Co-funded by the
Erasmus+ Programme
of the European Union

UNIDADE 1
Primeiros Programas

POLITEKNIKA IKASTEGIA TXORIERRI S.COOP.
isep Instituto Superior de Engenharia do Porto
TEI of Crete
Technological Educational Institute of Crete
apro

Objetivo e Conteúdos da Unidade 1

Objetivo

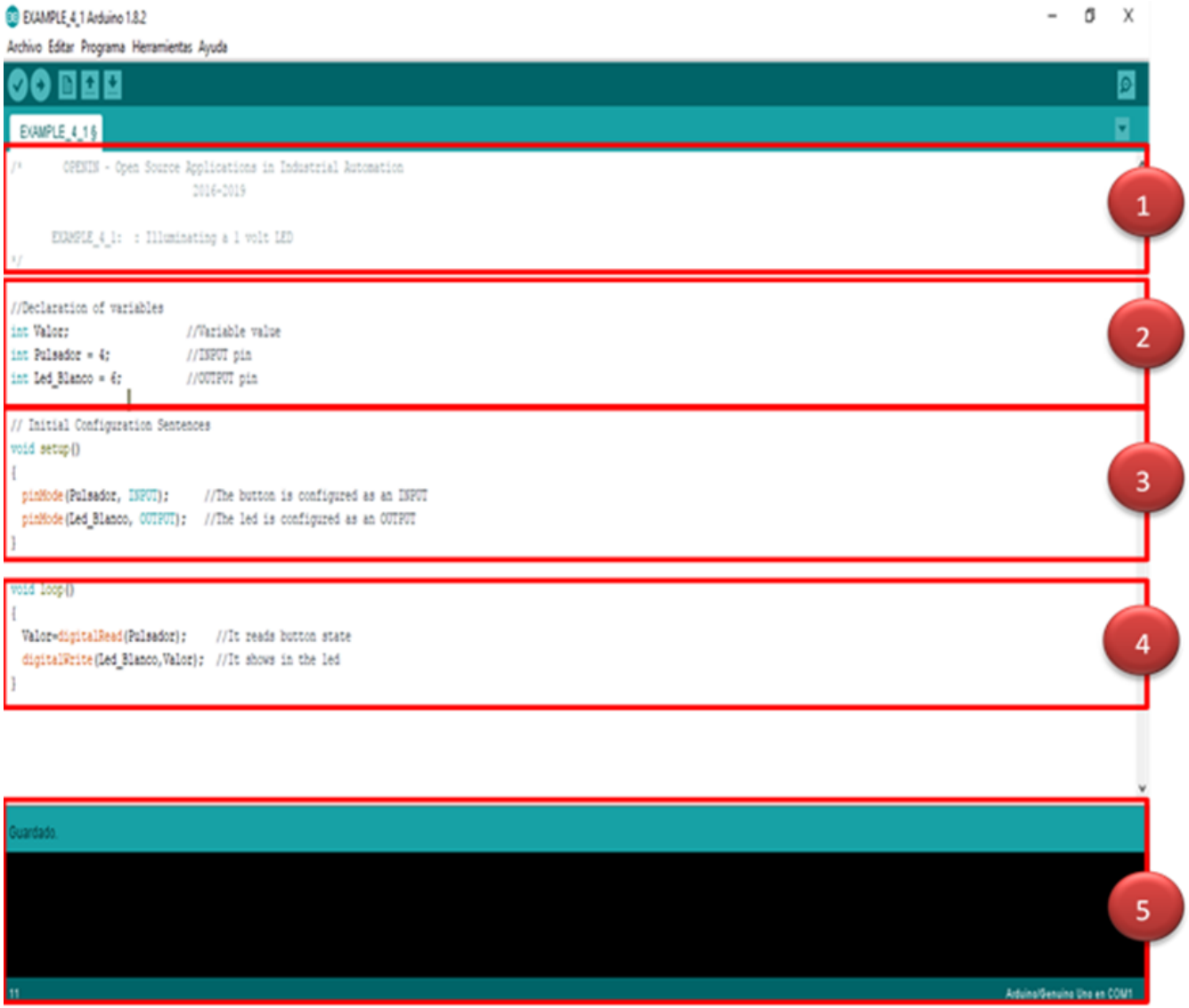
Criar alguns programas iniciais que permitam trabalhar rapida e facilmente com input e output (I/O) digitais. Vamos abordar os comandos básicos necessários para a utilização inicial de input e output digitais;

Conteúdos

- Explicar em que consiste um programa
- Apresentar comandos de programas
- Apresentar e explorer os operadores lógicos

Co-funded by the Erasmus+ Programme of the European Union

EM QUE CONSISTE UM PROGRAMA



Co-funded by the
Erasmus+ Programme
of the European Union

EM QUE CONSISTE UM PROGRAMA

- <u>SECÇÃO DE COMENTÁRIOS</u>

  - ✓ Todos os programas devem começar por fornecer determinadas informações
  - ✓ Esta informação denomina-se “Comentários Iniciais”
  - ✓ Pode incluir todos os comentários que quiser desde que apareçam entre os símbolos “/*” e “*/”

