de resíduos) (%)	41,7%	45,0%	55,0%	70,0%
Objectivo estratégico 2 - Prevenir ou reduzir os impactos adversos decorrentes da produção e gestão de resíduos					
Meta estratégica	Indicador de realização	Valor de referência	Meta 2013	Meta 2016	Meta 2020
1. Reduzir a produção de resíduos	Produção de resíduos (2009 = índice 100)	100 (28,8 Mt)	95,0	87,0	80,0
2. Reduzir a quantidade de resíduos eliminados	Eliminação de resíduos (2009 = índice 100)	100 (15,2 Mt)	90,41	66,24	38,07
3. Reduzir a emissão de gases com efeito de estufa do sector de resíduos	Mt CO <sub>2</sub> eq. emitidos para atmosfera pelo sector de gestão de resíduos	7,69 (ano 2007)	6,01	5,87	5,68

Fontes: Com base em APA/INE (2010), Eurostat (2010), Resource productivity, INE (2010), Contas nacionais e FMI (2010), World Economic Outlook - October 2010.

Tabela 2 - Metas a atingir 2011 - 2020

in PNGR 2011 - 2020 [17]

Para atingir os objetivos estratégicos foram definidos um conjunto de metas, Tabela 2, de forma a quantificar e avaliar o desempenho do mesmo, que em muito vão depender dos objetivos operacionais. Os objetivos e metas definidos de forma a alcançar os objetivos estratégicos são (Figura 4) [17]:

- Dissociar o crescimento económico do consumo de materiais
- Dissociar crescimento económico da produção de resíduos
- Aumentar a integração dos resíduos na economia
- Reduzir a produção de resíduos
- Reduzir a quantidade de resíduos eliminados
- Reduzir emissão de gases com efeito de estufa do setor dos resíduos

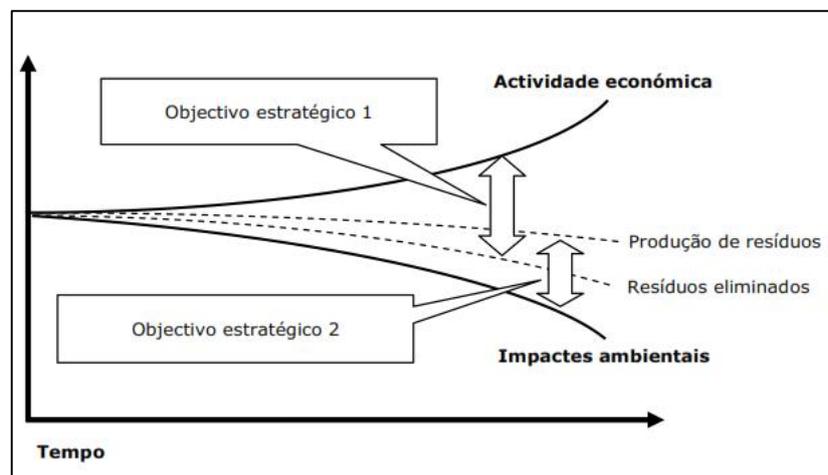


Figura 4 - Relação dos objetivos estratégicos.

in PNGR 2011 – 2020 [17]

### 3.5 Enquadramento e Relevância do iEcoSys no PNGR – 2011/2020

#### Objetivos Operacionais, PNGR – 2011/2020

Uma vez conhecidos os fatores transversais a toda esta problemática ambiental, relativa à gestão dos resíduos, conhecidas as estratégias e as metas, é preciso saber como concretizar os objetivos estratégicos do ponto de vista operacional. Desta forma, foram delineados objetivos operacionais para concretizar as metas na tabela a seguir apresentada (Tabela 3) [17].

Objectivo Operacional		Acções
OP1	Prevenir a produção de resíduos	<p>OP1.A1 - Promoção de acordos voluntários com sectores prioritários no sentido de fomentar a produção mais limpa e a concepção sustentável de produtos</p> <p>OP1.A2 - Promoção da comunicação/sensibilização para a prevenção da produção de resíduos</p> <p>OP1.A3 - Promoção de compras no sector público com critérios de sustentabilidade que previnam a produção de resíduos e fomentem a reutilização</p>
OP2	Promover o fecho dos ciclos dos materiais e o aproveitamento da energia em cascata	<p>OP2.A1 - Estabelecer e implementar um programa de acção para promover a procura de materiais passíveis de valorização</p> <p>OP2.A2 - Robustecer os sistemas de gestão de fluxos específicos aos quais se aplica a Responsabilidade Alargada do Produtor (RAP), numa óptica de criação de sinergias e avaliação da aplicação da RAP a fluxos emergentes</p> <p>OP2.A3 - Promover o estabelecimento de simbioses industriais</p> <p>OP2.A4 - Promover o estabelecimento de novas áreas industriais desenvolvidas com planos de racionalização de materiais e energia e a reabilitação de áreas industriais existentes</p>
OP3	Consolidar e otimizar a rede de gestão de resíduos	<p>OP3.A1 - Incentivar a proximidade da rede de recolha ao utilizador e a separação selectiva</p> <p>OP3.A2 - Potenciar sinergias de recolha e tratamento de resíduos numa lógica de complementaridade</p> <p>OP3.A3 - Uniformizar as regras de recolha e transporte para todos os operadores</p> <p>OP3.A4 - Promover a auto-suficiência e a competitividade dos operadores diminuindo a dependência externa</p>
OP4	Gerir e recuperar os passivos ambientais	<p>OP4.A1 - Implementar a estratégia de recuperação dos passivos ambientais</p> <p>OP4.A2 - Promover a monitorização e controlo dos locais pós-encerramento</p>
OP5	Fomentar a cidadania ambiental e o desempenho dos agentes	<p>OP5.A1 - Reforçar e apoiar as actividades de comunicação/sensibilização desenvolvidas pelos operadores de gestão e pelas entidades gestoras de fluxos específicos</p> <p>OP5.A2 - Instituir a atribuição de prémios anuais de desempenho</p> <p>OP5.A3 - Fomentar o envolvimento dos cidadãos e dos agentes no processo de tomada de decisão</p> <p>OP5.A4 - Promover a integração da gestão de resíduos e do uso sustentável dos recursos nos curricula educativos dos ensinos básicos e secundário</p>

Objectivo Operacional		Acções
OP6	Promover a formação e qualificação dos agentes	<p>OP6.A1 - Desenvolver um programa de formação no sentido do incremento das competências e da qualidade do serviço no sector de actividade dos resíduos</p> <p>OP6.A2 - Promover a implementação de sistemas de gestão ambiental, de qualidade e de higiene e segurança no trabalho</p> <p>OP6.A3 - Disseminar informação sobre boas práticas em sectores-chave produtores de resíduos</p>
OP7	Fomentar o conhecimento do sector numa lógica de ciclo de vida	<p>OP7.A1 - Promover a integração de sistemas de informação que garantam maior fiabilidade da informação e suporte aos processos de decisão e avaliação de tendências</p> <p>OP7.A2 - Desenvolver e actualizar numa base regular um sistema de indicadores sobre resíduos e fluxos de materiais e sua disponibilização pública</p> <p>OP7.A3 - Incentivar e apoiar a investigação e desenvolvimento no sector</p> <p>OP7.A4 - Promover e disseminar estudos de caracterização do sector</p>
OP8	Agilizar os processos administrativos e reforçar os mecanismos de controlo	<p>OP8.A1 - Optimizar o quadro legal e institucional</p> <p>OP8.A2 - Desmaterializar actos referentes à monitorização, avaliação e fiscalização das actividades de gestão de resíduos</p> <p>OP8.A3 - Reforçar as actividades de âmbito inspectivo e fiscalizador</p> <p>OP8.A4 - Reforçar as auditorias técnico-financeiras a OGR e sistemas de gestão de resíduos</p> <p>OP8.A5 - Criação de uma carteira de peritos qualificados para a certificação de processos ao nível da gestão de resíduos</p>
OP9	Adequar e potenciar a aplicação dos instrumentos económicos financeiros	<p>OP9.A1 - Promover o Mercado Organizado de Resíduos (MOR)</p> <p>OP9.A2 - Potenciar a eficácia da TGR</p> <p>OP9.A3 - Diferenciar as prestações financeiras relativas a Entidades Gestoras de fluxos de resíduos em função da reciclabilidade, perigosidade e impactes no ciclo de vida inerentes aos produtos colocados no mercado</p> <p>OP9.A4 - Articular as várias linhas de financiamento no sentido de optimizar as verbas disponíveis no sector de gestão de resíduos</p> <p>OP9.A5 - Avaliar e promover incentivos à recolha selectiva e reutilização de produtos</p>

Tabela 3 - Objectivos operacionais e respetivas acções.

in PNGR 2011 - 2020 [17]

## **Enquadramento do iEcoSys nos Objetivos Operacionais**

Cada um dos objetivos operacionais (OP) é definido e composto por um conjunto de ações delineadas com configurações concretas no sentido de atingir as metas a que se propõe.

Dos OP apresentados existe um alargado conjunto destes que a constituição das suas ações vão ao encontro do Sistema iEcoSys. A destacar com mais relevo estão os seguintes três OP [17]:

- OP5 - Fomentar a cidadania ambiental e o desempenho dos agentes, nas suas ações:
  - A1 - Reforçar e apoiar as atividades de comunicação e ou sensibilização desenvolvidas pelos operadores de gestão e pelas entidades gestoras de fluxos específicos
  - A2 - Instituir a atribuição de prémios anuais de desempenho.
  - A4 - Promover a integração da gestão de resíduos e do uso sustentável dos recursos nos currícula educativos dos ensinos básicos e secundário.
  
- OP7 - Fomentar o conhecimento do sector numa lógica de ciclo de vida, na sua ação:
  - A3 - Incentivar e apoiar a investigação e desenvolvimento no sector
  
- OP9 - Adequar e potenciar o uso de instrumentos económicos e financeiros, na sua ação:
  - A5 - Avaliar e promover incentivos à recolha seletiva e reutilização de produtos

in PNGR 2011 - 2020 [17]

Dos Objetivos Operacionais apresentados é de destacar o OP9, em particular na sua ação A5, que na sua totalidade vem ao encontro do sistema iEcoSys, visto que esta ação pretende atuar pela positiva onde é incentivada a recolha seletiva a fim de se valorizarem os resíduos e a reutilização dos produtos que estejam em fim de vida. O incentivo pode passar por descontos ou redução de tarifas, conceito PAYT.

Com o iEcosys pretendemos ir mais longe ao compensarmos diretamente o cidadão, detentor final e responsável por se desfazer do objeto, conceito GAYT, pelo produtor a quem é aplicado o conceito RAP, aplicando o conceito PAYT [23].

## 3.6 Smart Cities e Ambiente

### Definição de Sensor

Um sensor, tal como a palavra diz, é algo que sente ou que tem sensibilidade. No mundo da tecnologia um sensor é um componente de cariz eletrónico que tem a capacidade de *sentir* o meio envolvente e transpor essa informação para o mundo sob a forma de impulsos elétricos. Os sensores têm características que lhes permitem sentir as várias configurações do mundo real, tais como luz, distâncias, radiações, temperatura, humidade, peso, pressão atmosférica, som, imagem, movimento, entre outros, podendo ser utilizados para a leitura de algumas condições ou características ambientais.

Em termos técnicos, um sensor é um transdutor, ou seja, um componente que transforma um tipo de energia em outro tipo de energia. Ainda assim, um sensor é um tipo de transdutor muito específico, que transforma certas formas de energia em energia elétrica.

A vida no mundo moderno está de tal forma dependente dos sensores, que era impensável nos dias de hoje ter por exemplo, um motor de um carro com níveis de consumo rentáveis ou processos de fabrico de linhas de produção tão automatizadas. Desta forma, os sensores apresentam-se intrinsecamente ligados à satisfação das necessidades das sociedades atuais, indo da produção alimentar até à segurança aos mais variados níveis [11].

### IoT, Internet of Things

*Things*, em *IoT (Internet of Things)*, refere-se a uma extensa diversidade de dispositivos responsáveis pela monitorização através de sensores em ambientes, tais como, saúde, agricultura, águas costeiras, automóveis, entre outras.

A *Internet of Things (IoT)*, Internet das Coisas, é a ligação em rede dos objetos físicos ou “*Things*” embebidos com eletrónica, *software*, sensores e ligação em rede, o que permite capacitar as “*Things*” para recolha e troca dados. Desta forma é possível monitorizar os objetos e controlar os mesmos remotamente.

Este facto possibilita criar oportunidades de integração entre o mundo físico e os sistemas computadorizados com resultados de eficiência improváveis, precisão e um grande benefício económico. Cada “*Coisa*” está univocamente identificada

através do seu sistema computadorizado e operacional para interagir com a infraestrutura de internet existente.

As “*Things*” recolhem informação útil que juntamente com a ajuda de várias tecnologias flui automaticamente entre os vários dispositivos. Um dos exemplos mais evidentes do mercado é a aplicação dos sensores e atuadores às chamadas casas inteligentes.

Até 2020 estão previstos à volta de 50 biliões de objetos ligados em rede, permitindo tornar as *Smart Cities* uma realidade [17].

Entretanto a mudança e a evolução não ficaram por aqui, de tal forma que o paradigma atual progrediu para o *IoE*, *Internet of Everything* [11].

### **IoE, Internet of Everything**

A *Internet of Everything (IoE)* junta pessoas, dados, processos e coisas para estabelecer ligações em rede, mais relevantes e valiosas do que nunca, tornando informação em ações, criando novas competências, experiências mais ricas e incomuns, criando oportunidades económicas de negócio para pessoas e países [11].

### **Smart Cities**

O século XXI é o século das cidades. Prevê-se que em 2050 70% da população viva em cidades, pois atualmente 60% já está a viver em volta de núcleos urbanos, ultrapassando já a população rural desde 2007 [17].

As cidades constituem verdadeiros ecossistemas (Figura 5) onde as pessoas vivem e trabalham, tendo vindo a transformar-se no centro de uma sociedade baseada nos serviços e globalmente integrada, com mais poder económico, político e tecnológico do que nunca. Constituindo-se como grandes centros de consumo de recursos, calcula-se que sejam responsáveis pelo consumo de 75% da energia global e por gerar 80% dos gases que contribuem para o efeito de estufa.

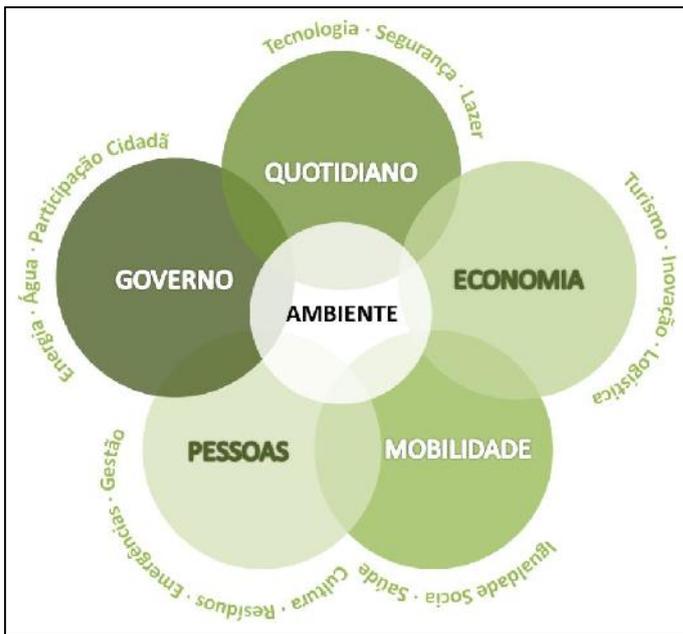


Figura 5 - Dinâmica das smart cities e o ambiente.

in PNGR 2011 - 2020 [17]

As cidades são fundamentais para a economia global, uma vez que aglomeram grande parte da população mais habilitada, criativa e empreendedora, apresentando-se como uma relevante origem de saber, pilar de grandes redes empresariais e de investimento que originam e dilatam a inovação, levando ao investimento e a economias globais.

A nível Europeu, existem vários desafios que a Europa tem de enfrentar até 2020 relativamente às cidades, de ordem social, tecnológica ou económica por forma a construir uma sociedade avançada, solidária, inclusiva e sustentável.

Sendo as cidades um núcleo central de desenvolvimento tecnológico e formadas pela sociedade de informação, têm um potencial de oportunidades infinito e muitas vezes por investigar. Os desafios não passam só por otimizar os serviços públicos ou pela minimização dos custos, passa também pelo impacto que têm nos cidadãos e na economia local.

Face ao exposto, a qualidade de vida dos cidadãos nas cidades do presente e especialmente as do futuro, passa pela resolução dos problemas ambientais. A necessidade de gestão dos recursos energéticos, alimentares, transportes e dos resíduos vai obrigar a uma elevada eficiência para que seja possível assegurar uma boa qualidade de vida aos seus cidadãos.

Smart City ou Cidade Inteligente é um conceito que surgiu na década de 90. Nesta década os problemas inerentes a este conceito passavam por resolver questões de sustentabilidade relacionados com eficiência energética e com redução das emissões de carbono.

O conceito SMART tomou outra dimensão, pois a gestão eficaz dos recursos disponíveis traz grandes lutas a nível tecnológico e social. Território, tecnologia e inovação são ideias que precisam de uma meditação abrangente e só assim será possível desenvolver as suas atividades de forma sustentável e integradora, em que todos ganham, desde a eficiência da cidade, como a qualidade de vida dos cidadãos.

A cidade tem de passar por uma introspeção para, numa primeira fase, atingir o patamar de inteligente [21]:

- Ser conhecedora das suas capacidades,
- Ser conhecedora do seu território,
- Ser conhecedora dos seus pontos fortes,
- Ser conhecedora dos seus recursos,
- Ser conhecedora do tipo de cidade que quer ser dentro de 10 anos,
- Ser conhecedora do tipo de serviços que terá de prestar aos seus habitantes,
- Ser conhecedora de tecnologias mais sustentáveis e mais avançadas, para melhorar a qualidade de vida dos seus cidadãos e alcançar o desenvolvimento económico.

Cidades inteligentes não é significado de cidades futuristas e altamente tecnológicas.

Uma cidade é denominada de inteligente quando [21]:

- Conseguir adotar adequadamente sistemas intensivos em tecnologia,
- Desenvolve a capacidade de criar, recolher, processar e transformar a informação para tornar os seus processos e serviços melhores e mais eficientes,
- Conseguir que a tecnologia não seja o fim mas antes o meio para fazer chegar os serviços aos cidadãos,
- Permite a comunicação interativa entre o cidadão e a administração pública,
- Conseguir uma oportunidade de crescimento sustentável.

Em suma, os cidadãos são o centro de tudo e nenhuma cidade pode atingir o estatuto de inteligente se o seu objetivo não for dar aos seus cidadãos serviços tidos como úteis, de maneira a satisfazer as suas necessidades e desenvolverem o seu bem-estar e qualidade de vida. Exemplo disso são os projetos desenvolvidos em grandes cidades com sistemas inteligentes, como o projeto de gestão avançada da água *SmartBay* em Galway, a iniciativa *Wired City* de Songdo ou o sistema de transportes *eSymphony* em Singapura [17].

O progresso para chegar a SMART City requer em primeiro lugar liderança e em segundo lugar visão estratégica de futuro. Depois, para que o processo de evolução tenha êxito, é preciso tempo, visto que este caminho passa por uma simbiose entre o público e o privado. Por fim aos dados recolhidos da habitual atividade da cidade é necessário transformar os mesmos em conhecimento com importância para ser partilhado e contribua para a eficiência com retorno para os cidadãos. A este processo chama-se aplicar inteligência.

