

| | | | | | | | |
|--------------------|------------------------------|----------|--------|---------------------------|-----------|----------|----|
| Curso | Engenharia Informática | | | Ano letivo | 2018/2019 | | |
| Unidade Curricular | Arquitectura de Computadores | | | ECTS | 6 | | |
| Regime | Obrigatório | | | | | | |
| Ano | 2º | Semestre | 2º sem | Horas de trabalho globais | | | |
| Docente (s) | Gonçalo Marques | | | Total | 168 | Contacto | 75 |
| Coordenador | António Martins | | | | | | |

GFUC previsto

1. OBJETIVOS DE APRENDIZAGEM

Após a conclusão da UC, os alunos deverão ser capazes de:

1. *Perspetivar a evolução futura dos computadores com base no conhecimento da sua evolução até aos dias de hoje.*
2. *Descrever as operações básicas matemáticas e lógicas e diferentes formas de representação de quantidades numéricas e não numéricas.*
3. *Identificar a arquitectura geral dos computadores.*
4. *Desenvolver algoritmos e implementar programas e/ou procedimentos em linguagens de baixo nível e comparar o seu desempenho com programas e/ou procedimentos implementados em linguagens de alto nível.*
5. *Identificar diferentes técnicas de aumento da velocidade de processamento das aplicações, quer ao nível do hardware quer ao nível do software.*
6. *Utilizar microcontroladores para aquisição e controlo de sinais analógicos e digitais bem como efetuar a comunicação do microcontrolador com um computador.)*

2. CONTEÚDOS PROGRAMÁTICOS

1. *Breve história dos computadores*
2. *Revisão de conceitos fundamentais para arquitectura de computadores*
 - a. *Bits, Bytes, Words*
 - b. *Representação de números em diferentes bases*
 - c. *Operações com números em diferentes bases*

- d. Representação de números negativos*
- e. Floating points*
- f. Representação de dados não numéricos*
- 3. Arquitectura geral dos computadores**
 - a. Organização geral de um CPU*
 - b. Diferentes tipos de BUS*
 - c. Dispositivos de I/O*
- 4. Arquitectura 80XXX.**
 - a. Arquitecturas CISC/RISC*
- 5. Organização e funcionamento da memória**
 - a. Diferentes hierarquias de memória*
 - b. Modo real/modo protegido*
 - c. Memória virtual*
 - d. Memória cache*
 - e. Alinhamento da memória e a sua influência no desempenho*
- 6. Introdução à programação em baixo nível**
 - a. Instruções*
 - b. Integração de Assembly com linguagens de alto nível*
- 7. Técnicas de aumento da velocidade de processamento**
 - a. Pipeline*
 - b. SIMD*
 - c. Arquitectura superescalar*
 - d. Previsão de saltos*
 - e. Execução especulativa*
- 8. Introdução aos microcontroladores**
 - a. Descrição do Arduino ao nível do hardware e software*
 - b. Desenvolvimento de aplicações de leitura de sinais analógicos e digitais*

- c. Desenvolvimento de aplicações de controlo de sinais analógicos e digitais.*
- d. Desenvolvimento de aplicações de comunicação com um computador*
- e. Utilização de interrupts: vantagens e limitações*

3. DEMONSTRAÇÃO DA COERÊNCIA DOS CONTEÚDOS PROGRAMÁTICOS COM OS OBJETIVOS DA UC

Os conteúdos 1, 3 e 4 permitem atingir o objetivo 1 da UC, uma vez que se mostra a evolução da arquitetura dos computadores, não só de uma forma genérica como também de uma forma específica.

O objetivo 2 é atingido pelo conteúdo 2 sendo um pilar essencial para atingir os objectivos seguintes.

O objetivo 3 é atingido pela conjugação de todos os conteúdos do programa uma vez que cada um deles tem uma contribuição para o conhecimento da arquitetura dos computadores.

O objetivo 4 é atingido pelos conteúdos 4, 5 e 6 uma vez que para desenvolver aplicações de baixo nível é necessário conhecer a arquitetura, o modelo de memória, e a linguagem de programação.