EXAMPLE_4_1

```
/*       OPENIN - Open Source Applications in Industrial Automation
                              2016-2019


       EXAMPLE_4_1:  : Illuminating a 1 volt LED
*/
```

1

Co-funded by the Erasmus+ Programme of the European Union

EM QUE CONSISTE UM PROGRAMA

- DECLAÇÃO DE VARIÁVEL E FUNÇÃO

  - ✓ Utilize esta segunda secção para declarar as variáveis e funções
  - ✓ Pense na variável como uma espécie de caixa ou recetáculo a que atribui um nome e possivelmente um valor também
  - ✓ De uma forma geral, quer as variáveis quer as funções têm de ser decladas ANTES de serem utilizadas no programa.

```
//Declaration of variables
int Valor;                  //Variable value
int Pulsador = 4;           //INPUT pin
int Led_Blanco = 6;         //OUTPUT pin
```

2

Co-funded by the Erasmus+ Programme of the European Union

EM QUE CONSISTE UM PROGRAMA

- TAREFA DE CONFIGURAÇÃO

  - ✓ Programas escritos em linguagem Arduino começam por executar algumas instruções ou funções de configuração.
  - ✓ De uma forma geral, os comandos ou funções de configuração só se executam uma vez quando o sistema faz RESET ou quando é ligado a uma fonte de energia.

```
// Initial Configuration Sentences
void setup()
{
  pinMode(Pulsador, INPUT);      //The button is configured as an INPUT
  pinMode(Led_Blanco, OUTPUT);   //The led is configured as an OUTPUT
}
```

3

Co-funded by the Erasmus+ Programme of the European Union

EM QUE CONSISTE UM PROGRAMA

- CORPO PRINCIPAL DO PROGRAMA

  - ✓ Todas as instruções, comandos e funções que constituem o seu programa têm de estar escritas nesta secção.
  - ✓ O controlador executa o mais rapidamente possível todas as funções que constituem o corpo principal do programa. Executa-as constantemente e indefinidamente da primeira à última.
  - ✓ Importante: Ao se declararem as variáveis e a configuração e as principais funções, certifique-se que terminam SEMPRE com este símbolo: ";".

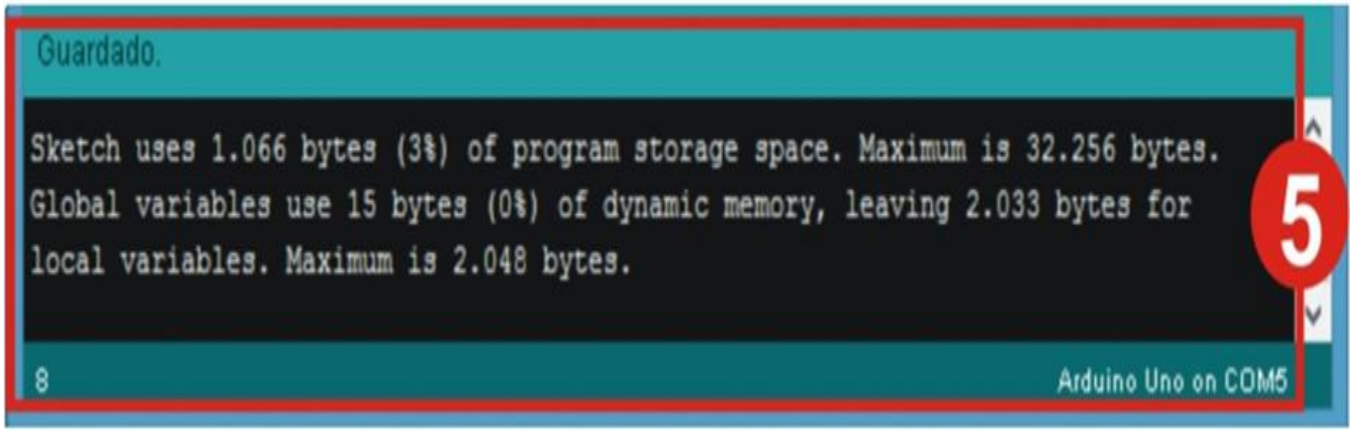
```
void loop()
{
  Valor=digitalRead(Pulsador);     //It reads button state
  digitalWrite(Led_Blanco,Valor);  //It shows in the led
}
```

4

Co-funded by the Erasmus+ Programme of the European Union

EM QUE CONSISTE UM PROGRAMA

- SECÇÃO DE MENSAGENS
  - ✓ Informa se está a guardar, a compilar ou a graver um programa na memória do controlador
  - ✓ Também informa se existem erros de compilação, o tipo de erro e onde se encontram
  - ✓ "Compilar" um programa significa traduzir o que escreveu em linguagem complexa Arduino para código binário (também conhecido como linguagem máquina); isto é qo que efetivamente fica gravado na memória FLASH do controlo



Co-funded by the Erasmus+ Programme of the European Union

COMANDOS DE PROGRAMAS

- A FUNÇÃO SETUP()

  - ✓ Esta função e todas as outras funções **nela inseridas são executadas APENAS quando o Sistema é reiniciado**

  - ✓ Existe habitualmente um número de outras funções inseridas nesta: configuração de pin de input e output, determindas variáveis de definição, bibliotecas, etc.