Como a evolução é uma constante na Europa, já existe um conceito de “comunidades e envolventes inteligentes”, cujo objetivo é fazer desaparecer as barreiras físicas e a “envolvente inteligente” pode passar por um bairro de uma cidade grande, uma vila ou uma região onde os múltiplos focos populacionais cooperam para aperfeiçoar os serviços e os oferecem de forma distribuída para poder favorecer as economias de escala [21].

*“A filosofia das SMART Cities centra-se em unir, mediante uma adequada planificação, todos estes conceitos com a finalidade de transformar as cidades e os territórios em espaços sustentáveis, inovadores e eficientes, nos quais o cidadão deve ser o eixo da mudança e o principal beneficiado do novo paradigma urbano, assegurando um nível adequado da qualidade de vida. O futuro das cidades passa por um modelo de projeto colaborativo comum. Soluções inteligentes para cidadãos inteligentes em territórios inteligentes e sustentáveis.”*

**in Guia Smart Cities [17]**

## Capítulo 4 - Tecnologia Arduino

O sistema projetado, o *iEcoSys*, usa tecnologia *RFID*, amplamente usada em grande escala no mercado para os mais variados fins [5] [6]. Outra das inovações introduzidas é o uso da tecnologia *ZigBee*, recorrendo a módulos *XBee*, que tem apenas custos de implementação, associados a uma plataforma de prototipagem eletrónica de *hardware* livre, projetada com um microcontrolador *Atmel AVR*, vulgo *Arduino*. O *Arduino* é o coração deste projeto [15].

### 4.1 Arduino

O *Arduino* (Figura 6) é uma plataforma de computação de código aberto, fácil de usar, extremamente poderosa e que conquistou um espaço junto dos entusiastas, académicos e profissionais, marcando uma posição que dificilmente perderá, mesmo tendo em conta o desenvolvimento tecnológico constante e com um crescimento exponencial.

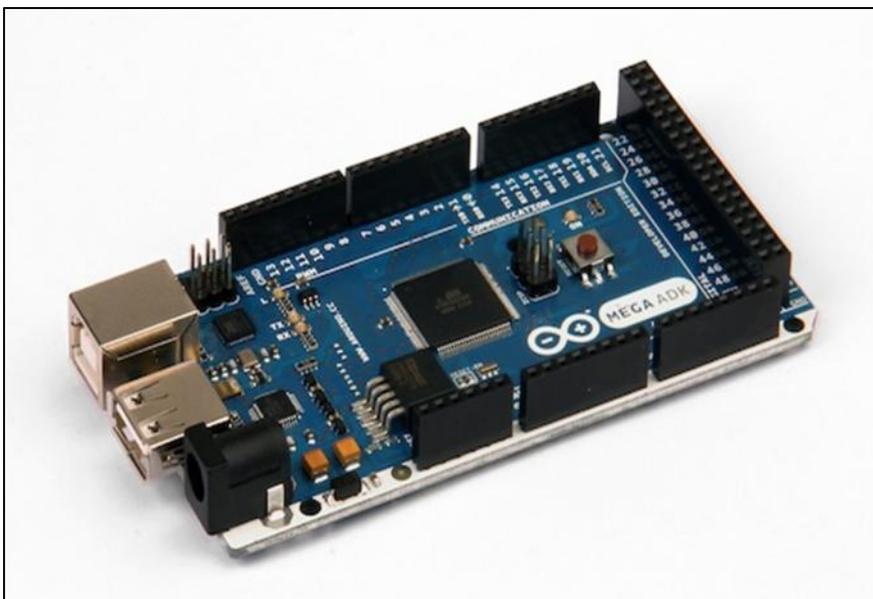


Figura 6 - Arduino Mega ADK.

microcontrolador.

O *Arduino* captura dados do meio ambiente através de uma série de portas às quais estão ligados sensores que monitorizam variações no meio envolvente. Trata essa informação convertendo-a para grandezas digitais e a partir daí executa

Esta plataforma é baseada numa simples placa com um Microcontrolador incorporado e um ambiente para desenvolver o código a ser executado no referido

comandos programados, permitindo assim controlar dispositivos, motores entre outros.

O *hardware* do *Arduino* é de código aberto e as placas podem ser montadas a um custo baixíssimo ou adquiridas [15].

#### 4.1.1 Porquê o Arduino?

Apesar de existirem diversas plataformas com microcontroladores disponíveis no mercado, como por exemplo a *Parallax Basic Stamp*, a *Netmedia's*, a *Phidgets* e a *MIT's Handyboard*, a plataforma *Arduino* foi escolhida como cérebro deste projeto devido à sua grande popularidade no meio universitário e as características que lhe são intrínsecas e que lhe deram popularidade junto dos docentes e alunos no ambiente académico.

O baixo custo desta plataforma é o primeiro fator a favorecer o acesso por parte dos intervenientes do meio académico. O acesso dos alunos permite aceder a esta plataforma e explorá-la de forma a obter o seu melhor. Os módulos *Arduino* pré-montados custam em grande parte menos de vinte euros e a versão mais barata do módulo *Arduino* pode ser montada à mão.

A seguir é analisado o *Arduino Atmega256 ADK*, escolhido para o desenvolvimento do *iEcoPonto* devida às características que possui, em particular o número de portas serie que disponibiliza [15].

#### 4.1.2 O Arduino Atmega256 ADK

Existem no mercado um número variado de versões da plataforma *Arduino*. De entre as várias versões podemos destacar o *Mini*, o *LilyPad*, o *Nano*, o *Mega*, o *Severino*, o *Roboduíno*, o *Duemilanove*, o *Uno*, entre tantas outras, sendo que todas elas, apesar de serem uma versão diferente, usufruem de um microcontrolador do fabricante *Atmel®* integrado na plataforma.

Uma das plataformas é o *Arduino Mega ADK*, com um microcontrolador alicerçado no *ATmega2560*. Em traços gerais o *Arduino Mega ADK* oferece 54 portas digitais de *input/output*, das quais 15 podem ser usadas para *Output PWM*. As portas analógicas de *input* desta plataforma são 16, com 4 *UART*, um oscilador de cristal a 16 *Mhz* e uma ligação de uma porta *USB*.

Para além das características anteriormente apresentadas, dispõe de uma interface anfitriã que permite a ligação de telemóveis com Sistemas *Android* [15].

#### 4.1.3 O Microcontrolador Atmega256

Microcontroller ATmega2560	
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
USB Host	Chip MAX3421E

O microcontrolador é o principal elemento num projeto de componentes com eletrónica. Em suma, é um computador em miniatura num *chip*, incluindo um processador, memória e periféricos de entrada e saída [15].

Tabela 4 - Características Atmega 256 ADK.

in [arduino.cc](http://arduino.cc) [15]

#### 4.1.4 Memória

O *Arduino MEGA ADK* dispõe de 256 KB de memória *flash* para armazenamento do código. Dos 256 Kb de memória disponíveis, 8 Kb estão reservados para o *bootloader*. Dispõe ainda de 8 Kb de *SRAM* e 4 Kb de *EEPROM* na qual pode ser feita a leitura e escrita de dados com a *librarie EEPROM* [15].

#### 4.1.5 Entradas e Saídas do Microcontrolador

O *Arduino MEGA ADK* possui 50 portas digitais que podem ser usadas como input ou output recorrendo às funções de programação *pinMode()*, *digitalWrite()*, e *digitalRead()*. Cada porta funciona a 5V e pode fornecer ou receber no máximo 40mA, dispondo de uma resistência que poderá ir até aos 50 K . Algumas das portas têm contudo funções específicas.

A transmissão e receção de dados *TTL* das *portas serie* são feitas também nas portas digitais com funções próprias, sendo exatamente 8 portas. De salientar

ainda que a porta 0 e a porta 1 estão também ligadas ao *ATmega8U2* chip que faz a ligação *USB to Serial TTL*

Existem outras 6 portas com funções específicas, os *interrupts*, ou seja, estas portas digitais podem ser configuradas para disparar um *interrupt*, por exemplo quando existe uma alteração do valor de uma variável usando a função *attachInterrupt()*.

Outra funcionalidade disponibilizada pelas plataformas *Arduino* é o *PWM – Pulse With Modulation*. Para esta funcionalidade dispomos de 14 portas digitais que fornecem um *PWM* de 8 bit de output com a função *analogWrite()*.

O *SPI* é um protocolo de comunicação também suportado pelo *Mega ADK*, utilizando as portas digitais 50 (*MISO*), 51 (*MOSI*), 52 (*SCK*) e 53 (*SS*), respetivamente. O controlo deste protocolo requer o uso da *SPI library*. Existe também o *USB Host, MAX3421E*, que comunica com o *Arduino* através do *SPI bus* e usa as portas digitais 7 (*RST*), 50 (*MISO*), 51 (*MOSI*) e 52 (*SCK*). Existe ainda o protocolo de comunicação *TWI*, 20 (*SDA*) e 21 (*SCL*), igualmente suportado pelo *Mega ADK*, sustentado pela *Wire library*.

Por fim, no que toca a portas digitais, temos ainda o *Led 13* que está conectado á porta digital 13 e que se liga quando o valor da porta está a *High* e desliga quando o valor da porta está a *Low*.

Na plataforma *Mega ADK* as portas analógicas de *input* são 16, com uma resolução de 10 *bits*, ou seja 1024 valores. Por defeito, os valores medidos devem ir no limite até 5 V, sendo possível alterar esse valor recorrendo á função *analogReference()* e á porta *AREF*.

Por último mas não menos importante, o *Reset* quando ativado a *LOW* faz o *reset* do microcontrolador, sendo também possível usar o botão físico existente na placa [15].

#### 4.1.6 Alimentação

O *Arduino MEGA ADK* pode ser alimentado através da ligação *USB* ou através de uma fonte de energia externa. Similarmente, é possível ligar a partir de uma fonte externa recorrendo à ligação *jack* existente na plataforma. A placa da plataforma *Arduino* pode funcionar a partir de uma fonte externa de energia que

pode variar numa gama entre os 5,5V e os 16V. De qualquer modo, uma alimentação inferior a 7V é possível mas a plataforma poderá ficar instável, no caso de se usar uma alimentação superior a 12V o regulador de tensão poderá sobreaquecer e danificar a placa, sendo a recomendação ótima ideal entre os 7V e os 12V. O regulador da fonte de tensão externa da placa pode fornecer no limite até 1500mA, sendo 750mA cedidos ao *Mega ADK* e os outros 750mA são alocados para os atuadores e sensores conectados à placa [15].

#### 4.1.7 Bootloader

O *bootloader* é um pequeno programa que é executado na memória *flash* do microcontrolador AVR e é inicializado assim que a plataforma é alimentada. O *Arduino* precisa de um *bootloader* para transferir o código concebido através de uma comunicação serial/USB. A função do *bootloader* é ler dados do programa da *UART*, *Universal Asynchronous Receiver/Transmitter* e escrevê-los na *flash* interna do microcontrolador. O *bootloader* pode ser carregado usando um programador ISP para microcontroladores AVR.

Para o *Arduino*, o *ISP*, *In-System Programmer*, é usado para carregar o *bootloader*. Na inicialização, o *bootloader* é executado e este por sua vez comunica com a porta série (*Pinos TX e RX*), permitindo assim desta forma que o *Arduino* seja programado [15].

#### 4.1.8 In-Circuit System Programming

O *ICSP*, *In-Circuit System Programming*, é uma tecnologia que permite que circuitos integrados que já estão montados e corretamente soldados nas placas sejam programados. No *Arduino*, o *ICSP* permite que os microcontroladores sejam programados diretamente com instruções AVR sem a necessidade de se utilizar o *IDE* do *Arduino*. O *ICSP* permite *upgrades* quando são disponibilizadas atualizações no *bootloader*, ou seja, o *ICSP* pode ser utilizado para programar o *Arduino* ou para usar o *Arduino* como um programador *ISP*, *In-System Programming* [15].

#### 4.1.9 Clock, Oscilador de Cristal

O oscilador de cristal do *Arduino MEGA ADK* opera a uma frequência de 16 MHz. Isto significa que o Arduino dispõe de um cristal que faz 16 milhões de ciclos por segundo. O oscilador de cristal é importante em qualquer equipamento que execute aplicações que dependam do tempo para serem bem executadas, nomeadamente microprocessadores, *webcams*, computadores, entre outros.

A precisão do cristal depende das suas especificações. No Arduino, o oscilador tem uma precisão de 100 ppm, *Pulse Position Modulation*, que em termos práticos significa que o cristal tem uma margem de erro de 100 ciclos a cada 1 milhão de ciclos. Logo, o erro máximo ou desvio a que o cristal está sujeito é da ordem dos 30s no período de 1 ano [15].

#### 4.2 Software

O código do ambiente de desenvolvimento (Figura 7) é aberto e a sua utilização é regida sob a licença GPL (*General Public License*), as *libraries* do microcontrolador C/C++ sob licença LGPL (*Library General Public License*) e os

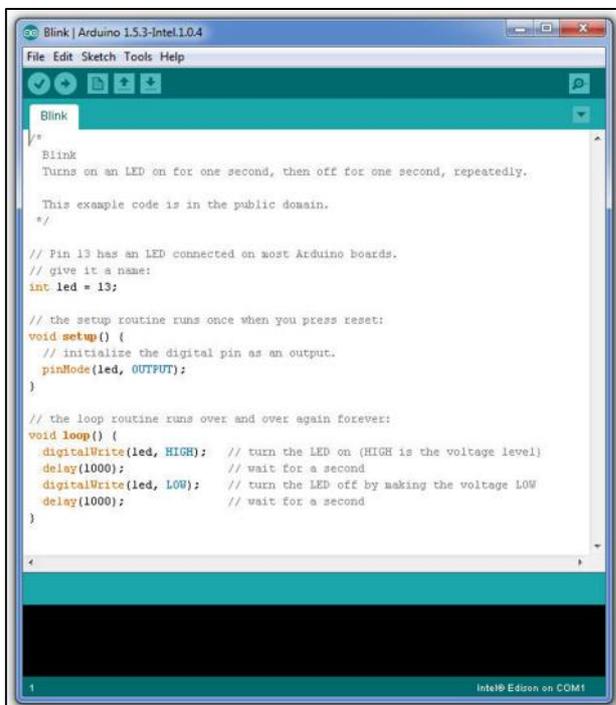


Figura 7 - Exemplo de uma janela do IDE Arduino.

esquemas e arquivos CAD (*Computer-aided design*) sob a licença *Creative Commons Attribution Share-Alike*. Um dos grandes pontos fortes da plataforma Arduino é o fato de ela ser *Open Source*.

O software do Arduino corre em sistemas operativos *Microsoft Windows*, *Apple Macintosh OSX* e *Linux*, estando a maioria dos sistemas dos microcontroladores limitada ao sistema operativo *Microsoft Windows*, sendo a

portabilidade outra mais-valia.

Outra característica importante do Arduino é que é possível programar o microcontrolador a partir do computador via USB (*Universal Serial Bus*).

O Arduino, uma vez desligado, guarda o último programa carregado em memória, sem necessidade de que o mesmo seja novamente carregado através do computador ou seja, só há necessidade de ligar o Arduino ao computador quando for necessário carregar um novo programa [15].

## 4.2 Programação e IDE

O Arduino é conhecido principalmente pelo seu *hardware*, mas é necessário software para programar o Hardware. Tanto o *software* como o hardware são chamados de “Arduino”. A conjugação de ambos permite a criação de projetos usando sensores e atuadores de forma a interagir com o mundo real. O *software* é *open source* e multiplataforma.

O Software para o microcontrolador foi desenvolvido com base numa linguagem de programação, que é uma variante da Linguagem C para Arduino, a partir de código aberto já existente para o efeito (Figura 8).

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```

Figura 8 - Exemplo de um programa que recorre a uma libreria.

O ambiente de programação do Arduino é fácil de usar, mesmo para principiantes e suficientemente flexível para que utilizadores profissionais possam aproveitá-lo ao máximo. Os programas de *software* desenvolvidos pelo utilizador são chamados de sketches.

Estes programas são criados com recurso ao Arduino *Integrated Development Environment (IDE)*. O *IDE* possibilita a escrita e edição do código, convertendo o mesmo em instruções que o *hardware* do Arduino compreende.

Ao processo de envio do código escrito pelo utilizador para o Arduino é chamado de *uploading*. Para o *uploading* ser feito com sucesso é necessário selecionar no *IDE* a porta *COM* à qual o Arduino está ligado e da mesma forma selecionar a plataforma Arduino com que se está a trabalhar.

Determinados projetos necessitam do recurso a *libraries*, normalmente disponibilizadas pelos fabricantes, que servem de interface com o Hardware que pretendemos acoplar à placa Arduino. Da mesma forma, é exequível a construção de uma *librarie* para uso próprio, recorrendo a outras *libraries*, se necessário.

A monitorização e *debug* do código é muito útil no desenvolvimento da aplicação, na medida que possibilita ter noção e *feedback* do progresso e dos resultados, por forma a se fazer uma análise crítica dos dados obtidos. O *IDE* do Arduino disponibiliza um Monitor da Porta Serie que permite visualizar a informação recorrendo à função *Serial.println()* [15].

## Capítulo 5 - Sistema iEcoSys

### 5. Descrição do Sistema

No Sistema iEcoSys o fluxo de informação começa com o *cidadão* que adquire *iSacos* ou *iBags* em inglês, que permitem a sua identificação no sistema no ato do depósito. O utilizador apenas tem de identificar o *iSaco* de lixo que está a depositar, junto do *iEcoPonto*, sendo este constituído por um tradicional ecoponto com um sistema de identificação e pesagem de *iSacos*.

Após o depósito, o sistema regista na base de dados que tipo de lixo foi depositado, o respetivo peso, data e hora.

Cada *cidadão* terá uma conta corrente onde será creditado o valor a receber pelos resíduos, em função do seu peso, incentivando assim a separação do lixo por parte dos cidadãos (Figura 9).

Como contrapartida pedagógica de incentivo à reciclagem efetuada pelo cidadão, é proposto um desconto percentual a definir na fatura da Água, em função da quantidade de lixo depositado corretamente, bem como do valor comercial do mesmo.

