

Learning Data Structures – same difficulties in different countries?

Beatriz Pérez-Sánchez and Paula Morais

Abstract— Data structures have been a core discipline in Computer Science studies. Several difficulties related to teaching and learning of these contents have been detected by the academic community. With the aim of obtaining a better knowledge about these situations we present the conclusions obtained after evaluating the results achieved by the students of the subject Programming II, in the first course of the Degree in Computer Engineering at University of A Coruña, UDC, (Spain) and the students of the subject Data Structures where similar contents are taught in the Computer Science Degree at Portucalense University, UPT, (Portugal).

Index Terms — *absenteeism, learning strategy, assessment, performance, data structures, abandonment.*

I. INTRODUCTION

For years Data Structures has been a core discipline in undergraduate computer curriculum, mainly because of the fundamental nature of its content. Over the years it has been recognized significant difficulty by students, in their learning mainly due to the abstract nature of the subject and its underlying programming component. Different studies refer to various difficulties among which are: the absence of an appropriate method of work by students, the complexity in designing algorithms to manipulate data structures or the use of unsuitable pedagogic methods by teachers [1],[2]. Such circumstances lead to two problems of vital importance, abandonment of matter by students (absenteeism) and as a result, a failure rate that can be considered high. In order to achieve a better understanding of these situations and keeping the focus of interest in increasing learning success of this subjects, this article presents a comparison of the results achieved by students enrolled in the Programming II course, belonging to the first degree in Computer Engineering from the University of a Coruña, UDC, (Spain) and the course of Data Structures, which provides similar content in the Bachelor of Computer Science at the University Portucalense, UPT, (Portugal).

It is noteworthy that both entities collaborate since 2011, thanks to the Lifelong Learning Erasmus Programme for staff mobility. The aim of this program is to make easy the transfer of knowledge and experiences among teachers through the realization of educational visits to other reference institutions.

This article is structured as follows: section II details the underlying theoretical foundations of the study; section III presents the main information of interest in both institutions; section IV presents a comparative analysis of the results, focusing on two main issues success rates and abandonment,

for 2011/2012, 2012/2013 and 2013/2014 courses; and finally in section V the main conclusions are exposed.

II. CONTEXT. TEACHING AND LEARNING DATA STRUCTURES

Data Structures form one of the core elements of undergraduate computer curriculum, since many years. Association for Computer Machinery (ACM), the oldest computing scientific and educational society, has been involved, since 1968, in the development of curriculum recommendations. In that year it published the text “Curriculum68: Recommendations for the undergraduate program in computer science” [3], in which a course named Data Structures was included. Also in the 1978 text, ACM describes the CS2 course: *Computer Programming II* [4].

Since then all ACM recommendations (Computing Curricula) have had a wide influence, being used as a guide for the design of computer curricula worldwide. Despite all the changes proposed in the ACM recommendations for Computer Science curricula, study data structures has remained mandatory.

Over the years there have been various problems associated with the teaching and learning of data structures and different issues have been raised by the educational community. Among them worth mentioning some of the most significant:

How to help improve the learning process? Students encounter difficulties when dealing with the contents due to the abstract nature of the subject. In order to facilitate the learning process different tools to visualize the data structures have been proposed over the years [5] - [7].

What should be taught to students? Does the teaching should focus on the details of each data structure? Or maybe would it be convenient to focus learning on their use and application? Some theories argue that, despite the importance of the student knows how to build data structures, knowing when and how to use them is a priority [8], [9].

How to make algorithms programming for manipulating data structures easy? In a discipline of data structures students not only have to know the basic theoretical concepts but also acquire skills that enable them to select the most suitable data structures for solving a given problem and how to program them.

Usually the topic of programming is an added difficulty for students. Different strategies and tools to mitigate these effects have been presented [10] - [12].

What programming language shell be used for the study of data structures? It is a controversial issue that has been a source of debate throughout the years. Already in 1997 Weiss

described the experience of using Java in teaching the discipline of data structures [13]. Some authors even publish different versions of the same book, using different programming languages [14]-[17]. Such publications vary in the language used and focus the study of data structures in an abstract manner using pseudocode, which helps to focus solely on the data structures and algorithms associated with them without regard to implementation details. This experience allowed them to conclude that the language is merely an implementation tool. This reinforces our view that the issue of programming language selection is not relevant. While the topic is widely discussed in the literature, the institutions apply various alternatives (Java, C / C++ or other).

III. DESCRIPTION

2010/2011 was the year of implementation of the Degree in Computer Engineering at the University of Coruña (UDC), adapted to the European Higher Education Area (EHEA). It was also the first year of introduction of the data structure contents in the curricular plan, in a discipline called Programming II (ProII).

The experience of introducing those contents in the referred course plan makes us fully aware of several potential problems. This concern led to make, just from the time of the inclusion of the new material, an intensive monitoring of academic results achieved by students, in order to draw meaningful conclusions, and evaluate both its performance and our teaching skills.

In this way we can know the main weaknesses in order to tackle them, as far as possible, with the appropriate corrective actions.

ProII is a first year course belonging to the Programming Languages block from the Basic Training Module of the degree in Computer Engineering, with a total of 6 ECTS credits associated. It maintains a strong dependence on matters of Programming I (ProI), Discrete Mathematics and Basic Computer taught in the previous semester. The contents that in UDC are assigned to the discipline of ProII are taught in subjects with different names depending on each university and appear positioned in different ways in curricula, including: (i) as a course of between 6 and 7.5 ECTS, taught in the second semester of the degree (in UDC), (ii) as part of a course of 12 ECTS credits, taught in the second semester of the degree, and also including the contents that are taught in ProI at UDC, (iii) as a course of between 6 and 7.5 ECTS, taught in the third semester (second year) of the degree.