O objetivo 5 é atingido pelos conteúdos 4, 5, 6 e 7 uma vez que os mesmos enfocam as diferentes técnicas referidas nesse objectivo.

O objetivo 6 é atingido pelo conteúdo 8 que introduz os alunos, com aplicações práticas, no mundo dos microcontroladores.

4. BIBLIOGRAFIA PRINCIPAL

Obrigatória

Apontamentos do docente

Recomendado:

Arquitectura de Computadores de Guilherme Arroz, Arlindo Oliveira e José Monteiro

<http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>

<http://www.arduino.cc/>

5. METODOLOGIAS DE ENSINO (REGRAS DE AVALIAÇÃO)

Metodologias de ensino:

- 1. Lição expositiva*
- 2. Lição interativa*
- 3. Resolução de problemas*
- 4. Trabalho de projeto*

Regras de avaliação:

Avaliação contínua (Frequência e Exame):

- 1. 35% para o conjunto de 2 Trabalhos práticos de grupo.*
- 2. 65% para a frequência a realizar no final do semestre. A nota mínima é de 7 valores.*

Avaliação por exame final na Época de Recurso ou Época Especial:

- 1. 100% para o exame teórico.*

6. DEMONSTRAÇÃO DA COERÊNCIA DAS METODOLOGIAS DE ENSINO COM OS OBJETIVOS DA UNIDADE CURRICULAR

1. Lição expositiva está coerente com os objetivos devido à necessidade de apresentar os conteúdos teóricos aos alunos, para que depois possam começar a desenvolver os trabalhos práticos da disciplina. Enquadram-se especialmente neste ponto os objetivos 1, 3 e 5.

2. *Lição interativa está coerente com os objetivos pois a demonstração prática de uma determinada solução potencia a procura de soluções para novos problemas. Serão particularmente abrangidos por este tipo de lição os pontos 2 e 4.*

3. *Resolução de problemas está coerente com os objetivos uma vez que é absolutamente fundamental os alunos fazerem os seus próprios programas para não só consolidarem os conhecimentos teóricos apreendidos, como também para ganharem experiência no saber fazer e não apenas no saber como se faz. Os pontos 4 e 8 dos objetivos têm aqui a sua principal forma de ser atingido.*

4. *Trabalho de projeto está coerente com os objetivos pois abrange o desenvolvimento de uma aplicação que permite aos alunos exercitar as suas capacidades de conceção e realização prática, o que implica o desenvolvimento de algoritmos e a implementação de programas e/ou procedimentos tal como conta do objetivo 4.*

7. CONTACTOS E HORÁRIO DE ATENDIMENTO

E-mail: goncalosantosmarques@gmail.com

8. OUTROS

TRABALHO INTERMÉDIO

Elaborar um programa para o Arduino que utilize interrupts.

TRABALHO FINAL

Elaborar um programa para o Arduino que implemente uma estação meteorológica e faça a publicação dos dados na internet.

Regras Gerais

O trabalho final obrigatório será escolhido pelos alunos de entre a lista proposta ou, em casos excepcionais, e com o acordo do docente, poderá ser um outro proposto pelos alunos dando-se preferência a trabalhos que utilizem o Arduino. Os relatórios destes trabalhos devera ser entregue 2 dias úteis antes da data da frequência. A sua apresentação e defesa será feita no dia da frequência tendo cada grupo 10 minutos para fazer essa apresentação.

Os trabalhos serão feitos em grupos com o máximo de 3, não sendo permitido, sob qualquer pretexto, a formação de grupos maiores. Apesar do trabalho ser em grupo a avaliação será individual.

Todas as situações detectadas de cópia, mesmo que parcial, de relatórios e/ou trabalhos entre grupos implicará a anulação dos respectivos trabalhos. A mesma situação se verificará se a cópia tiver como origem trabalhos de anos anteriores.

Chama-se a atenção para as questões de plágio de livros ou de informações na Internet. Sempre que seja necessário fazer uma transcrição ela deverá ser feita em “itálico”, com aspas, e com uma referência clara para a origem do texto.

Data:

O regente da UC:

(Gonçalo Marques)

O coordenador da área:

(António Martins)