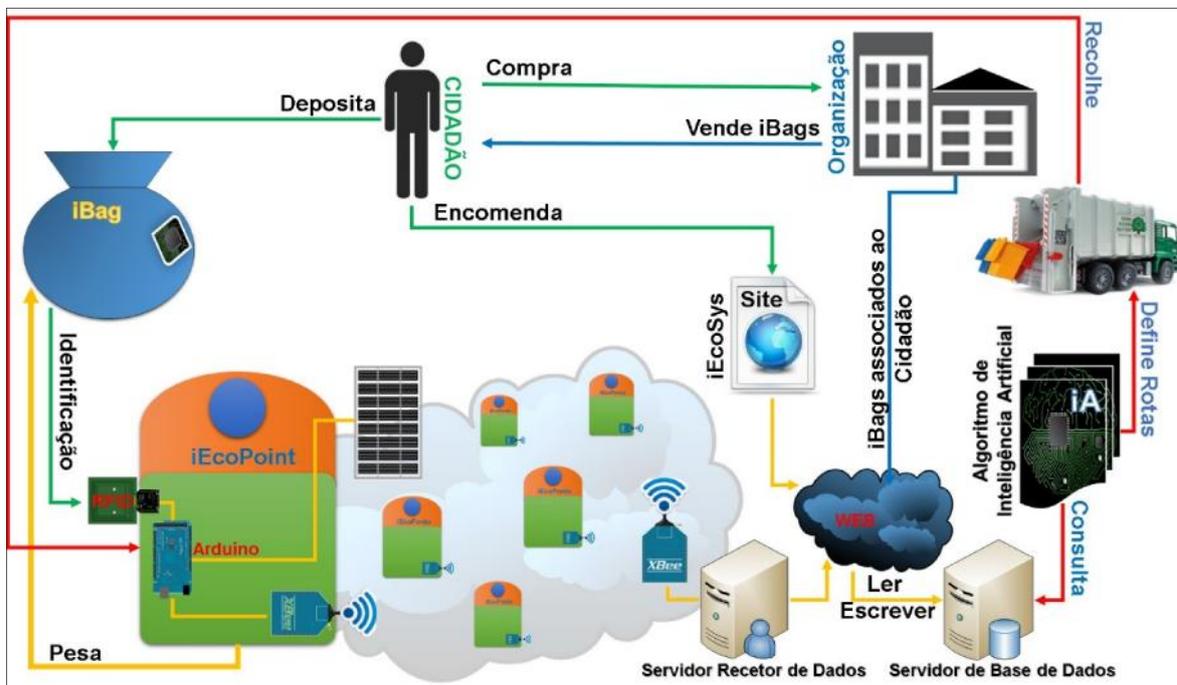


Figura 9 - Algoritmo geral da arquitetura do Sistema iEcoSys.

O Sistema *IEcoSys*, permite a identificação do *cidadão* e do tipo de resíduos depositados no *iEcoPonto*, *abaixo descrito*, possibilitando também saber o local onde foi feito o depósito e a que horas foi feito devido aos *iEcoPontos* e os *iBags*.

*Exemplo: A Maria depositou 2,2 kg de papel no ecoponto X2 às 18h30.*

Uma vez implementado, o Sistema *IEcoSys* permite extrair outro tipo de informações que uma Base de Dados devidamente estruturada pode permitir, tal como a quantidade de resíduos depositados pelo cidadão num determinado período de tempo, que *tipo de resíduos* depositou em maior quantidade e qual o *iEcoPonto* mais utilizado para fazer depósitos. Do lado da gestão do processo, torna possível saber quem fez a recolha daquele ecoponto, data e hora, estimativas de enchimento dos ecopontos e informação se os mesmos já estão ou não cheios, ou seja, todos as *Queries* que uma Base de Dados possa permitir, envolvendo os *atores* que de alguma forma integram o Sistema *iEcoSys*, como o *cidadão*, o *iBag*, o *iEcoPonto*, os *funcionários de recolha*, os *gestores do processo*, entre outros.

## 5.1 Descrição do Processo

A melhor forma de se explicar o sistema, passa pela explicação dos vários estádios do processo que realiza o Sistema *iEcoSys*, de forma a explicar *à posteriori*, em pormenor, cada uma das etapas do processo e as respetivas tecnologias utilizadas.

### 5.1.1 Registo do cidadão no Sistema

O primeiro passo no processo passa pelo registo do *cidadão* no Sistema *iEcoSys* (Figura 10). O registo pode ser feito via web no *site* pelo próprio *cidadão* ou presencialmente na *organização* que gere o processo. O registo pode ser feito por um funcionário que faça parte da entidade gestora, usando uma aplicação que corre no sistema informático local e desenvolvida para o efeito. Este registo permite a identificação do *cidadão*, bem como a criação de uma conta corrente.

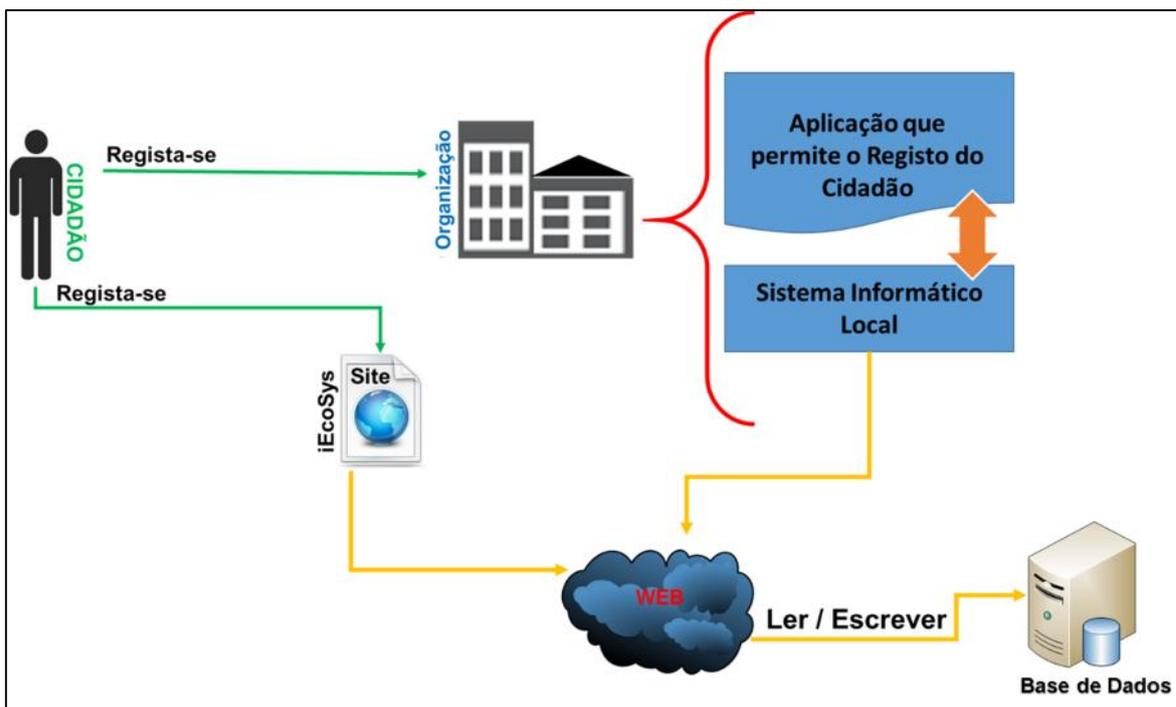


Figura 10 - Registo do cidadão no Sistema.

### 5.1.2 Compra, Venda e Encomenda de iBags

O segundo passo do processo passa pela compra de *iBags*, pelo *cidadão*. A compra dos *iBags* é feita na *organização* (Figura11) que gere o processo ou, em alternativa, podem ser encomendados pelo *site*.

No ato da compra os *iBags*, que têm uma identificação única, são associados ao *cidadão*. A associação da identificação única dos *iBag*, é feita pela aplicação existente na organização e desenvolvida para o efeito.

No caso de a encomenda ser feita via web, no ato da encomenda os *iBags* são automaticamente associados ao *cidadão*, uma vez que estes fazem a encomenda a partir da sua conta pessoal.

O pagamento dos *iBags* pode ser feito descontando da conta corrente do *cidadão*.

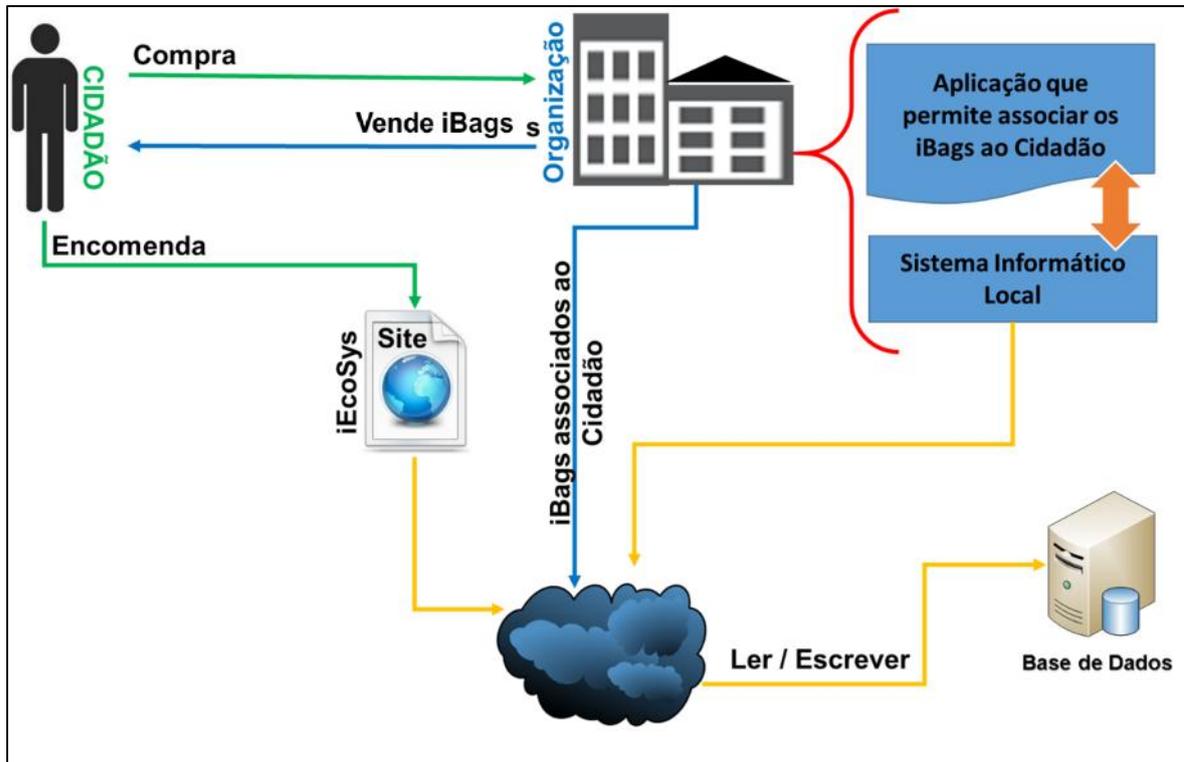


Figura 11 - Compra, venda e encomenda de iBags.

### 5.1.3 O iBag

O *iBag* (Figura 12) é um componente chave neste processo. O *iBag* é criado com a união de uma *tag RFID* a um saco de lixo comum. As tags RFID dispõem de identificadores únicos que permitem associar os *iBags* ao *cidadão*, como foi descrito no ponto anterior. É possível ao cidadão ter os seus próprios *iBags*, adquirindo apenas as *tags RFID* para colar nos sacos comuns de supermercados ou outros de que disponha.

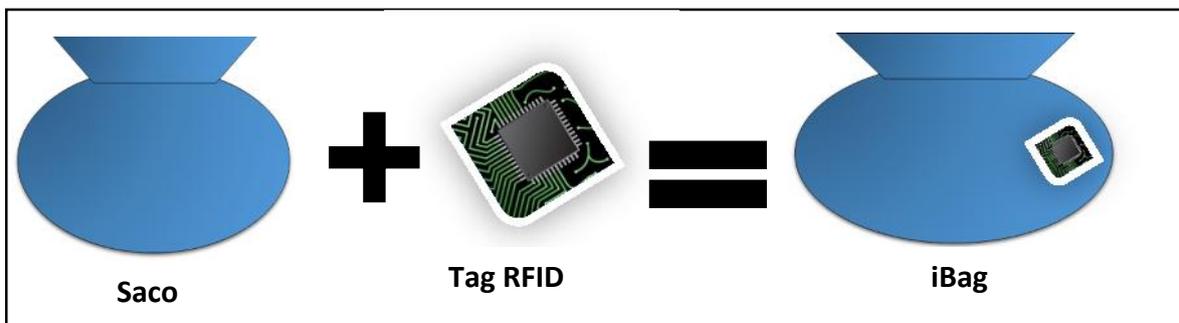


Figura 12 – O iBag.

Outra característica importante associada ao *iBag* é o facto de ser possível guardar informação digital no mesmo, devido à capacidade das *tags RFID* armazenarem informação. Desta forma, foi definido digitalmente na *tag RFID* o *tipo de resíduos* que um *iBag* deve conter para que no ato de depósito o *microcontrolador*, ao receber a informação do *iBag* (*ID da Tag + Tipo de Resíduos*), através do *leitor RFID*, possa acionar a porta do *iEcoPonto* correspondente ao *tipo de resíduos* que está a ser depositado.

Para que o *cidadão* identifique de forma intuitiva o *tipo de resíduos* que deve colocar num *iBag*, o *iBag* ou saco ou a *tag RFID* têm a cor correspondente ao *tipo de resíduos* a que estão associados. As cores dos *iBag* com a *tag RFID* são as cores globalmente padronizadas para cada *tipo de resíduo*.

Esta funcionalidade evita que de cada vez que é feito um depósito, o *iEcoPonto* tenha de consultar a base de dados para saber o *tipo de resíduos* a que pertence o *iBag* que se pretende depositar, evitando assim perda de tempo e sobrecarga da rede. No caso de haver uma falha de rede, contínua a ser possível, com esta funcionalidade, identificar o *tipo de resíduos* que se pretende depositar.

As características do *iBag* tornam possível anular um depósito, cujo *tipo de resíduos* que o *iBag* contém não esteja de acordo com o *tipo de resíduos* que o *iBag* deveria ter (Figura 13).

#### 5.1.4 Depósito

A fase de depósito (Figura 13) começa com a identificação do *iBag* junto do leitor de *RFID* existente no *iEcoPonto*. A abertura do *iEcoPonto* correspondente ao tipo de resíduos do *iBag* permite ao cidadão depositar o *iBag*. A porta fecha, o *iBag* é pesado e a informação é enviada através da placa de rede *wireless* para o Servidor de Receção de Dados através da rede *XBee*.

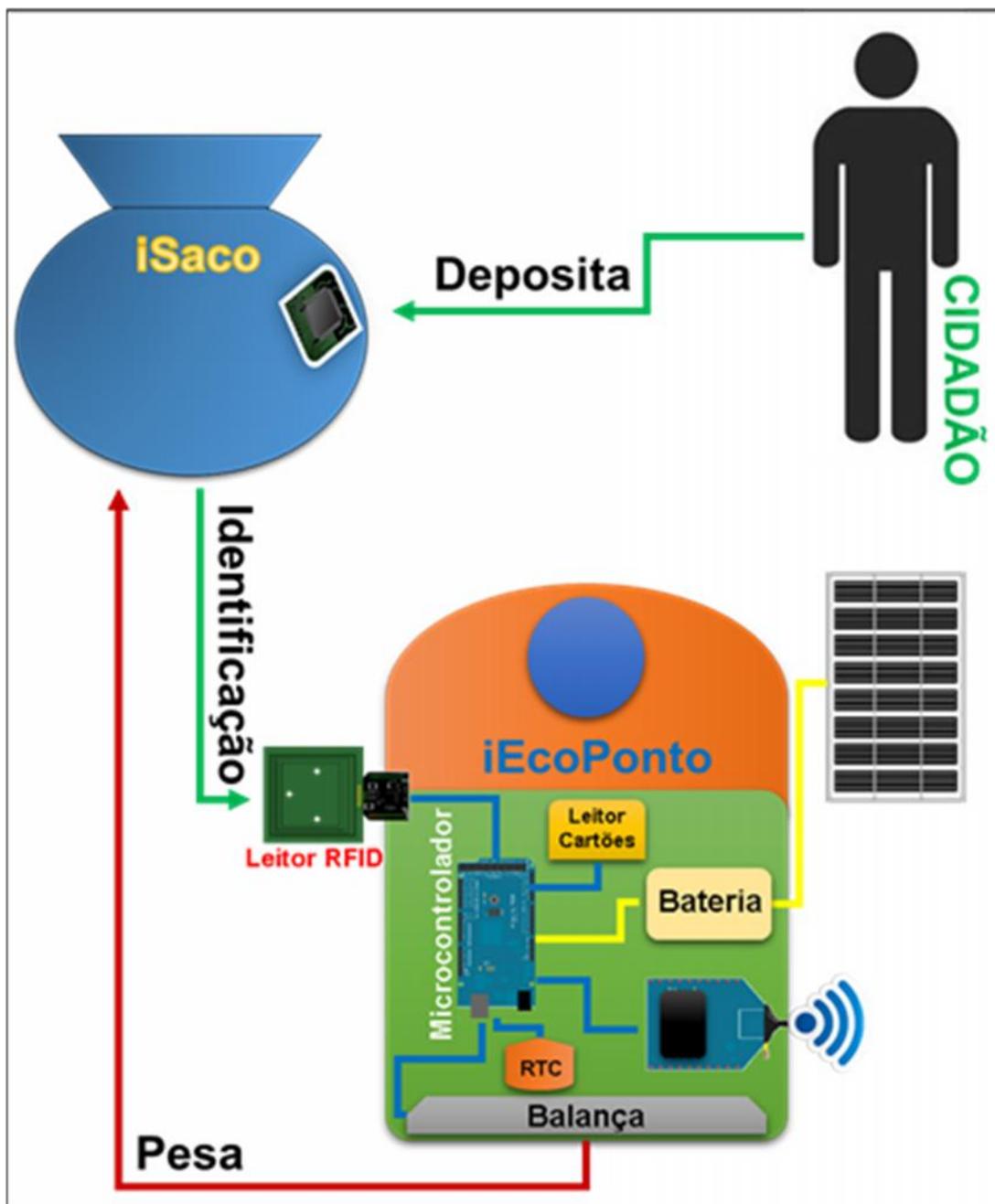


Figura 13 - Depósito no *iEcoponto*.

## 5.2 O iEcoPonto

O *iEcoPonto* também é um componente chave neste processo. O *iEcoPonto*, (Figura 16), é criado com a ligação de um *Ecoponto* independentemente da natureza do mesmo, subterrâneo ou superficial, com um sistema informático/eletrónico, de leitura, processamento e envio de dados (Figura 14 e Figura 15).

