Parecer favorável, por unanimidade,
na reunião do CTC de 13/09/2023.
O Presidente do CTC,

Assinado por: **JOÃO ALFREDO FERREIRA DOS SANTOS**
Num. de Identificação: 06254943
Data: 2023.09.14 15:26:15+01'00'

Ficha de Unidade Curricular – Programação II

Assinado por: **CARLOS EDUARDO DE MENESES RIBEIRO**
Num. de Identificação: 06906457
Data: 2023.08.28 10:33:32+01'00'

Assinado por: **Pedro Manuel de Almeida Carvalho Vieira**
Num. de Identificação: 10302000
Data: 2023.08.29 10:56:32 +0100

Assinado por: **MÁRIO PEREIRA VÉSTIAS**
Num. de Identificação: 09139891
Data: 2023.08.29 19:53:45+01'00'

Aprovada pelo CP. A presidente do CP
Assinado por: **SANDRA MARIA DA SILVA FIGUEIREDO ALEIXO**
Num. de Identificação: 07789825
Data: 2023.08.30 13:21:59 +0100

CHAVE MÓVEL

1. Caracterização da Unidade Curricular.

1.1. Designação da unidade curricular *(1.000 carateres).*
Programação II / Programming II

1.2. Sigla da área científica em que se insere *(100 carateres).*
IC

1.3. Duração[1] *(100 carateres).*
Semestral

1.4. Horas de trabalho[2] *(100 carateres).*
160

1.5. Horas de contacto[3] *(100 carateres).*
67,5 (T: 45; PL: 22,5)

1.6. ECTS *(100 carateres).*
6

1.7. Observações[4] *(1.000 carateres).*
UC comum a LEETC e LEIRT

1.7. Remarks *(1.000 carateres).*
Common to LEETC and LEIRT

2. Docente responsável e respetiva carga letiva na Unidade Curricular *(preencher o nome completo) (1.000 carateres).*
Mário Henrique Carrasqueira Simões – 67,5 horas de contacto

3. Outros docentes e respetivas cargas letivas na unidade curricular *(1.000 carateres).*
Miguel Pinto Campilho Gomes – 67,5 horas de contacto
Luís Manuel Dias Damas – 67,5 horas de contacto

4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes). *(1.000 carateres)*.
Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:
1. Implementar, testar e corrigir programas de complexidade média, organizados em módulos escritos em linguagem C, podendo envolver o uso de bibliotecas, normalizadas ou outras;
2. Relacionar os mecanismos da linguagem C com o modelo computacional que suporta a execução;
3. Desenvolver estruturas de dados dinâmicas, recorrendo à definição de tipos e ao alojamento dinâmico de memória, e algoritmos, caraterizando o respetivo custo computacional;
4. Desenvolver e utilizar implementações genéricas de algoritmos, aplicáveis a diversos tipos de dados, com recurso à parametrização das operações a realizar;
5. Identificar e explorar as funcionalidades das bibliotecas normalizadas;
6. Descrever como os programas são compilados, ligados, carregados e executados.

4. Intended learning outcomes (knowledge, skills and competences to be developed by the students). *(1.000 characters).*
Students who successfully complete this course unit will be able to:
1. Implement, test and correct programs of medium complexity, organized in several modules written in C language, and that may involve the use of libraries;
2. Relate the mechanisms of the C language to the computational model that supports execution;
3. Develop dynamic data structures, using the definition of types and dynamic allocation of memory, and algorithms, characterizing their computational cost;

4. Develop and use generic implementations of algorithms, applicable to several types of data, using the parameterization of the operations to be performed;
5. Identify and exploit the functionalities of the standard libraries;
6. Describe how programs are compiled, linked, loaded, and executed.

5. Conteúdos programáticos *(1.000 carateres).*

I. Gestão da memória nos programas escritos em linguagem C (alojamento permanente, variáveis automáticas e alojamento dinâmico). Revisões de ponteiros e sua aritmética. Programação genérica. Programação recursiva.
II. Organização dos programas em módulos. Compilação, ligação e execução de programas. Conceito de pré-compilação. Criação e utilização de macros e de ficheiros para inclusão. Automatização do processo de compilação e ligação.
III. Utilização da biblioteca normalizada, famílias de funções e suas funcionalidades. Ligação estática e dinâmica. Uso de outras bibliotecas. Criação de bibliotecas.
IV. Algoritmia e estruturas de dados. Criação dinâmica de estruturas de dados – arrays com alojamento dinâmico, listas ligadas, árvores binárias e hash-table. Algoritmos de ordenação. Inserção e remoção ordenadas. Balanceamento das árvores binárias.

5. Syllabus *(1.000 characters).*

I. Memory management in programs written in C language (permanent allocation, automatic variables, and dynamic allocation). Revisions of pointers and their arithmetic. Generic programming. Recursive programming.
II. Organization of programs in modules. Compilation, linkage, and execution of programs. Pre-compilation concept. Creating and using macros and inclusion header files. Automation of the compilation and linkage process.
III. Use of the standard library, function families, and their functionalities. Static or dynamic linkage. Use of other libraries. Creation of libraries.
IV. Algorithms and data structures; Dynamic creation of data structures - arrays with dynamic allocation, linked lists, binary search trees and hash-table. Sorting algorithms. Sorted insertion and removal. Binary tree balancing.

6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular *(1.000 carateres).*

Os temas são introduzidos por ordem crescente de complexidade. Os objetivos 1 e 2 são suportados pelo somatório dos conhecimentos adquiridos em todos os temas. O objetivo 3 é suportado pelo tema IV. O objetivo 4 é suportado pelos temas I e III. O objetivo 5 é suportado pelos temas II e III. O objetivo 6 é suportado pelo tema II.