  - ✓ Pode integrar qualquer comando e funções que desejar na função setup() desde que **incluídas entre chavetas: "{...}"**

Sintaxe:

```
Void setup()
{
.......
}
```

Co-funded by the Erasmus+ Programme of the European Union

COMANDOS DE PROGRAMAS

- A FUNÇÃO PINMODE()

  - ✓ Configura um dos pins de controle do Arduino como um input ou como um output
  - ✓ Normalmente aparece no início de um programa e está incluída na função setup()
  - ✓ Todos os pins digitais são automaticamente configurados como inputs quando o sistema é reiniciado.

Sintaxe:

pinMode(pin,mode);

pin: É o número do pin que se vai configurar; pode encontrar-se entre 0 e 13 no Arduino Mode: Estabelece se trabalha como INPUT ou como OUTPUT

Co-funded by the Erasmus+ Programme of the European Union

COMANDOS DE PROGRAMAS

- A FUNÇÃO DIGITALREAD()

  ✓ Esta função lê e devolve o estado binário lógico (“1” ou “0”, “HIGH” ou “LOW”) de qualquer um dos pins de controle do Arduino

Sintaxe:

digitalRead(pin);

pin: Apresenta o número de pins que vamos ler; pode encontrar-se entre 0 e 13 no Arduino

Co-funded by the Erasmus+ Programme of the European Union

COMANDOS DE PROGRAMAS

- A FUNÇÃO DIGITALWRITE()

  ✓ Escreve ou define um valor binário (“1” ou “0”, “HIGH” ou “LOW”) através de um **pin de output**

Sintaxe:

digitalWrite(pin, value);

pin: Mostra o número do pin que vamos ler; pode estar definido entre 0 e 13 no Arduino
Value: Indica o valor a ser definido (“1” ou “0”, “HIGH” ou “LOW”).

Co-funded by the Erasmus+ Programme of the European Union

OPERADORES LÓGICOS

- O OPERADOR *NOT*

  ✓ Este operador **expressa negação (NOT)** e representa-se por um **ponto de exclamação (!)**

Exemplo:

Value = ! digitalRead(12);

Value é igual ao nível 1 se → pin 12 no nível 0
Value é igual ao nível 0 se → pin 12 no nível 1

Co-funded by the Erasmus+ Programme of the European Union

OPERADORES LÓGICOS

- O OPERADOR *AND*

  - ✓ This operation generates a level “1”, also known as “true”, **when ALL the elements** that you relate to each other are also **at level “1”**

  - ✓ It’s **represented** by these symbols: “**&&**”

Exemplo:

Value =digitalRead(4) && digitalRead(8) && digitalRead(12);

Value equals level 1 if → pins 4,8 and 12 are at level 1
Value equals level 0 if → pins 4,8 or 12 are at level 0

Co-funded by the Erasmus+ Programme of the European Union

OPERADORES LÓGICOS

- O OPERADOR *OR*

  - ✓ Esta operação gera um nível "1", também conhecido como "true", **quando QUALQUER UM dos** elementos que interrelacionar também se encontrarem no **nível "1"**

  - ✓ É **representado** por estes símbolos: "**||**"

Exemplo:

Value =digitalRead(4) || digitalRead(8) || digitalRead(12);

Value é igual ao nível 1 se → pins 4, 8 ou 12 estão no nível 1
Value é igual a 0 se → pins 4, 8 e 12 estão no nível 0

Co-funded by the Erasmus+ Programme of the European Union

OPERADORES LÓGICOS

- COMBINAR OPERADORES

  - ✓ Pode combinar vários operadores lógicos na mesma função
  - ✓ Tem de usar parênteses para estabelecer em que têm de ser acedidos e calculados

Exemplo:

Value = (digitalRead(8) && ! digitalRead(12)) || digitalRead(4));

Value igual ao nível 1 se → pin 8 no nível 1 e pin 12 no nível 0 ou pin 4 no nível 1

Value igual ao nível 0 se → pin 8 no nível 0 ou pin 12 no nível 1 e pin 4 no nível 0

Co-funded by the Erasmus+ Programme of the European Union

Co-funded by the
Erasmus+ Programme
of the European Union

UNIDADE 1: Primeiros programas

Obrigado!

Politeknika Ikastegia Txorierri S.Coop.
isep Instituto Superior de Engenharia do Porto
TEI of Crete
Technological Educational Institute of Crete
apro