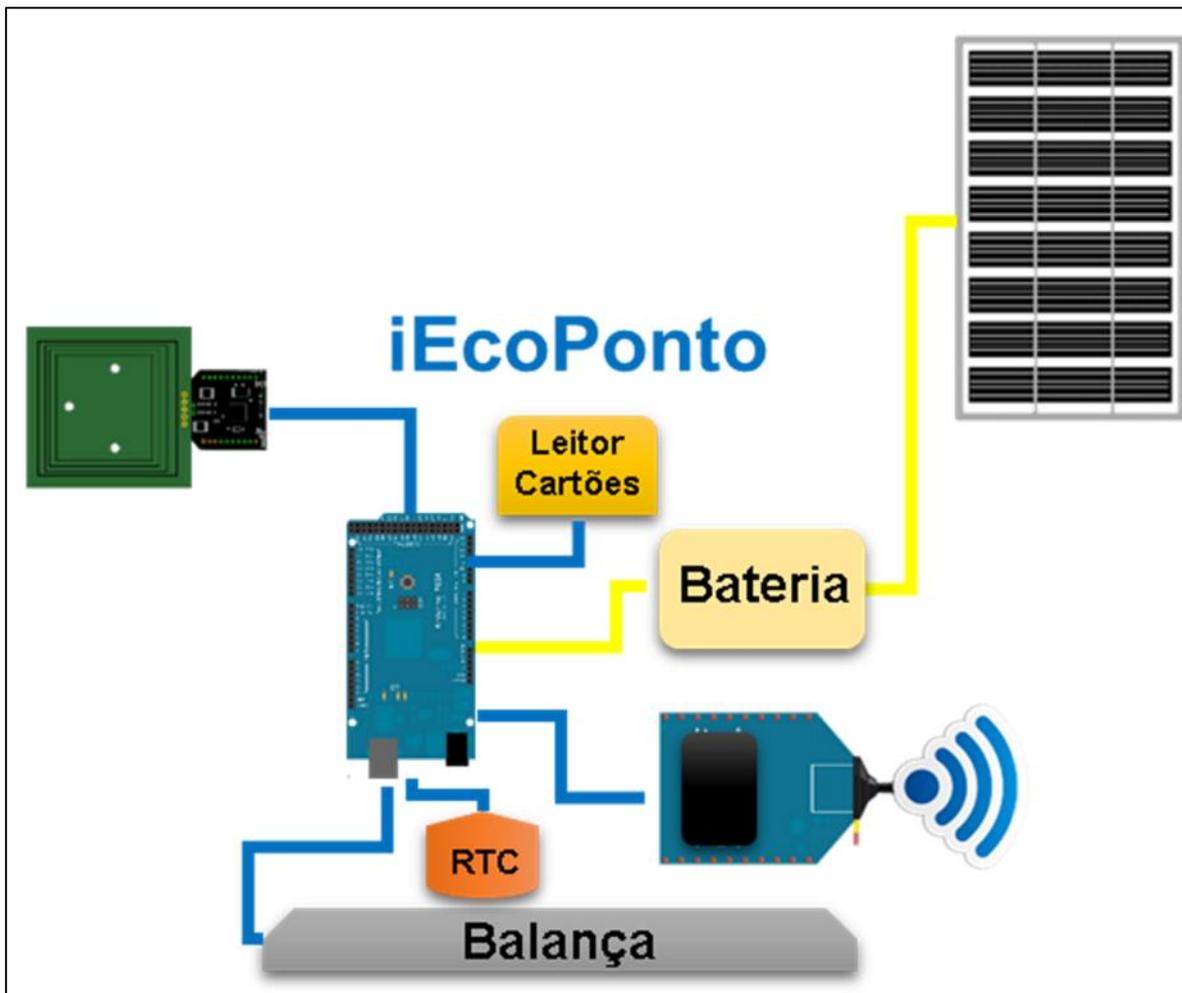


Figura 14 - Sistema informático / eletrónico de leitura processamento e envio de dados do *iEcoPonto*..

Este sistema é constituído por:

- **Leitor RFID** – O leitor RFID permite ler a identificação do *iBag*, e o *tipo de resíduos*, enviando os mesmos para o microcontrolador.
- **Microcontrolador** – O microcontrolador é programado através de um computador com um conjunto de instruções que permitem a recolha e envio

de dados relativos ao processo, tais como *ID* (Identificação) da *tag* e peso dos *resíduos*.

Outra das funcionalidades do microcontrolador é controlar a abertura e o fecho das portas do *iEcoPonto*, em função do tipo de resíduos que o cidadão quer depositar num determinado momento. Os sinais visuais que indicam a abertura e fecho das portas são efetuados por *Leds* (*Light Emitting Diode*), também controlados pelo microcontrolador. O controlo de componentes que possam ser adicionados ao sistema para o *input* ou *output* de informação é sempre feito pelo microcontrolador, devidamente programado para o efeito.

- **Balança** – A balança permite efetuar a leitura do peso dos resíduos para que este seja contabilizado. A balança envia a informação via porta série para o microcontrolador.

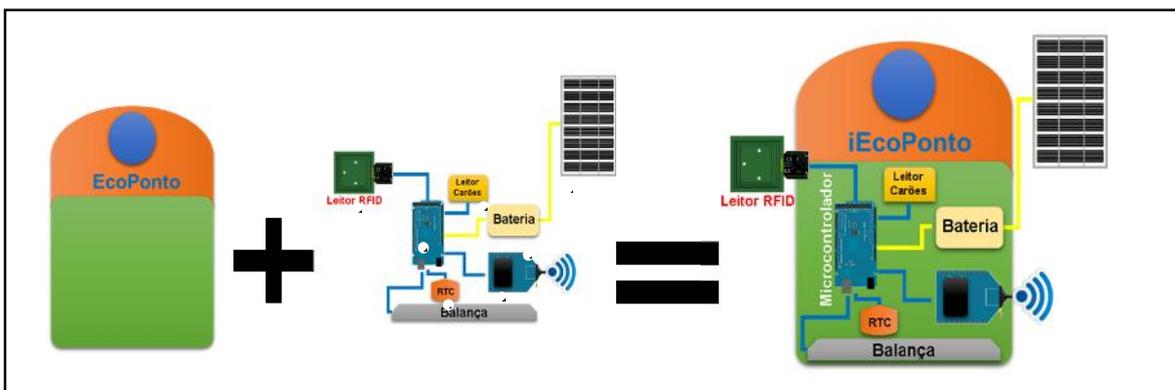


Figura 15 – Obtenção do *iEcoPonto*.

- **RTC – Real Time Clock** – O Real Time Clock fornece ao microcontrolador a data e hora, quando solicitadas. Permite assim ao microcontrolador associar, ao identificador do *iBag* depositado, a data e hora de depósito. No entanto, esta informação só é guardada no caso de o microcontrolador não receber uma resposta do *Servidor de Receção de Dados* de que o *iBag* não foi devidamente registado.
- **Leitor de Cartões e Cartão de Memória** – O *leitor de cartões* e o *cartão de memória*, são usados para na eventualidade de haver uma falha de comunicação na rede, serem armazenadas num *cartão de memória* as

informações relativas aos depósitos efetuados, salvaguardando-se assim a informação.

- **Placa de Rede Wireless** – Estas *placas de rede wireless* têm um alcance que pode atingir um quilometro e meio, tornando assim possível a comunicação com outros *iEcoPontos*, permitindo estabelecer redes mesh de forma a fazer chegar a informação ao *Servidor de Receção de Dados*.
- **Bateria e Painel Solar** – O painel solar torna possível carregar a bateria usando a luz solar. A bateria alimenta o sistema, tornando o mesmo energeticamente independente.

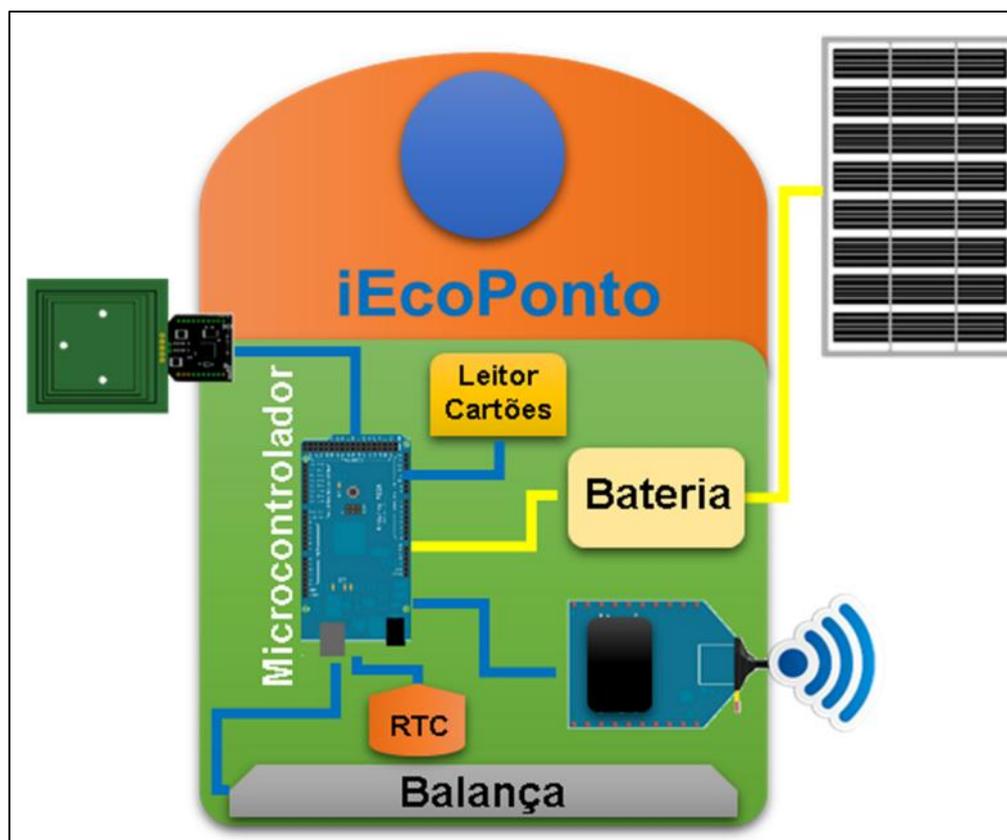


Figura 16 -O iEcoPonto.

### 5.3 Rede Mesh

A *Rede Mesh* (Figura17) é estabelecida com a implementação dos *iEcoPontos* que estando a distâncias que permitam estabelecer comunicação, o que pode ir até 1,5 Km. Esta rede pode ser usada para comunicações com outros fins, úteis em meio urbanos, estabelecendo assim aquilo que é denominado de uma rede de sensores usadas nas *Smart Cities*. Esta rede permite que a informação chegue ao *servidor de receção de dados* mesmo que o *iEcoPonto* esteja a uma distância superior a 1,5km do mesmo.

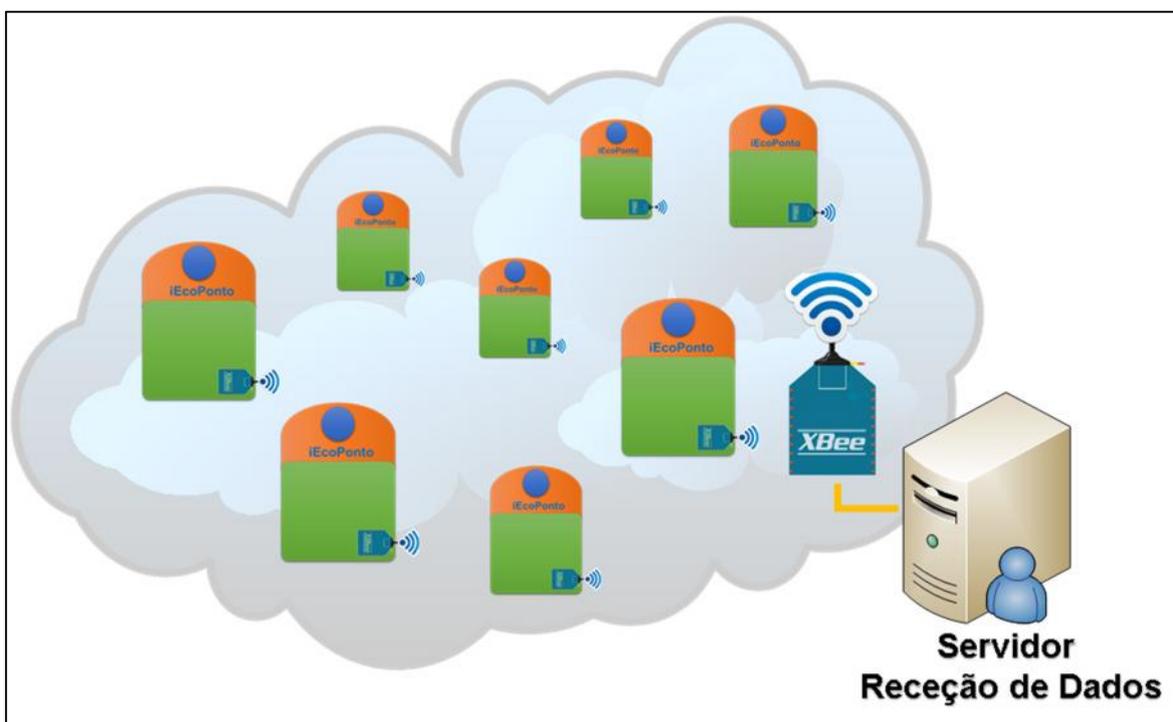


Figura 17 - Rede Mesh.

### 5.4 Servidor de Recepção de Dados

O *Servidor de Recepção de Dados* tem como finalidade receber os dados relativos ao *Depósito* (Figura 18) que um determinado *cidadão* efetua, enviando os mesmos para uma *Base de Dados*. O *Servidor de Recepção de Dados* é constituído por um computador que corre uma aplicação desenvolvida para o efeito e que recebe e processa os dados a partir da *placa de rede wireless* igual à existente nos *iEcoPonto* enviando-os para o *Servidor de Base de Dados*, que pode estar no próprio computador ou em outro computador.



Figura 18 - Servidor de Receção de Dados.

## 5.5 Servidor de Base de Dados

O Servidor de *Base de Dados* (Figura 19) é constituído por um SGBD – Sistema de Gestão de Base de Dados, no qual foi implementada uma Base de Dados de acordo com as necessidades do *Sistema iEcoSys*.

A *Base de Dados* recebe os dados vindos do *Servidor de Receção de Dados* e faz o cruzamento dos mesmos de acordo com a estrutura implementada, tornando assim possível disponibilizar os dados que são solicitados.

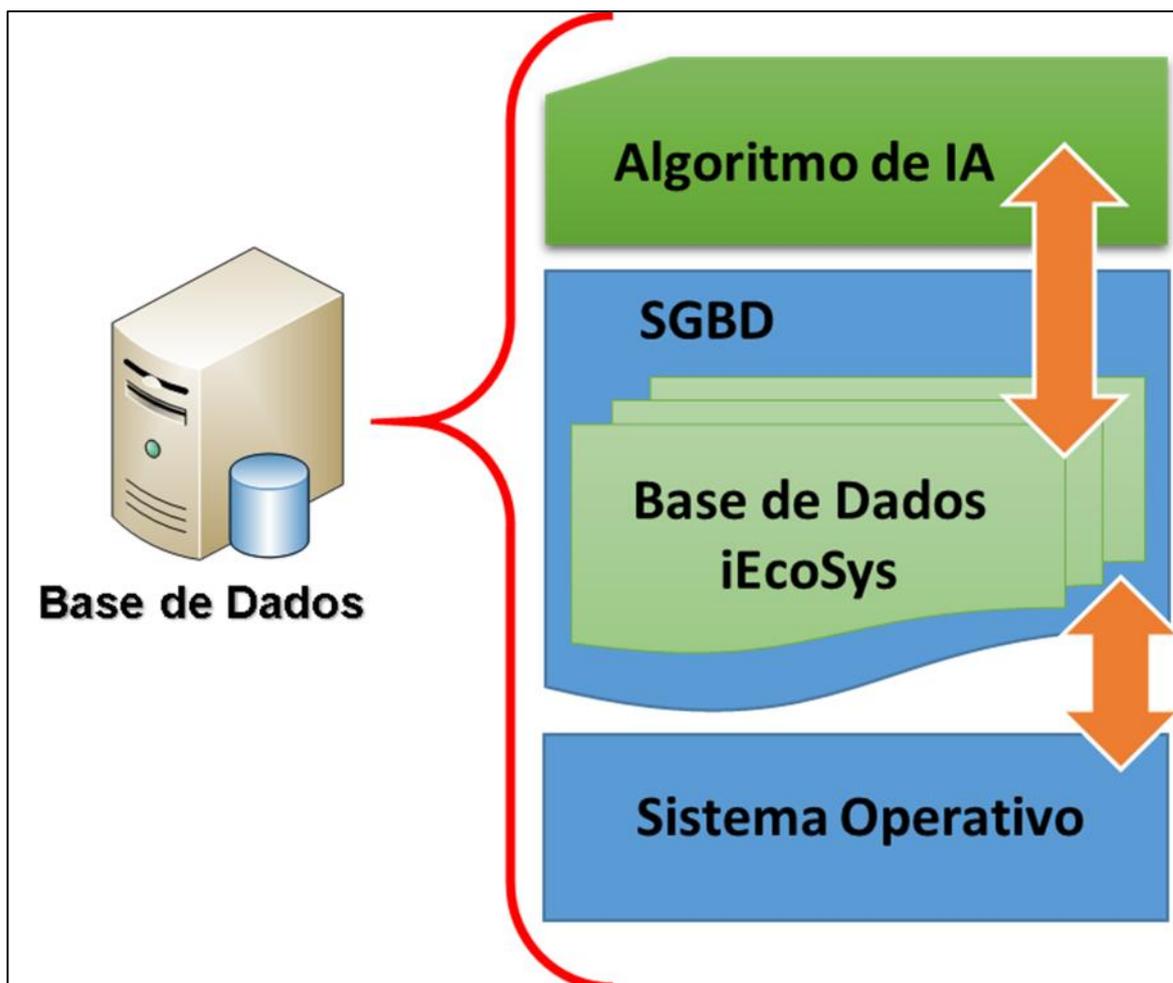


Figura 19 – Servidor de Base de Dados.

## 5.6 Definição de Rotas de Recolha

Com base na informação recolhida e inserida na Base de Dados é possível definir *rotas de recolha* dos resíduos de acordo com a quantidade de resíduos que existem num determinado *iEcoPonto* cujo registo se encontra na base de dados. Para a definição das Rotas pretende-se recorrer a Algoritmos de Inteligência Artificial que, de acordo com os dados existentes na base de dados, preveem a possibilidade de um *iEcoPonto* estar ou não cheio até a próxima recolha, definindo assim a rota do Veículo tendo em conta a necessidade ou não da recolha do *iEcoPonto em causa* num determinado momento, levando a consequentes poupanças energéticas, desgaste dos veículos e rentabilização dos recursos humanos.

## 5.7 Benefício do cidadão

O Sistema iEcoSys torna possível definir uma retribuição ao *cidadão* em função da *quantidade* e do *tipo de resíduos* depositados.

Desta forma o *cidadão* passa a ter um incentivo, os resíduos produzidos são valorizados e é criada uma cadeia de valor.

Um dos exemplos de incentivo seria o desconto na fatura da água, onde atualmente são faturados ao *cidadão* os resíduos não reciclados dos contentores indiferenciados.

Outro exemplo de retribuição ao *cidadão* poderia passar pelo envolvimento das grandes cadeias de supermercados que disponibilizariam ao *cidadão* iBags aquando da venda dos seus produtos. O *cidadão*, após o depósito dos iBags nos iEcoPontos, teria descontos à *posteriori* na respetiva grande superfície comercial como forma de incentivo.

Desta forma, poderia mudar-se o paradigma de **PAYT – Pay As You Throw** para o paradigma **GAYT – Get As You Throw** (Figura 20) [23].