6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes *(1.000 characters).*

Topics are introduced in increasing order of complexity. Objectives 1 and 2 are supported by the sum of the knowledge acquired in all themes. Objective 3 is supported by the theme IV. Objective 4 is supported by the themes I and III. Objective 5 is supported by the themes II and III. Objective 6 is supported by the theme II.

7. Metodologias de ensino (avaliação incluída) *(1.000 carateres).*

O ensino é teórico (45h) e prático (22,5h). Na parte teórica há exposição de conceitos, exemplos de aplicação e exercícios de consolidação. Na parte prática há realização de exercícios, por parte dos estudantes, e apoio à realização dos trabalhos práticos.
A avaliação tem as componentes escrita (E) e prática (P).
A componente E pode ser distribuída (1.º teste, T1; 2.º teste, T2) ou por exame. T1 é realizado na fase letiva. Na época normal de exame, o estudante pode optar por T2, se obteve o mínimo em T1, ou exame. Na época de recurso, pode optar por exame parcial de T2 ou exame.
A componente P é baseada em 3 trabalhos práticos (P1, P2, P3) e avalia o desempenho através da respetiva discussão, realizada em grupo, sendo atribuídas classificações individuais.
A classificação final é: CF=0,5E + 0,5P.
Os mínimos para aprovação são: CF>=9,5; E >=9,5; P >= 9,5.
T1 >= 8,0 e T2 >= 8,0; E = 0,5T1+0,5T2. P=0,3P1+0,3P2+0,4P3.

7. Teaching methodologies (including assessment) *(1.000 characters).*

Teaching is theoretical (45h) and practical (22.5h). The theoretical part exposes concepts, application examples

and consolidation exercises. In the practical part, there are exercises carried out by the students, and support for carrying out practical work.
The assessment has written (E) and practical (P) components.
The E component can be distributed (1st test, T1; 2nd test, T2) or per exam. T1 is carried out in the class phase. In the normal exam period, the student could opt for T2, if he obtained the minimum in T1, or exam. At the time of appeal, you can opt for partial T2 exam or exam.
The P component is based on 3 practical assignments (P1, P2, P3) and evaluates the performance through the respective discussion, carried out in groups, with individual classifications being attributed.
The final classification is: CF=0.5E + 0.5P.
The minimums for approval are: CF>=9.5; E>=9.5; P >= 9.5.
T1 >= 8.0 and T2 >= 8.0; E=0.5T1+0.5T2. P=0.3P1+0.3P2+0.4P3.

8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular *(3.000 carateres).*

Os estudantes obtiveram numa unidade curricular anterior os conceitos gerais de programação, usando a linguagem C, e estão a obter noutra unidade curricular os conceitos de arquitetura de computadores.
Nas aulas são apresentados os temas, de I a IV, com vista ao domínio dos objetivos referidos. A exposição é complementada, em interação com os estudantes, pela realização, teste e correção de programas exemplificativos dos temas expostos. Estes programas exemplificativos, além de concretizarem os conceitos de programação lecionados são usados para explorar as ferramentas de programação, com ênfase na metodologia de depuração de erros.
Os trabalhos práticos, têm complexidade crescente e são alinhados com os temas das aulas, que fornecem conceitos de base para a sua realização. Nos trabalhos, os estudantes confrontam-se com problemas e dúvidas que, ao elaborar soluções, promovem o desenvolvimento das suas competências.
O plano de trabalhos práticos está estruturado em séries de exercícios (SE) organizadas por grupos de temas. A escolha dos temas agrupados nas sucessivas séries de exercícios é orientada para promover os objetivos de aprendizagem de acordo com a relação indicada em 6.

8. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes *(3.000 characters).*

Students obtained in a previous curricular unit the general concepts of programming, using the C language, and are acquiring the concepts of computer architecture in another curricular unit.
In classes, the themes are presented, from I to IV, with a view to the mastery of the referred objectives. The exposition is complemented, in interaction with the students, by the accomplishment, test and correction of programs exemplifying the exposed themes. These programs, in addition to realizing the programming concepts taught, are used to explore the programming tools, with an emphasis on the methodology of debugging.
The practical work, to be carried out by the students, has increasing complexity, and is coordinated with the themes of the classes, which provide basic concepts for its accomplishment. In the laboratory works, the students develop their competences though the contact with the problems raised by the exercises.
The laboratory work plan is structured in series of exercises (SE) organized by groups of themes. The choice of the themes grouped in the successive series of exercises is driven to promote the learning objectives according to the relationship indicated in 6.

9. Bibliografia de consulta/existência obrigatória *(1.000 carateres).*

C Programming - A Modern Approach, 2nd Ed.; K. N. King; W. W. Norton, 2008;
ISBN 9780393979503

Introduction to Algorithms, Third Edition; Cormen, Leiserson, Rivest, Stein; The MIT Press, 2009;
ISBN 9780262033848/9780262533058

The C Programming Language, 2nd Ed.; Dennis M. Ritchie, Brian W. Kernighan; Prentice Hall, 1988;
ISBN: 9780133086249/9780131103627

Algorithms in C, Parts 1-4: Fundamentals (Data Structures, Sorting, Searching) 3rd Ed.; Robert Sedgewick;
Addison-Wesley Professional, 1997;
ISBN: 9780201314526

[1] Anual, semestral, trimestral, …
[2] Número total de horas de trabalho.
[3] Discriminadas por tipo de metodologia adotado (T - Ensino teórico; TP - Ensino teórico-prático; PL - Ensino prático e laboratorial; TC - Trabalho de campo; S - Seminário; E - Estágio; OT - Orientação tutorial; O - Outro).
[4] Assinalar sempre que a unidade curricular seja optativa.