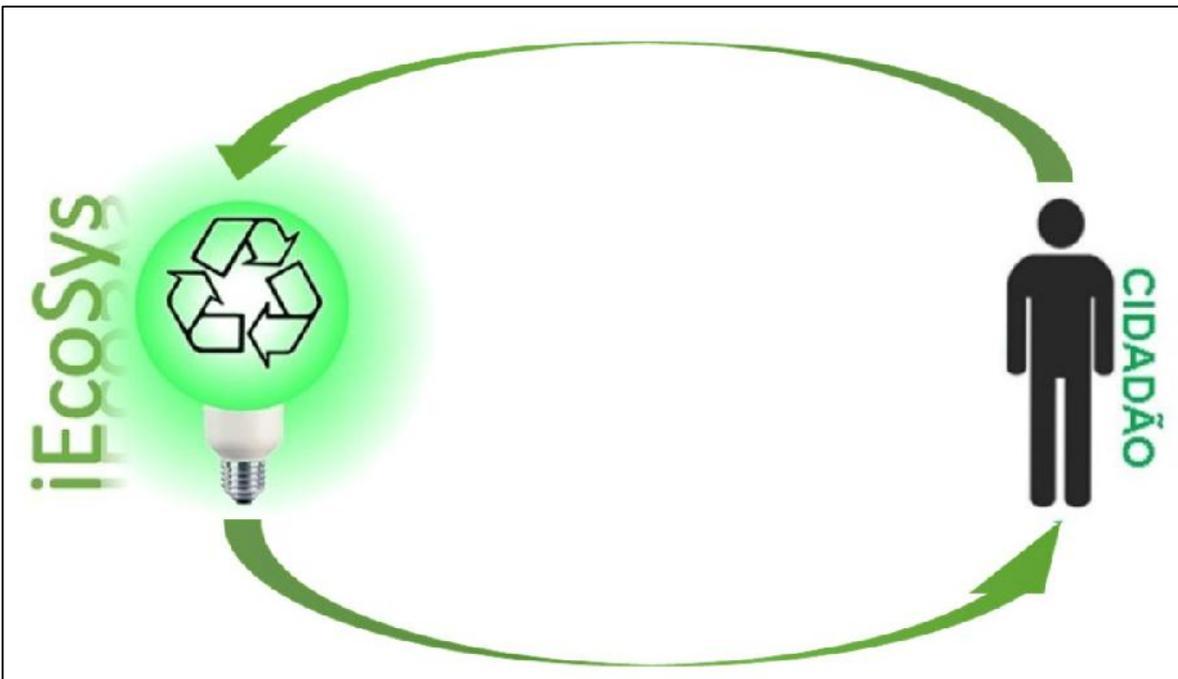


Figura 20 - Benefício do Sistema iEcoSys / cidadão.

## Capítulo 6 - Implementação do iEcoPonto

### 6.1 Introdução

Definidas as características de funcionamento e dos processos que compõem o *Sistema iEcoSys*, foram produzidos os requisitos funcionais e não funcionais do sistema por forma a obter o fluxo de informação pretendido.

Abaixo, a Figura 21, reproduz a primeira arquitetura desenhada e implementada para o sistema iEcoSys de forma simplificada, em laboratório que funcionava com as limitações que lhe eram inerentes. O sensor de pressão simulava uma balança e juntamente com o leitor RFID e o XBee simulavam o que seria o princípio do *iEcoponto*.

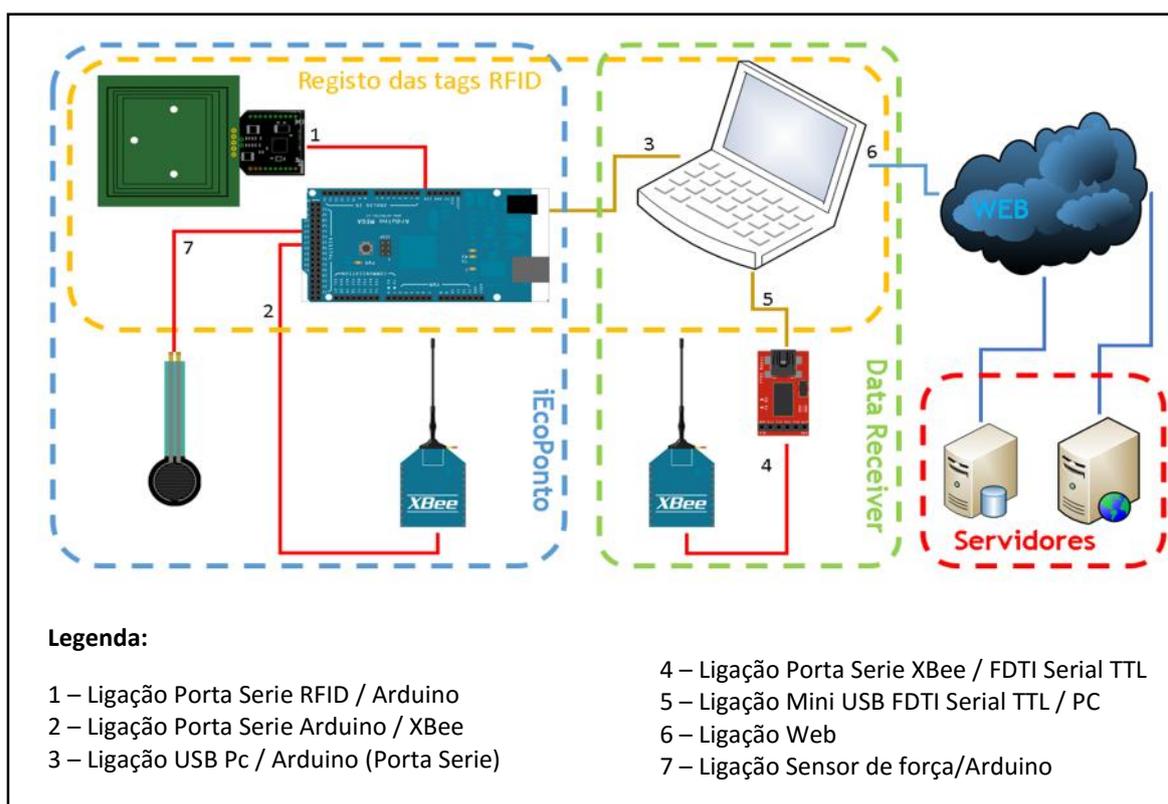


Figura 21 - Arquitetura do iEcoSys implementada inicialmente.

O *Data Receiver* não teve qualquer evolução em termos técnicos visto que a sua conceção original não poderia ser de outra forma para se obter o pretendido [23].

## 6.2 Estrutura do iEcoPonto

O *iEcoPonto* foi construído com recurso ao aproveitamento de materiais, diminuindo ao máximo aquisição de equipamentos, recorrendo sempre que possível à reutilização de elementos quer a nível de materiais, de equipamentos, quer a nível de componentes que pudessem ser reciclados.



Figura 22 - *iEcoPonto* em fase de construção, vista frontal e posterior.

A reciclagem dos produtos disponíveis para reutilizar no protótipo foi de alguma forma com o intuito de contribuir para a sustentabilidade ambiental, visto que se trata de um projeto *verde*, fazendo todo o sentido a reutilização e reciclagem de grande parte dos elementos do protótipo.

A estrutura do *iEcoPonto* é essencialmente de cartão aproveitado de caixas de equipamentos, tendo sido alteradas para o efeito, como se pode observar na Figura 22, onde se ilustram as vistas frontal e posterior da estrutura em cartão, com a adaptação de portas frontais e da cablagem. Na construção e montagem do *iEcoPonto* foi necessário embutir um conjunto de cabos que permitissem controlar os sensores e atuadores do ecoponto a partir do microcontrolador, usando essencialmente cabo de rede reaproveitado.

## 6.2.1 Sinalizadores Luminosos

Os *Leds* no *iEcoPonto* têm a função de indicador luminoso. Na parte superior de cada um dos pontos de depósito existem dois *Leds*, verde e vermelho, respetivamente.

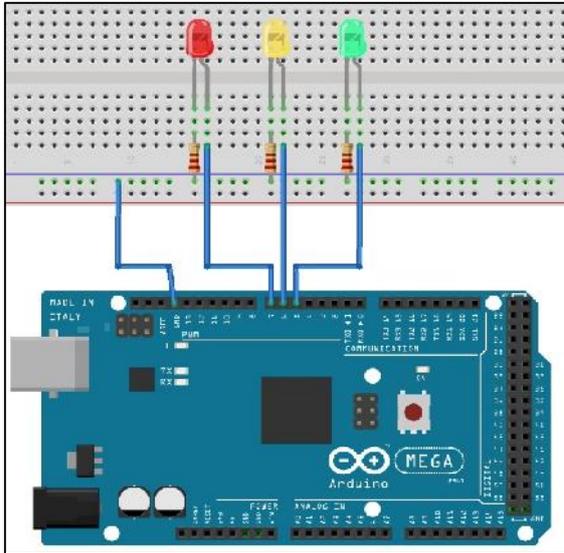


Figura 23 - Exemplo de ligação de *Leds* ao *Arduino Mega*.

Os *Leds* vermelhos por defeito estão ligados e assim que é efetuado um depósito de um determinado tipo de lixo, o *Led* verde existente no ponto de depósito correspondente ao tipo de lixo que está a ser depositado é ligado e o *Led* vermelho apaga-se, indicando assim visualmente onde se efetua o depósito.

As ligações ao microcontrolador são feitas diretamente às portas digitais Figura 23, usando apenas, adicionalmente, resistências de 1 K

para proteção dos mesmos. O controlo, ligar ou desligar os *Leds*, é feito aquando da programação do microcontrolador. Na programação das portas digitais do microcontrolador é necessário definir a porta como de *input* ou *output*, recorrendo ao comando *pinMode*. As portas são ativadas ou desativadas através do comando *digitalWrite*. Quando a variável está a *HIGH* a porta é ativada e os *Leds* são acesos,

```
void setup() {
  // inicializa a porta 4 como porta de saída / escrita.
  pinMode(4, OUTPUT);
  // escreve o estado da porta 4 como LOW / desligado
  digitalWrite(4, LOW);
}

void loop() {
  // liga o LED mudando o estado da porta 4 para HIGH (5volts)
  digitalWrite(4, HIGH);
  // espera por 1 segundo (ou 1000 milésimos de segundo)
  delay(1000);
  // desliga o LED mudando o estado da porta 4 para LOW (0volts)
  digitalWrite(4, LOW);
  // espera por 1 segundo (ou 1000 milésimos de segundo)
  delay(1000);
}
```

Figura 24 - Exemplo de programação das portas digitais

e desligados quando a variável está a *LOW*. Na Figura 24 está um exemplo completo de controlo de portas digitais do microcontrolador [15].

Em anexo é apresentado o código implementado para controlar os *leds*. (Anexo I)

## 6.2.2 iSacos e as Tags RFID

A tecnologia *RFID* ou *radio frequency identification*, teve a sua origem no período da segunda guerra mundial, em Inglaterra, na aplicação à aviação militar, com o passar das décadas esta tecnologia evoluiu tanto ao nível do seu formato físico, quer ao nível da sua aplicação. Atualmente existe um sem número de aplicações que vão desde chaves eletrónicas, até aos pagamentos multibanco, passando pela identificação de produtos.

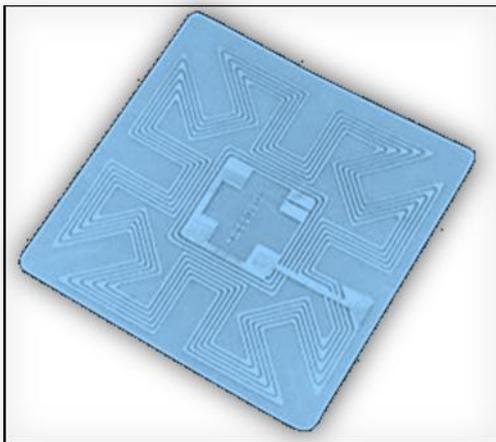


Figura 25 - Tag RFID.

As *tags RFID*, Figura 25, têm um papel preponderante no *Sistema iEcoSys* pois são incorporadas em sacos de lixo comum. Ao conjunto saco do lixo mais *tag* foi atribuído o nome de *iSaco*. As *tags*, para além do *ID único* que têm, já anteriormente associado a um cidadão, contêm na sua memória de 8 *Kbit*, o código que corresponde ao tipo de lixo. Desta forma, quando é efetuado o depósito, o

microcontrolador sabe qual a porta do ecoponto que deve abrir e que corresponde ao tipo de lixo a depositar [13].

As *tags RFID* assumem essencialmente duas tipologias:

- *Ativas*: as *tags* ativas caracterizam-se por terem uma fonte de energia,
- *Passivas*: as *tags* passivas caracterizam-se por não terem uma fonte de energia, estas apenas são ativas sempre que é fornecida energia quando se encontram junto do leitor *RFID*.

Outra grande particularidade que caracteriza as *tags RFID* tem a ver com a frequência de comunicação. As *tags* existentes no mercado possibilitam a comunicação em diferentes frequências de acordo com as respetivas normas ISO. As *tags* mais usadas são as passivas. As *tags* são essencialmente constituídas por um filamento de cobre que funciona como uma antena para emitir e receber o sinal, por um circuito integrado para guardar e processar informação e por um modelador e demodulador de sinal de radiofrequência que recebe energia a partir do leitor *RFID*.

As tags usadas no projeto pertencem a uma gama que comunica a uma frequência de 13,56 Mhz, com o protocolo ISO14443A, dispendo de uma EEPROM de 8 Kbit [13].

### 6.2.3 Identificação dos iSacos

O leitor RFID, Figura 26, permite fazer a leitura das tags RFID dos sacos do lixo iBags e tem como características importantes um chip PN532 da NXP, com

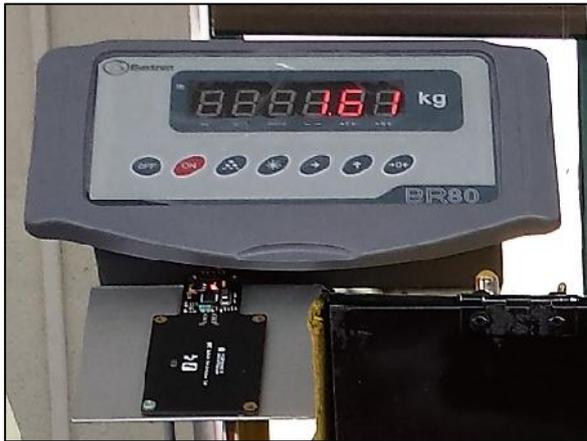


Figura 26 - Leitor RFID adaptado à estrutura da Balança.

capacidade de leitura e escrita de dados, suportando os modelos ISO 14443, MIFARE, FeliCa™ e NFCIP, e fazendo leituras a uma distância de 5cm.

Quando se aproxima a tag do leitor RFID, este lê o ID único existente na Tag e os dados que indicam o tipo de resíduos [5].

Os dados são enviados para o microcontrolador via porta série, como apresentado na Figura 27. Embora a comunicação entre os dois componentes seja feita de forma standard, via porta série, TX – RX fio azul, RX – TX fio verde, alimentação de 5V cabo vermelho e terra cabo preto, como apresentado na imagem, o controlo do leitor RFID é feito pelo microcontrolador através do envio de

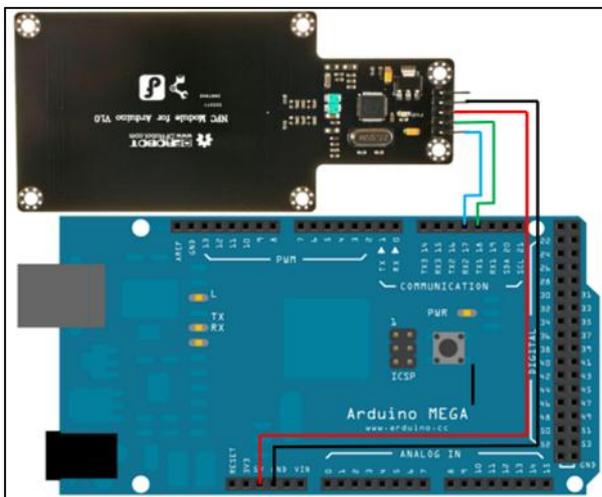


Figura 27 - Ligação do leitor RFID ao Arduino.

comandos de acordo com um determinado protocolo, estabelecendo a sincronização entre os dois componentes para que os dados cheguem corretamente à porta série do microcontrolador, garantido assim a sua integridade e sem perdas [15].

O código implementado encontra-se em anexo. (Anexo II)

## 6.2.4 Pesagem dos iSacos

Para se contabilizar o lixo relativo a um determinado cidadão é necessária a utilização de um instrumento de pesagem, ou seja, uma balança.



Figura 28 - Balança.

Face ao exposto, é necessário uma balança com características que tornem possível a contabilização dos resíduos depositados, ou seja, que permita uma gama de pesagens alargada e com alguma precisão. A balança utilizada é uma *Baxtran*, com báscula *monocélula TMM-150P*, com visor BR80 com proteção IP54, permite pesagens até 150 quilos, com uma precisão de 10 gramas. Outra questão prende-se com a necessidade de fazer chegar informação ao microcontrolador, o que levou à optar-se por uma balança com uma saída de dados RS 232 (Figura 28).

Com uma margem de pesagem suficientemente alargada, uma precisão suficientemente boa e uma porta de saída de dados, era necessário fazer chegar essa informação ao microcontrolador corretamente, sendo necessário adquirir um



Figura 29 - Conversor RS232 - TTL de cor verde, soldado a uma placa de circuito impresso.

componente extra que convertesse sinais do padrão RS232 da balança para que os convencionais níveis de tensões TTL do microcontrolador. Os conversores RS232 para TTL,

transformam tensões de comunicação que variam entre os 12V e os -12V nos

valores *TTL* de 0V e 5V do microcontrolador *Arduino*, que desta maneira recebe os valores do peso referentes aos resíduos depositados (Figura 29).

O código implementado para fazer a leitura dos valores recebidos da balança, encontra-se em anexo. (Anexo III)

### 6.2.5 Abertura das Portas Frontais

O Servomotor, Figura30, é uma máquina, eletromecânica, que apresenta movimento proporcional a um comando. Os servomotores recebem um sinal de



Figura 30 - Exemplos de servomotores.

controlo que verifica a posição atual para controlar sua movimentação, indo para a posição desejada externamente sob feedback de um dispositivo denominado de *encoder*.

Em contraste com os motores contínuos que giram indefinidamente, o eixo dos servomotores possui a liberdade de apenas cerca de 180º graus.

O servomotor é constituído essencialmente por três partes principais:

- O sistema atuador, constituído por um motor elétrico de corrente contínua, estando também presente um conjunto de engrenagens que formam uma caixa de redução, ajudando a amplificar o torque,
- O sensor normalmente um potenciômetro solidário ao eixo do servomotor. O valor da resistência elétrica indica a posição angular em que se encontra o eixo,
- O circuito de controlo, formado por componentes eletrónicos discretos ou circuitos integrados que recebem um sinal do sensor e um sinal do controlo, acionando o motor no sentido necessário para posicionar o eixo na posição desejada.

Os servomotores são um dos componentes do iEcoPonto devido à força que lhes é característica e à possibilidade de controlo do ângulo de rotação, o que facilita o controlo de abertura dos iEcoPontos, sendo assim utilizados para abertura e fecho nas portas frontais metálicas.

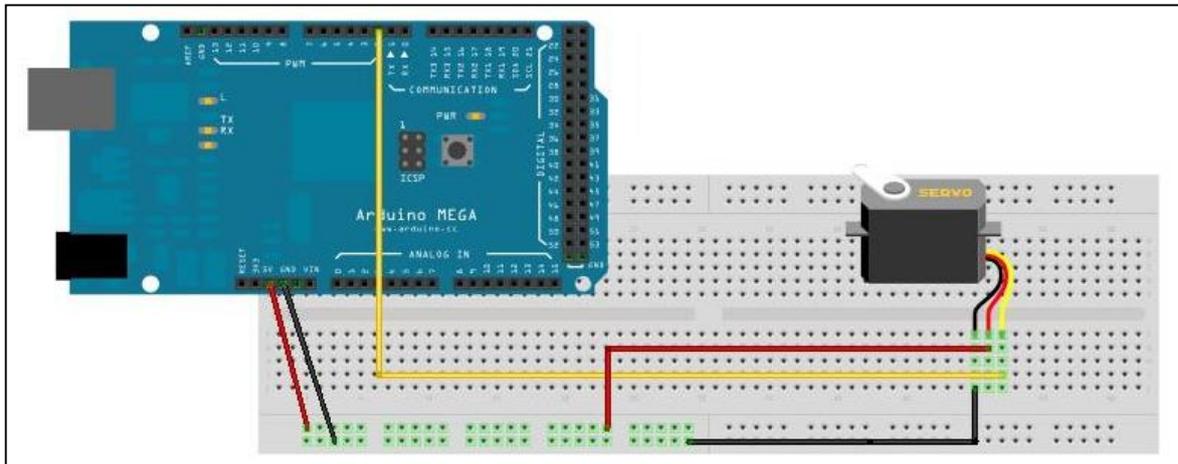


Figura 31 - Exemplo de uma Ligação de um Servo Motor ao Arduino.

Os servomotores estão fixados no interior de cada um dos compartimentos do iEcoPonto para ser feita a abertura e fecho das portas os serviços estão ligados diretamente ao microcontrolador o controlo das ações é feito aquando da programação do microcontrolador. As ligações são feitas por 3 cabos responsáveis pela alimentação, *Vcc* e *Gnd* e um terceiro que é responsável pelo controlo do servo (Figura 31) [15].

Para efeitos de controlo dos servomotores foi implementado o código disponibilizado em anexo (Anexo IV).

### 6.2.6 Abertura das Portas Posteriores

Os motores de corrente contínua são outros dos componentes utilizados *iEcoPonto*. Estes são utilizados para abertura e fecho das portas situadas na parte posterior do *iEcoPonto*. Estas portas são abertas após a pesagem dos *iSacos* dos resíduos ter sido efetuada e fechadas imediatamente. Os motores e a estrutura mecânica de abertura e fecho foram reutilizados de leitores de cd e dvd antigos e estão ligados a um circuito integrado denominado *L293D*, Figura32, que permite controlar o sentido de rotação de motores ao receber um sinal do microcontrolador para acionar a direção ou sentido de rotação do motor que se pretende.

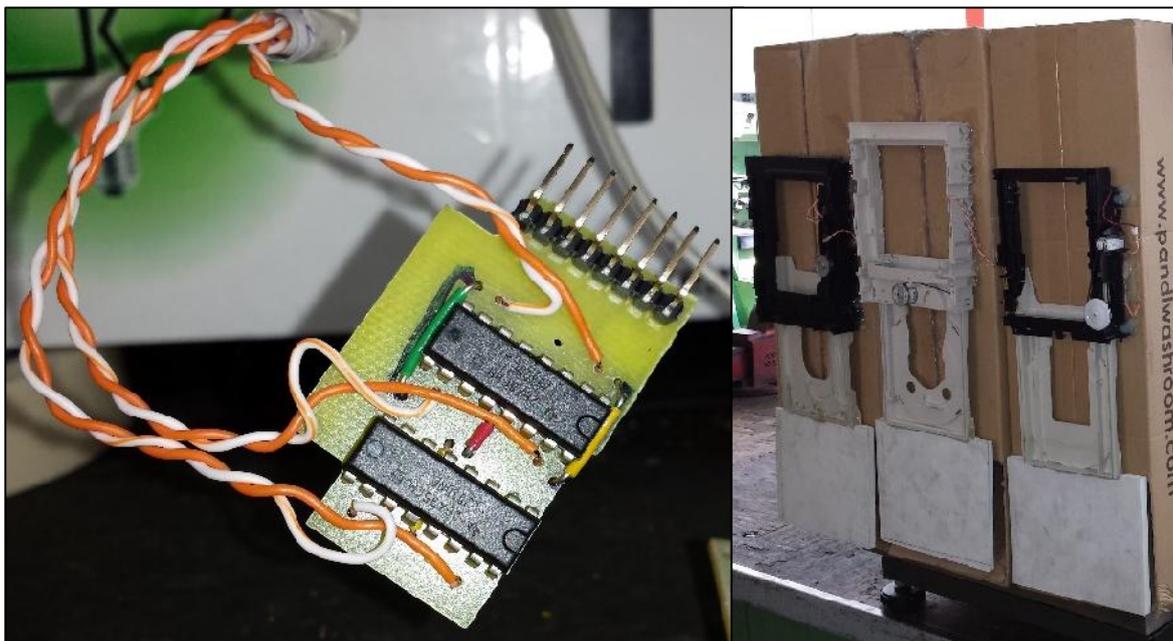


Figura 32 - Circuitos Integrados L293D (Ponte H) utilizados no iEcoPonto.

O L293D é um circuito integrado usado para controlar motores de corrente contínua em ambos os sentidos, contendo duas Pontes H. As Pontes H são circuitos compostos por transístores, díodos e algumas resistências para proteger os transístores. Este CI, Figura 33, de 16 pinos pode controlar um par de motores em simultâneo, independentemente do sentido de rotação.

O Controlo do motor pelo microcontrolador é feito pelas portas digitais. Os fios usados, azul claro e azul-escuro, Figura 34, são ligados das portas digitais do

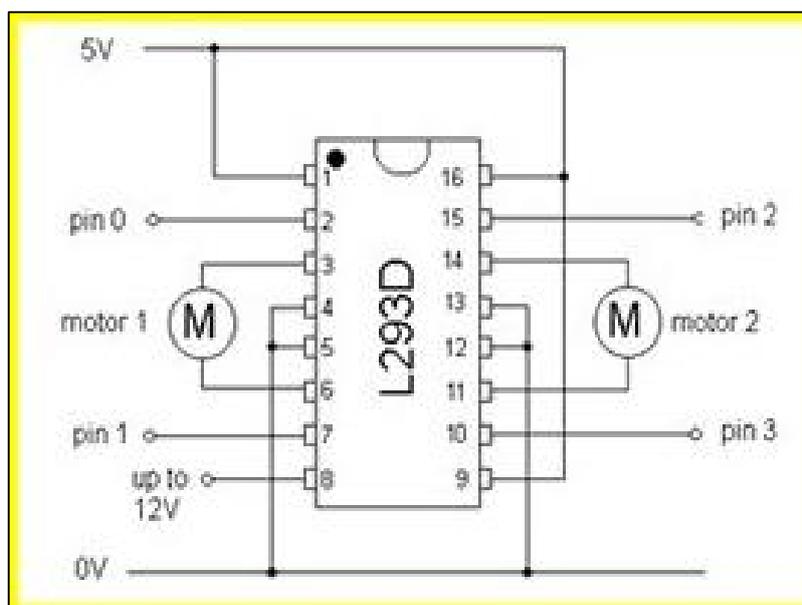


Figura 33 - Esquema elétrico de ligação do CI L293D e os motores.

microcontrolador aos pinos 2 e 7 do *L293D*, ativando um pino a *HIGH* e outro a *Low* e vice-versa, alternando assim o sentido de rotação do motor. Também é possível alimentar o motor usando uma fonte de alimentação externa, recorrendo ao uso do pino 9 do *CI L293D*. Existe um terceiro fio, cinzento, que permite controlar a velocidade do motor. O fio cinzento na Figura 34, está ligado a um pino digital com uma característica particular. Esse pino digital permite um sinal digital especial, o *PWM – Pulse with Modulation*.

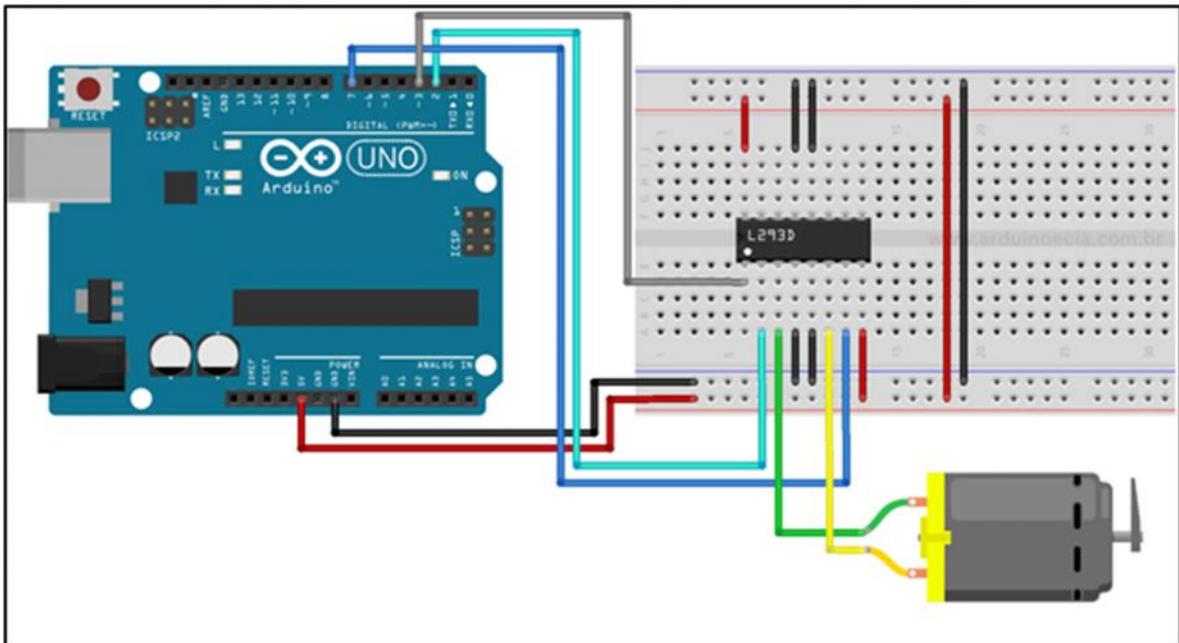


Figura 34 - Exemplo de ligação de um motor CC a um CI L293D controlado por um microcontrolador Arduino.

O sinal PWM basicamente é o tempo que o sinal digital que está a *Height* ou a *Low*, durante um determinado período de tempo. Desta forma, através da largura do pulso de uma onda quadrada, é possível o controlo da potência ou velocidade de um motor [15].

A programação para controlo do sentido de rotação do motor é similar à dos Leds, visto que apenas é necessário controlar as portas digitais.

### 6.2.7 Módulo de Comunicação XBee

O XBee é um módulo de comunicação de redes sem fios que usa o padrão de comunicação da *ZigBee Alliance*, concebido em 1998 (Figura 35).

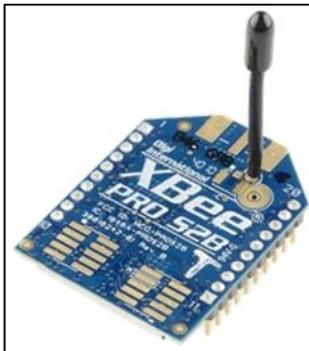


Figura 35 - Módulo XBee.

*ZigBee Alliance* é uma associação constituída por grandes empresas da área das tecnologias. As empresas que fazem parte deste consórcio são a Philips, a SAMSUNG, a Motorola, a Atmel, entre outras, que trabalham em conjunto com o objetivo de criarem uma alternativa às atuais comunicações sem fios.

A necessidade por parte das empresas de um padrão de comunicação simples, robusto, de baixo consumo energético, baixas taxas de transferência e baixo custo levou à formação deste padrão de comunicação.

A *ZigBee* usa a norma IEEE 802.15.4 desde maio de 2003, data em que foi homologada, tendo sido revisto em 2006. Estes dispositivos dispõem de comunicações sem fios com baixas taxas de transferência, importante para aplicações em redes WPAN - *Wireless Personal Area Network*, que requerem baterias com durabilidade e que confirmam uma autonomia alargada.

Os elementos XBee que implementam o padrão *ZigBee* são fabricados pela *DIGI Internacional*, uma empresa especializada em tecnologias *wireless*, também ela integrando a *ZigBee Alliance*. Este módulo XBee tem um consumo de 3,3V a 295mA e opera nos 868 MHz na Europa, com uma taxa de transferência que pode ir até aos 250 kbits, e um alcance que pode atingir 1.600 metros e uma antena de +17 dBm, ideal para as comunicações intermitentes de um sensor. As redes ZigBee também permitem a encriptação dos dados com *128 bit symmetric encryption keys* [14].

## Topologias ZigBee

As Redes ZigBee podem ter dois tipos de dispositivos, os dispositivos *FFD* e os dispositivos *RFD*, existindo três classes de dispositivos lógicos que definem a rede, *Coordenador*, *Router* e *Dispositivo final*. Os elementos *FFD*, *Full Function Device*, dispositivos de funções completas, são dispositivos complexos e numa topologia de Rede ZigBee podem ter o papel de *Coordenador*, *Router* ou *dispositivo final*. Os elementos *RFD*, *Reduced Function Device*, *Dispositivos de Funções Reduzidas*, são dispositivos mais simples e numa topologia de rede ZigBee podem assumir o papel de *Dispositivo final*. Na prática, podem ser simples interruptores ou sensores.

O *ZC*, *ZigBee Coordinator*, só pode ser implementado através de um dispositivo *FFD*. O coordenador é responsável pela iniciação, distribuição de endereços e manutenção da rede. Faz o reconhecimento de todos os Nós e pode servir como ponte entre várias redes ZigBee. Por outro lado, o *ZR*, *ZigBee Router*, apenas pode ser implementado a partir de um dispositivo *FFD*, tendo as funções de nó normal na rede. Contudo, tem características adicionais que lhe permitem funcionar como um *router* mediador entre nós. Finalmente, o *ZED*, *ZigBee End*

*Device*, é onde os atuadores ou sensores são ligados, e pode-se usar dispositivos *FFD* ou *RFD*.

A partir dos componentes anteriormente descritos e com as características que lhes são próprias, resultam redes cujas topologias podem assumir as seguintes configurações (Figura 36):

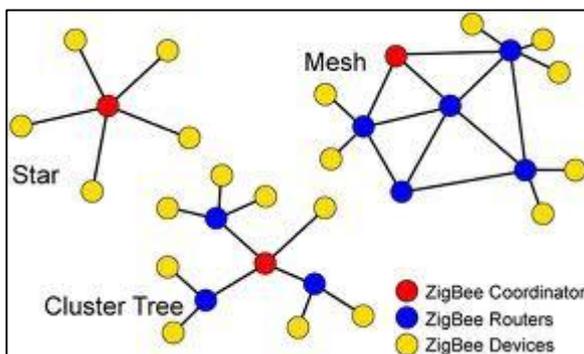


Figura 36 - Topologias de redes ZigBee.

- Mesh: nesta topologia a rede ajusta-se automaticamente, ou seja, a rede organiza-se automaticamente para otimizar o fluxo de dados. Os múltiplos caminhos exequíveis para a comunicação entre os nós, torna possível abarcar uma longa área geográfica,

- Cluster Tree: idêntica à topologia Mesh, mas uma rede em árvore o *coordenador* tem o papel de um nó na troca de informação entre os nós *Router* e *End Device*.
- Star: é a topologia de rede *ZigBee* das mais simples de serem implementadas, é sendo formada por um nó *Coordenador* e por vários nós *End Device* [14].

## Programação X-CTU

Os módulos *XBee* produzidos pela *DIGI* permitem que o seu *firmware* seja alterado de acordo com as necessidades das aplicações a implementar.

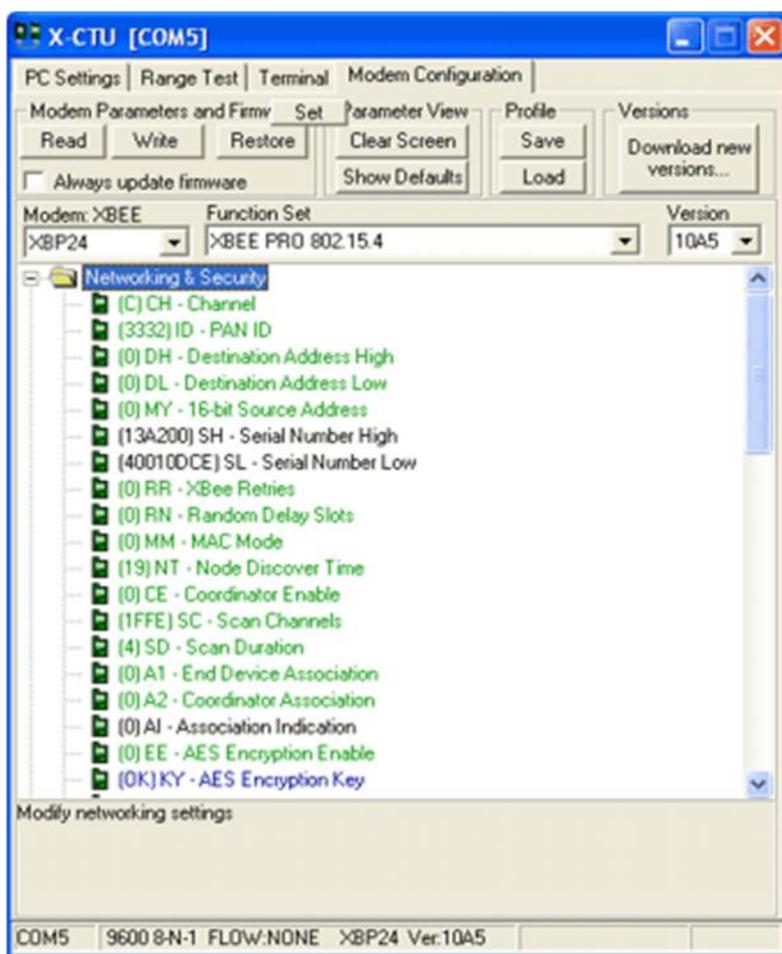


Figura 37 - Uma janela do software do X-CTU.

Assim, a *DIGI* disponibilizou um *software* o *X-CTU*, que possibilita a alteração do *firmware* dos módulos. Os módulos podem ser ligados a um dispositivo que permita comunicar via porta série com o computador, como por exemplo um *FTDI* que converte os sinais da porta série em formato *TTL* para sinais *USB*.

Uma vez estabelecida a comunicação entre o *X-CTU* e o módulo *XBee* é possível ler os valores das definições do mesmo

e fazer alterações nos respetivos campos (Figura 37).

O *X-CTU* disponibiliza, de acordo com as características físicas do módulo o *firmware* disponível e compatível com o mesmo, sendo executado o *update*. Uma vez feito o *update*, o módulo fica pronto a usar [22].

## Implementação

O XBee é o componente que permite fazer a comunicação via *wireless* entre os *iEcoPontos*, estando ligado diretamente ao microcontrolador. A comunicação é feita diretamente pela porta série (Figura38). A rede *mesh* constituída pelos *iEcoPontos* que comunicam entre si, cuja distância pode atingir um raio de um quilômetro em condições ideais, permite fazer chegar a informação relativa a um dado depósito num determinado momento à *Base Station* (Figura 41).

Os módulos usados na implementação são os *XBee Pro S2B*, configurados usando o *X-CTU*, com recurso ao firmware disponível, um como *End Device* e outro como *Coordenador*.

O módulo que fica no *iEcoPonto* fica configurado com o *firmware End Device* e o módulo que ficou na *Base Station* fica como *Coordenador*.

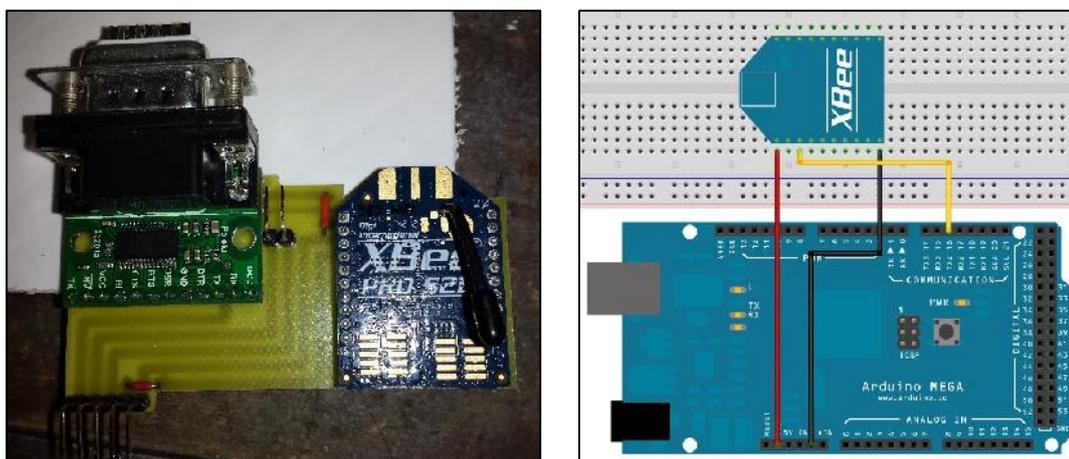


Figura 38 - Modulo XBee de cor Azul, montado no circuito impresso desenhado para o efeito.

## Regulador de Tensão

Os circuitos integrados reguladores de tensão têm como finalidade a conservação da tensão num circuito elétrico, ou seja, manter a tensão produzida por uma determinada fonte nos limites requeridos pelo componente elétrico que necessita de ser alimentado, sendo a tensão de entrada sempre superior à sua tensão de saída.

Na implementação do *iEcoPonto* houve a necessidade de alimentar um módulo XBee que permite uma alimentação nominal máxima de 3,3V. O regulador de tenção *CI LM1117* faz a conversão dos 6,4V em 3,3V necessários para alimentar o módulo referido (Figura 39).

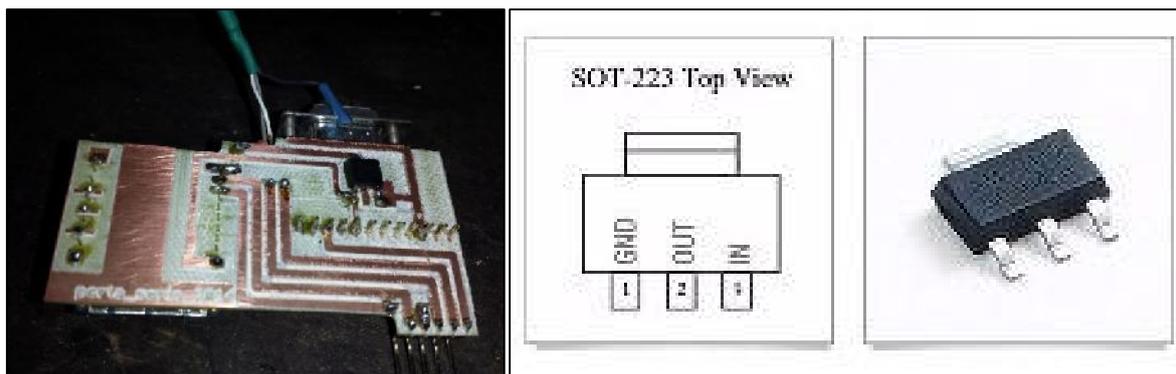


Figura 39 - Circuito impresso com o regulador de tensão LM 1117 para alimentar o módulo XBee à esquerda e pinout do regulador, à direita.

## 6.2.8 Painel Solar - Acumuladores

A existência de um painel solar e de acumuladores energéticos (baterias) tem como objetivo dar autonomia energética ao *iEcoPonto*.

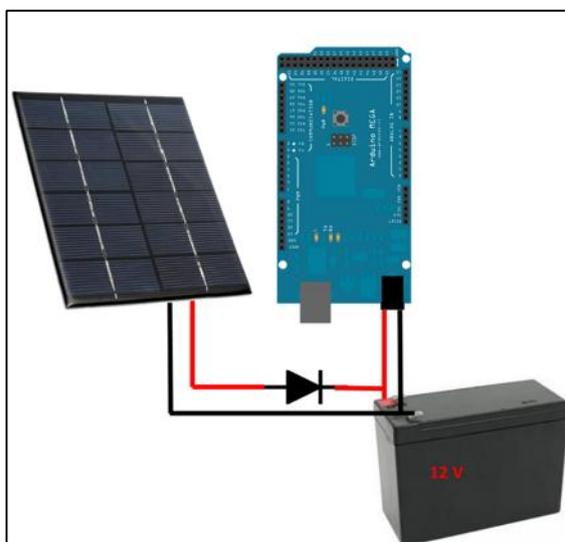


Figura 40 - Circuito de alimentação do microcontrolador.

A autonomia energética do *iEcoPonto* vai contribuir também para a sustentabilidade ambiental ao recorrer apenas excepcionalmente a outra fonte de energia, caso seja necessário, por exemplo a rede pública.

A opção tomada para conceber a autonomia do *iEcoPonto*, passou por usar um painel solar de 10W com um *output* de 12V e de duas baterias em paralelo que alimentam a plataforma (Figura 40). A alimentação do sistema é

feita diretamente através da plataforma *Arduino*. Poderia também ter sido feita uma divisão do circuito em dois, dividindo o mesmo em circuito de controlo e de potência, o que terá de ser feito aquando da implementação de um protótipo para ambiente real. Contudo, optou-se por esta configuração aquando da implementação inicial visto que os elementos constituintes do protótipo não requeriam um consumo superior ao que a plataforma podia fornecer. A alimentação da balança é feita de forma independente, visto que já dispunha de uma bateria de origem, pelo que houve a necessidade de ser feita a conexão da *terra* de ambos os circuitos por

causa das comunicações da *porta série* que tem como referência a *terra*, que necessita de ser comum em ambos os dispositivos de comunicação. Também a alimentação do *leitor RFID* é feita recorrendo à bateria da Balança por uma questão de logística, devido ao local físico onde se encontra, recorrendo-se à ajuda de um regulador de tensão 7805 que alimenta o *leitor* com 5V. Relativamente à autonomia da balança, não foi possível testar de forma exaustiva a independência energética, contudo os testes do uso regular foram bons.

### 6.3 Comunicação do iEcoPonto para a Base Station

#### XBee – FTDI – PC

A receção dos dados da rede *Mesh* implementada pelo *iEcoPonto* é feita por uma *Base Station*. Neste caso em particular, a *Base Station* é constituída por um

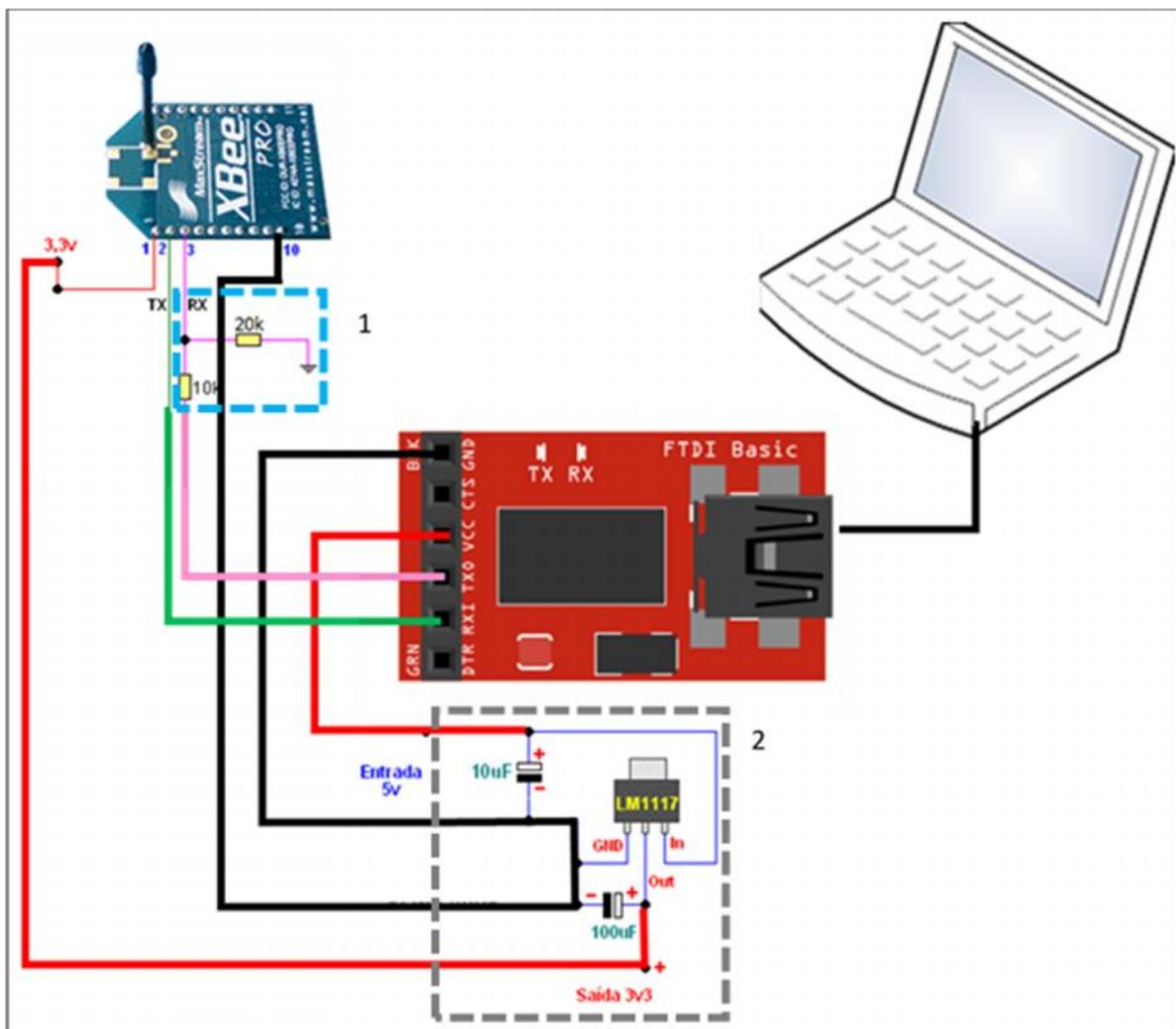


Figura 41 - Ligações da Base Station.

*XBee*, um *FTDI chip* e um computador. O *XBee* já referido anteriormente tem a particularidade de ser programado em modo *Coordenador* em relação ao anterior, ou seja, recebe os dados de todos os *iEcoPontos*.

O *FTDI Chip* é um componente que recebe os sinais de dados do *XBee* na porta série em formato *TTL* e os converte para sinais de dados *USB*. A ligação entre o computador e o *FTDI* é feita por um cabo *Mini-USB* do lado do *FTDI* e *USB* do lado do Computador (Figura 41).

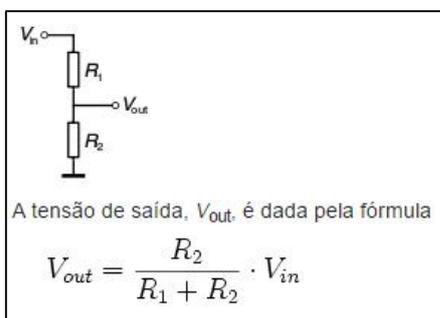


Figura 42 – Fórmula de cálculo do divisor de tensão.

A ligação entre o *XBee* e o *FTDI* é feita por um cabo concebido com alterações. As transformações feitas no cabo passaram por adicionar um regulador de tensão e um divisor de tensão. O regulador de tensão acrescentado, *CI LM1117*, tem a função de fornecer uma tensão de 3,3V ao *XBee*, visto que o *FTDI* disponibiliza apenas 5V.

O divisor de tensão foi construído com recurso a duas resistências cuja fórmula de cálculo se encontra na Figura 42. A necessidade do divisor de tensão prende-se com o facto dos níveis *TTL* dos sinais de dados para a comunicação terem de ser iguais, visto que a tensão de funcionamento do *XBee* e do *FTDI* são diferentes.

Uma vez configurado o *hardware*, é possível recorrendo a um programa que permita seleccionar a porta *COM* a que está associada a porta *USB* onde o *FTDI* está ligado, visualizar os dados vindos do *iEcoPonto*.

## 6.4 Fluxograma do Algoritmo do iEcoPonto

O funcionamento do *iEcoPonto* tem como ponto de partida a leitura de uma *tag RFID* embutida num saco do lixo, cujo conjunto é denominado de *iSaco* e que desencadeia um conjunto de passos representados no fluxograma da figura 43.

A leitura correta da *tag*, despoleta o armazenamento dos dados *ID da Tag* e *ID do tipo de Lixo*. Consequentemente, desencadeia um conjunto de ações nos atuadores. Os atuadores a ser acionados dependem do *ID do tipo de lixo*.

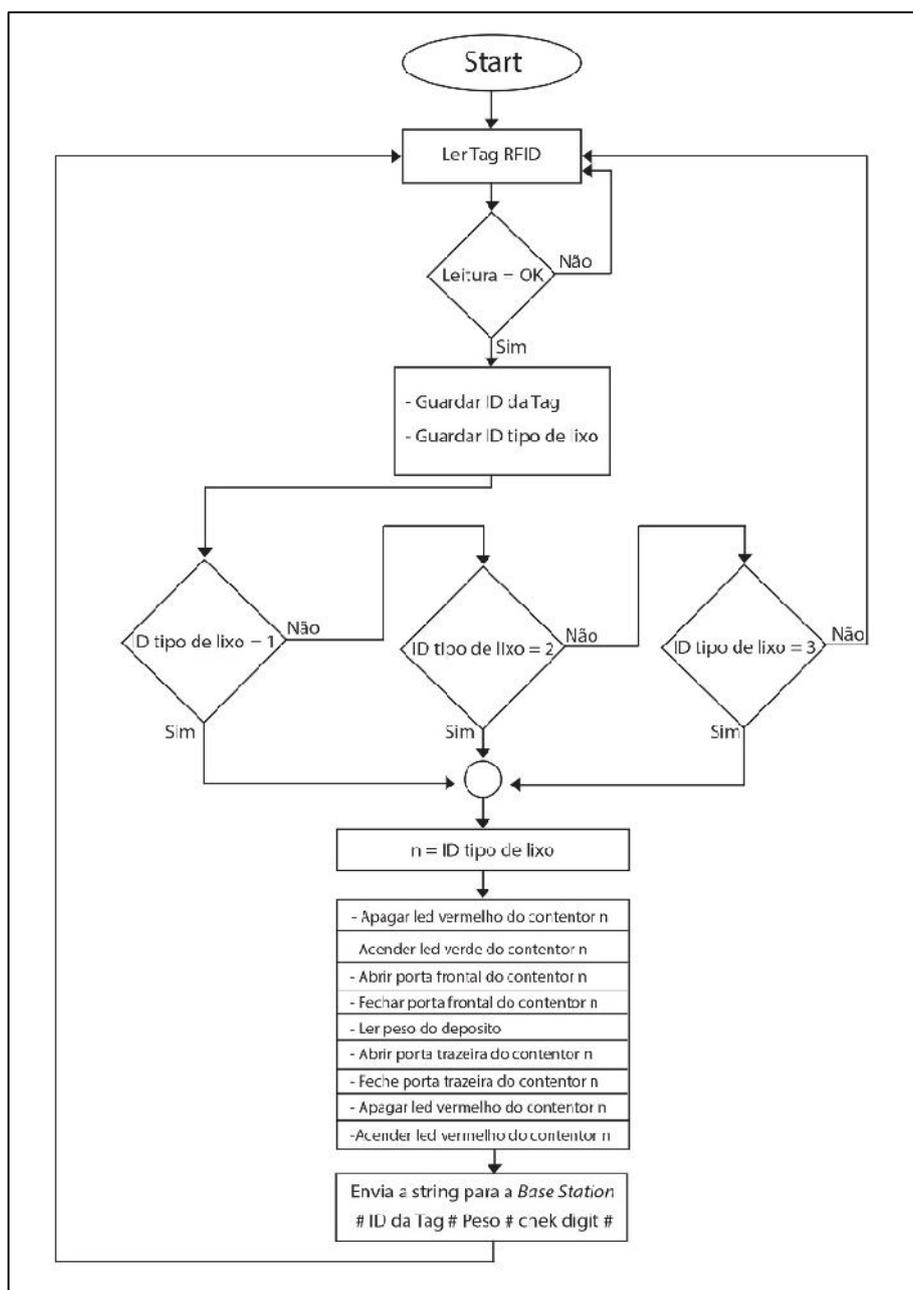


Figura 43 - Fluxograma do funcionamento do iEcoPonto.

De seguida, são executadas um conjunto de ações para que o processo do depósito seja concluído. Por último, o microcontrolador envia através do XBee os dados relativos ao depósito.



Figura 44 - Protótipo do iEcoPonto.

O envio e receção de dados está por vezes sujeito a interferências por vários fatores. Assim, de forma a garantir que a informação que chega ao destino é válida, foi definido um protocolo com os dados necessários e indispensáveis ao funcionamento e razão de ser do sistema: *ID da Tag, n.º Ecoponto, Peso e check Digit* que é o número de controlo.

Na Figura 44 é apresentada a fotografia do iEcoPonto concluído, onde se pode observar vários componentes tais como a balança, o leitor RFID, os leds e um iSaco.

# Capítulo 7 - Conclusões

## 7.1 Conclusão

Este trabalho visou apresentar um protótipo inteligente de *RSU* com vista à implementação de um sistema com o conceito *GAYT* (Get As You Throw), como forma de incentivo à reciclagem.

O sistema *iEcoSys* é composto por tecnologias e componentes que vão desde o *iSacos* que contêm tags *RFID*, ao *iEcoPonto*, composto por um leitor *RFID*, um *Arduino*, um *XBee*, entre outros componentes de eletrónica adicionais, passando por servidores de receção de dados, servidores de base de dados, indo até ao *frontend* disponibilizado ao cidadão.

A idealização, conceção, desenho da arquitetura e implementação do *iEcoSys* teve como suporte de apoio os conhecimentos adquiridos ao longo do curso de engenharia informática em áreas como programação, eletrónica, arquitetura de computadores, sistemas digitais, base de dados, consolidados no período passado no *MagicKey*.

O presente protótipo encontra-se ainda em fase preliminar de testes laboratoriais com vista à sua validação e otimização experimental, designadamente através da deteção de eventuais melhorias de funcionamento, nomeadamente, está a ser estudada/analizada a validação dos resíduos pós recolha.

Esta é a estrutura base do projeto, contudo tendo esta base como ponto de partida, é possível desenvolver um alargado conjunto de sistemas de gestão associados às atividades desenvolvidas pelas organizações, que interagem com o *iEcoSys*, permitindo desta forma rentabilizar ainda mais o sistema, com a diminuição dos custos operacionais.

Os resultados obtidos no funcionamento do sistema são muito promissores, representando um contributo significativo para a sustentabilidade ambiental.

O sistema *iEcoSys*, com os devidos aperfeiçoamentos e ajustamentos à realidade observada *in loco*, pode ser a resposta tecnológica/conceptual para revolucionar de uma forma substancial a forma de como se faz reciclagem, tornando-se um incentivo à sua prática pelo *cidadão*, levando assim à adesão em massa das populações.

## 7.2 Trabalho Futuro

O trabalho futuro passa por vários pontos a desenvolver no seguimento do meu percurso académico no Mestrado de Computação Móvel, destacando-se os seguintes itens:

- Desenvolvimento das aplicações de *backend* de suporte ao sistema iEcoSys,
- Desenvolvimento das aplicações de *frontend* de suporte ao sistema iEcoSys e para o cidadão,
- Aplicação móvel para anulação de depósitos incorretos e para o depósito de objetos especiais com grande volume que não se enquadrem na recolha seletiva ordinária,
- Uma aplicação de inteligência artificial que a partir dos dados existentes na base de dados recolhidos pela *Base Station*, defina as rotas mais curtas das viaturas de recolha de acordo com as necessidades dos *iEcoPontos* isto é se estão com um volume de enchimento acima do definido,
- Um jogo educativo e de divulgação do iEcoSys, contribuindo para a aprendizagem do processo de reciclagem e da importância do papel do cidadão no processo com vista à consciencialização e redução da sua pegada ecológica,
- Conceção de um protótipo em tamanho Real do *iEcoPonto* final,
- Por fim a realização do estudo de campo para prova de conceito do iEcoSys.

## Bibliografia

- [1] L. Dahlén, H. Åberg, A. Lagerkvist, P. Berg, “Inconsistent pathways of household waste”, *Waste Management* 29, pp.1798-1806, 2008
- [2] B. Bilitewski, “From traditional to modern fee systems”, *Waste Management* 28, pp. 2760–2766, 2008
- [3] J. Reichenbach, “Status and prospects of pay-as-you-throw in Europe – A review of pilot research and implementation studies”, *Waste Management*, pp. 2809–2814, 2008
- [4] I. Puig-Ventosa, “Charging systems and PAYT experiences for waste management in Spain”, *Waste Management* 28, pp.2767–2771, 2008
- [5] M. Hannan, M. Arebey, R. Begum, H. Basri, “Radio Frequency Identification (RFID) and communication technologies for solid waste bin and truck monitoring system”, *Waste Management* 31, pp. 2406–2413, 2011
- [6] L. Huida, F. Bingchen, Y. Liheng, L. Yanan, “RFID-Based Hazardous Waste Management Platform Establishment”, *Procedia Engineering*, n.º 29, pp.4–8, 2012
- [7] C. Gonçalves, “Aplicação do Sistema Pay-As-You-Throw no Município de Lagoa – Conceção e Implementação”, Universidade do Algarve, 2010
- [8] União Europeia, “Diretiva 2012/19/EU do Parlamento Europeu e do Conselho”, *Jornal Oficial da União Europeia*, pp. 38-71, Julho de 2012.
- [9] EGF – Empresa Geral do Fomento, Grupo Águas de Portugal, “Relatório e Contas 2012”, Março de 2013.
- [10] Eurostat, the statistical office of the European Union, Newsrelease, March 2013.
- [11] G. Hancke, B. Silva, G. Hancke Jr., “The Role of Advanced Sensing in Smart Cities.” *Sensors*, 13, pp. 393-425, 2013.
- [12] K. Finkenzeller, “RFID Handbook”, 2003
- [13] FM - Fudan Microelectronics, “FM11RF08 8Kbits Contactless Card IC”, Functional specification, Ver 2.1, May 2008.
- [14] F. Robert, “Building Wireless Sensor Networks”, O’ReiLLY, 2011.
- [15] Arduino website, “<http://www.arduino.cc/>”, ultimo acesso: 12/02/2014.
- [16] M. Michael, “Arduino Cookbook”, O ReiLLY, USA, 2012.

- [17] P. Ferrão, et al, “PNGR – Plano Nacional de Gestão de Resíduos 2011 – 2020”, Lisboa 2011
- [18] Política na Rede, “<http://www.politicanarede.com/2013/08/em-pequim-metro-aceita-garrafa-pet-como.html>”, ultimo acesso: 14/04/2015.
- [19] RVM Installation, “<http://www.incomrecycle.com/en/about/4.html>”, ultimo acesso: 19/06/2015.
- [20] PPLWARE, “<http://pplware.sapo.pt/gadgets/high-tech/e-se-a-lata-do-lixo-lhe-desse-sinal-wifi-para-navegar-na-internet/>”, ultimo acesso: 23/06/2015
- [21] J. Cebreiros; M. Pérez Gulín; “Guia Smart Cities “Cidades com Futuro””, Espanha 2014
- [22] DIGI, “<https://docs.digi.com/display/XCTU/Configure+your+modules>”, último acesso: 22/06/2015.
- [23] P. Reis et al, “Sistema Inteligente de Valorização de Lixo Urbano”, ICUBI 2013, International Conference on Engineering, Covilhã 2013
- [24] INPI – Instituto Nacional de Propriedade Industrial, “Boletim da Propriedade Industrial”, nº 2015/02/06/ (026/2015), 6 de fevereiro de 2015
- [25] P. Reis et al, “Intelligent System for Valorizing Solid Urban Waste”, Sistemas y Tecnologías de Información: Actas de la 9ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI'2014), Spain, vol. 1, pp. 88-91, AISTI - Asociación Ibérica de Sistemas y Tecnologías de Información, Junio 2014.
- [26] P. Reis et al, “iEcoSys – An Intelligent Waste Management System”, New Contributions in Information Systems and Technologies, volume 1 pp 843-853, Springer International Publishing, 2015
- [27] P. Reis et al, “Sistema Inteligente de Incentivo à Reciclagem”, Sistemas y Tecnologías de Información: Actas de la 10ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI'2015), Spain, AISTI - Asociación Ibérica de Sistemas y Tecnologías de Información, Junio 2015.

# ANEXOS

## Anexo I

### Código para controlo dos Leds

```
void setup() {  
  
    // Defenição das portas digitais como Output  
    pinMode(37, OUTPUT);  
    pinMode(39, OUTPUT);  
    pinMode(41, OUTPUT);  
    pinMode(43, OUTPUT);  
    pinMode(45, OUTPUT);  
    pinMode(47, OUTPUT);  
  
    //inicialização dos Leds Vermelhos e verdes  
  
    //Ecoponto Amarelo  
    digitalWrite(37, LOW); //Led verde  
    digitalWrite(39, HIGH); //Led vermelho  
    //Ecoponto Verde  
    digitalWrite(41, LOW); //Led verde  
    digitalWrite(43, HIGH); //Led vermelho  
    //Ecoponto Azul  
    digitalWrite(45, LOW); //Led verde  
    digitalWrite(47, HIGH); //Led vermelho  
  
}  
  
void leds( tipo_lixo)  
{  
    if (tipo_lixo==1)  
    {  
        digitalWrite(37, HIGH); //Led verde  
        digitalWrite(39, LOW); //Led vermelho  
    }else if (tipo_lixo==2)  
    {  
        digitalWrite(41, HIGH); //Led verde  
        digitalWrite(43, LOW); //Led vermelho  
    }else if (tipo_lixo==3)  
    {  
        digitalWrite(45, HIGH); //Led verde  
        digitalWrite(47, LOW); //Led vermelho  
    }  
}
```

## Anexo II

### Código para controlo do Leitor RFID e Leitura das tags RFID

```
uint8_t dataRX[35]; //Receive buffer.
uint8_t dataTX[35]; //Transmit buffer.
uint8_t _UID[4]; // stores the UID (unique identifier) of a card.
uint8_t keyAccess[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF } ; // stores the key or password.
uint8_t address = 0x04; //Address to read.
uint8_t ATQ[2]; //Answer to request
uint8_t state; //state of the process
uint8_t aux[16]; //Auxiliar buffer.
int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956, 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956};

void setup()
{
    //Start serial port 115200 bps:
    Serial.begin(115200);
    delay(10);
    Serial.print("RFID/NFC @ 13.56 MHz module started");
    delay(1000);
    tone(11, tones[6], 1000);
    //!It is needed to launch a simple command to synchronize
    getFirmware();
    configureSAM();
}

void loop()
{
    Serial.print("\n");
    Serial.println(" Aproxime o iSaco:");
    ////////////////////////////////////////////////////////////////////
    //Get the UID Identifier
    init(_UID, ATQ);
    Serial.print("\n");
    Serial.print( "Idi do iSaco: ###");
    print(_UID , 4);
    ////////////////////////////////////////////////////////////////////
    //Authenticate a block with his keyAccess
    state = authenticate(_UID, address, keyAccess);
    Serial.print("\n");

    if ( state == 0 ) {
        Serial.println("Idi do Bloco: OK");
        tone(11, tones[6], 1000); delay(35);
    } else {
        Serial.println("Authentication failed");
        tone(11, tones[1], 1000); delay(35);
    }
}
```

```

}
////////////////////////////////////
//Read from address after authentication
state = readData(address, aux);
Serial.print("\n");

if (state == 0) {
    Serial.println("Leitura do Bloco: OK");
    tone(11, tones[6], 1000); delay(35);
} else {
    Serial.println("Leitura do Bloco: Falhou");
    tone(11, tones[1], 1000); delay(35);
}

Serial.print("\n");
Serial.print("Info na Tag: *#*");
print(aux , 16);
Serial.print("\n");
delay(2000);
}
//*****
//!The goal of this command is to detect as many targets (maximum MaxTg)
// as possible in passive mode.
uint8_t init(uint8_t *UID , uint8_t *ATQ)    //!< Request InListPassive
{
    Serial.flush();

    dataTX[0] = 0x04;        //!

```

```

    if ((dataRX[9]== 0xD5) & (dataRX[10] == 0x4B) & (dataRX[11] == 0x01)) {
        return 0;
    } else {
        return 1;
    }
}
//*****
//!A block must be authenticated before read and write operations
uint8_t authenticate(uint8_t *UID, uint8_t blockAddress, uint8_t *keyAccess)
{
    dataTX[0] = 0x0F;
    lengthChecksum(dataTX);
    dataTX[2] = 0xD4;
    dataTX[3] = 0x40; // inDataEchange
    dataTX[4] = 0x01; //Number of targets
    dataTX[5] = 0x60; // Authentication code
    dataTX[6] = blockAddress;
    for (int i = 0; i < 6 ; i++) {
        dataTX[i + 7] = keyAccess[i];
    }
    dataTX[13] = UID[0]; dataTX[14] = UID[1];
    dataTX[15] = UID[2]; dataTX[16] = UID[3];
    dataTX[17] = 0x00;
    checksum(dataTX);
    sendTX(dataTX , 18 ,14);

    if ((dataRX[9]== 0xD5) & (dataRX[10] == 0x41) & (dataRX[11] == 0x00)) {
        return 0;
    } else {
        return 1;
    }
}
//*****
//!Write 16 bytes in address .
uint8_t writeData(uint8_t address, uint8_t *blockData) //!Writing
{
    Serial.print("      kkk      ");
    dataTX[0] = 0x15;
    lengthChecksum(dataTX); // Length Checksum
    dataTX[2] = 0xD4;
    dataTX[3] = 0x40;//inDataEchange CODE
    dataTX[4] = 0x01;//Number of targets
    dataTX[5] = 0xA0;//Write Command
    dataTX[6] = address; //Address
}

```

```

for (int i = 0; i < 16; i++) {
    dataTX[i+7] = blockData[i];
}
dataTX[23] = 0x00;
checksum(dataTX);
sendTX(dataTX , 24 ,14);

if ((dataRX[9]== 0xD5) & (dataRX[10] == 0x41) & (dataRX[11] == 0x00)) {
    return 0;
} else {
    return 1;
}
}
//*****
//!Read 16 bytes from address .
uint8_t readData(uint8_t address, uint8_t *readData) //!Reading
{
    Serial.print("A Ler: ....");

    dataTX[0] = 0x05;
    lengthChecksum(dataTX); // Length Checksum
    dataTX[2] = 0xD4; // Code
    dataTX[3] = 0x40; // Code
    dataTX[4] = 0x01; // Number of targets
    dataTX[5] = 0x30; //ReadCode
    dataTX[6] = address; //Read address
    dataTX[7] = 0x00;
    checksum(dataTX);
    sendTX(dataTX , 8, 30);
    memset(readData, 0x00, 16);

    if ((dataRX[9]== 0xD5) & (dataRX[10] == 0x41) & (dataRX[11] == 0x00)) {
        for (int i = 12; i < 28; i++) {
            readData[i-12] = dataRX[i];
        }
        return 0;
    } else {
        return 1;
    }
}
}

```

```

//!The PN532 sends back the version of the embedded firmware.
bool getFirmware(void)  //!< It is needed to launch a simple command to synchronize
{
    Serial.print(".....");

    memset(dataTX, 0x00, 35);
    dataTX[0] = 0x02; // Length
    lengthChecksum(dataTX); // Length Checksum
    dataTX[2] = 0xD4; // CODE
    dataTX[3] = 0x02; //TFI
    checksum(dataTX); //0x2A Checksum
    sendTX(dataTX , 5 , 17);
    Serial.print("\n");
    Serial.print("Your Firmware version is : ");
    for (int i = 11; i < (15) ; i++){
        Serial.print(dataRX[i], HEX);
        Serial.print(" ");
    }
    tone(11,tones[7],1000);
    Serial.print("\n");
}

//*****
//!Print data stored in vectors .
void print(uint8_t * _data, uint8_t length)
{
    for (int i = 0; i < length ; i++){
        Serial.print(_data[i], HEX);
        Serial.print(" "); //espaço entre os bites
        tone(11,tones[i],1000); delay(35);
    }
    Serial.print("\n");
}

//*****
//!This command is used to set internal parameters of the PN532,
bool configureSAM(void)!! Configure the SAM
{
    Serial.print(" A Configurar.... ");

    dataTX[0] = 0x05; //Length
    lengthChecksum(dataTX); // Length Checksum
    dataTX[2] = 0xD4;
    dataTX[3] = 0x14;
    dataTX[4] = 0x01; // Normal mode
}

```

```

    dataTX[5] = 0x14; // TimeOUT
    dataTX[6] = 0x00; // IRQ
    dataTX[7] = 0x00; // Clean checksum position
    checksum(dataTX);

    sendTX(dataTX , 8, 13);
}
//*****
//!Send data stored in dataTX
void sendTX(uint8_t *dataTX, uint8_t length, uint8_t outLength)
{
    Serial.write(0x00);
    Serial.write(0x00);
    Serial.write(0xFF);

    for (int i = 0; i < length; i++) {
        Serial.write(dataTX[i]);
    }

    Serial.print(0x00);
    getACK();
    waitResponse();// Receive response
    getData(outLength);
}
//*****
//!Wait for ACK response and stores it in the dataRX buffer
void getACK(void)
{
    delay(5);
    waitResponse();
    for (int i = 0; i < 5 ; i++) {
        dataRX[i] = Serial.read();
    }
}
//*****
//!Wait the response of the module
void waitResponse(void)
{
    int val = 0xFF;
    int cont = 0x00;
    while(val != 0x00) { //Wait for 0x00 response
        val = Serial.read();
        delay(5);
        cont ++;
    }
}

```

```

    }
}
//*****
//!Get data from the module
void getData(uint8_t outLength)
{
    for (int i=5; i < outLength; i++) {
        dataRX[i] = Serial.read();//read data from the module.
    }
}
//*****
//!Calculates the checksum and stores it in dataTX buffer
void checkSum(uint8_t *dataTX)
{
    for (int i = 0; i < dataTX[0] ; i++) {
        dataTX[dataTX[0] + 2] += dataTX[i + 2];
    }
    byte(dataTX[dataTX[0] + 2]= - dataTX[dataTX[0] + 2]);
}
//*****
//!Calculates the length checksum and sotres it in the buffer.
uint8_t lengthCheckSum(uint8_t *dataTX)
{
    dataTX[1] = byte(0x100 - dataTX[0]);
}

```

## Anexo III

### Código para receção dos dados da balança

```

char a;
char b[6]={0,0,0,0,0,0};

int i=0;
int j=0;

void setup()
{ //inicialização da porta serie com o baudrate 2400
  Serial.begin(2400);
  Serial3.begin(2400);
}

void loop() {
  while (Serial3.available(>0)
  {
    // leitura da porta serie 3 os valores da balança
    a=Serial3.read();
    if(a=='\n')
    {
      Serial.println(j);
      for(i=j+1;i>=0;i--)
      {
        Serial.print(b[i]);
      }
      Serial.println();
      j=0;
    }
    else
    {
      if(a!='.')
      {
        b[0]='0';
        b[j+1]=a;
        j++;
      }
    }
  }
  Serial.flush();
}

```

## Anexo IV

## Código para controlo dos servomotores

```
#include <Servo.h>
void porta_frontal(int tipo_lixo)
{ //inicialização dos servos
  Servo myservol;
  Servo myservo2;
  Servo myservo3;
  int pos = 0;

  if (tipo_lixo==1){ // abertura da porta 1
    myservol.attach(6);
    for(pos = 0; pos < 180; pos += 1)
    {
      myservol.write(pos);
    }
    delay(1000);
    // fecho da porta 1
    for(pos = 180; pos>=20; pos-=1)
    {
      myservol.write(pos);
    }
  }else if (tipo_lixo==2){// abertura da porta 2
    myservo2.attach(5);
    for(pos = 0; pos < 180; pos += 1)
    {
      myservo2.write(pos);
    }
    delay(1000);
    // fecho da porta 2
    for(pos = 180; pos>=20; pos-=1)
    {
      myservo2.write(pos);
    }
  }else if (tipo_lixo==3){// abertura da porta 3
    myservo3.attach(4);
    for(pos = 0; pos < 180; pos += 1)
    {
      myservo3.write(pos);
    }
    delay(1000);
    // fecho da porta 3
    for(pos = 180; pos>=20; pos-=1)
    {
      myservo3.write(pos);
    }
  }
}
```

## Anexo V

### Código para controlo dos motores de corrente contínua

```

void setup()
{
  // Defenição das portas digitais como Output
  //Motor 1
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  //Motor 2
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  // Activar / Desactivar
  pinMode(9, OUTPUT);
  //Motor 3
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
}

void motor_cc (int tipo_lixo)
{
  if (tipo_lixo==1)
  { //Abertura da porta 1
    digitalWrite(13, LOW);
    digitalWrite(12, HIGH);
    int i=250;
    //Abertura suave da porta
    for (i=250; i<1; i--)
    {
      analogWrite(9, i);
      delay(12);
    }
    //Fecho da porta 1
    digitalWrite(13, HIGH);
    digitalWrite(12, LOW);
    i=250;
    for (i=250; i<1; i--)
    {
      analogWrite(9, i);
      delay(12);
    }
  }
  else if (tipo_lixo==2)
  { //Abertura da porta 2
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    int i=250;
    //Abertura suave da porta
    for (i=250; i<1; i--)
    {
      analogWrite(9, i);
      delay(12);
    }
  }
}

```

```
    //Fecho da porta 2
    digitalWrite(11, HIGH);
    digitalWrite(10, LOW);
    i=250;
    for (i=250; i<1; i--)
    {
        analogWrite(9, i);
        delay(12);
    }
}else if (tipo_lixo==3)
{
    //Abertura da porta 3
    digitalWrite(8, LOW);
    digitalWrite(7, HIGH);
    int i=250;
    //Abertura suave da porta
    for (i=250; i<1; i--)
    {
        analogWrite(9, i);
        delay(12);
    }
    //Fecho da porta 3
    digitalWrite(8, HIGH);
    digitalWrite(7, LOW);
    i=250;
    for (i=250; i<1; i--)
    {
        analogWrite(9, i);
        delay(12);
    }
}
```

## Anexo VI

### Código para envio dos dados via XBee

```
void setup() {  
  Serial2.begin(115200);  
  
}  
  
void envioxbee( int peso, int idecoponto, int idtag)  
{  
  int checkdigit = peso+idecoponto+idtag;  
  
  String string = String('#'+idtag+'#'+peso+'#'+idecoponto+'#'+checkdigit+'#');  
  
  Serial2.println(string);  
  
}
```