After an in-depth assessment of the results, detected the main problem and, in order to meet the teaching approach at other universities, both in terms of skills, and organizational and evaluation, a collaboration with the Portuguese University situated in Porto (Portugal) emerged, through the Lifelong Learning Erasmus program for teachers mobility. Those responsible for the destination university offered to participate in the teaching of the subject Data Structures (DS), where analogous contents to ProII are taught.

The main difference we found was that DS is a second year, first semester course, with 6 ECTS credits. It maintains a strong dependence on Algorithms and Programming and

Object Oriented Programming courses, taught in the first academic year.

This situation will allow us to know if, at least in part, the problems identified in our university may be related to the status of delivery of the contents of interest in the curriculum. The teacher's mobilities, which were repeated in successive courses, allowed to know the planning and content structuring in the host university, share knowledge and experience and begin a collaboration that continues today.

In the next paragraphs we describe the learning objectives and the pedagogical and assessment methods used in each of the universities involved in the study.

A. Learning objectives

In order to provide a general idea of the contents taught in the courses ProII (UDC) and DS (UPT) and also to know what the expected results are a brief description of them is presented.

The main ProII contents can be summarized in three blocks:

- (1) Pointers and dynamic memory management,
- (2) Introduction to abstract data types (ADTs),
- (3) Programming using dynamic and complex data structures.

As learning outcomes, it is expected that students acquire the following skills:

- (a) To know and use dynamic memory management mechanisms,
- (b) To know ADT specification mechanisms,
- (c) To program algorithms for the manipulation of data structures designed to solve real problems (lists, stacks, queues and trees),
- (d) To internalizing good programming practices.

Meanwhile the curricular unit DS in UPT aims to introduce:

- (1) The basic concepts of abstract data structures,
- (2) The behavior associated according to the static and dynamic performances.

As learning outcomes, it is expected that students acquire the following skills:

- (a) To know the operation and understand the usefulness of the most relevant abstract data structures (lists, stacks, queues and trees)
- (b) To know how to select and use the appropriate data structures to solve a given problem,
- (c) To know how to implement data structures using dynamic memory management.

B. Methodology

The experience presented in this paper includes a comparative study of the results achieved in both universities considering the objectives mentioned above, dropout rates and success. To this end the results achieved during the 2011/2012, 2012/2013 and 2013/2014 courses are handled. In order to analyze in detail the results of both universities it is also of interest to know the teaching methodology used, that in both cases is based on students active learning.

With regard to UDC, teaching is divided into three sections: (1) lectures, (2) laboratory practice and (3)

troubleshooting seminars. In the theoretical classes the topics and objectives, with a fully practical guidance and requiring student collaboration are described. In the troubleshooting seminars practical assumptions for development, resolution and sharing are presented. This type of group activity promotes the development of useful skills for professional practice as communication and teamwork [18]. Laboratory practices require the construction and use of data structures in a high level programming language for solving small practical problems. This mandatory work is developed in groups to enhance trans-disciplinary skills such as teamwork, decision-making, organization and planning.

In UPT there are two main blocks: (1) theoretical/practical classes where the fundamental concepts of abstract data types are presented, often using interpellation with students and learning by cases, and (2) practical classes aim to develop specific skills, through the resolution of practical exercises on the computer that are addressed in groups of two students.

On the subject of the programming language used there are differences between institutions. Pascal is the language used in the UDC; the choice is justified only for practical reasons because this is the language used to introduce students in their first contact with programming in the ProI subject in the first semester. Since ProII is taught in the second semester and considering the difficulties students have in programming, it was considering not a good idea to introduce another language. An alternative might be some language widely used in the professional field as C, C ++, Java, C # or Python. This occurs at UPT where, after a first year of programming (in the first semester students use C in Algorithms and Programming and Java in Objet Oriented Programming, in the second semester) students have sufficient background to implement the various data structures in a language like Java. Anyway, as indicated previously, it is considered that the language is only an implementation tool.

C. Evaluation

This section describes the evaluation criteria applied in both universities, information of interest because it could influence the final grades achieved by students. In both entities a method of continuous evaluation is used, mainly based on practical assignments, developed both individual and pair wise. In both entities the evaluation involves three aspects: a theoretical written test, practical (made in pairs), and a series of practical tests that are individually resolved. However, the weighting associated with each of these items in the calculation of the final mark differs considerably in both institutions.

In UDC the assessment of theoretical written test is very high, 75%, while the remaining 25% is split between the two other items at a rate of 20% practice, and 5% problem resolution. UPT gives the highest score to practical student assignments, thus only 30% is allocated to the written examination while the remaining 70% is distributed 40% to practice in pairs and 30% to individual practical work.

In both universities the practical work aims the resolution of a simplified version of a real problem using the data structures presented in theory, and its implementation using the referred programming languages. In the case of UPT

students perform 3 practical tests throughout the course, the first two in pairs and a last one individual. In the case of UDC students must perform two practices in pairs. In both institutions the results of each practice are provided to students in order to give the necessary feedback before the next test. In addition, it is worth mentioning that a minimum score on the practical test is required to pass the subject.

IV. RESULTS

Concerned about the results achieved by our students and with the aim of raising the necessary changes, we conducted an exhaustive study for three academic years, 2011/2012, 2012/2013 and 2013/2014. As previously commented along these years the main problems identified were: (a) the success rate and (b) the rate of students absent from the tests. Thus, in this section we address both issues in detail based on the results. In order to shed some light on the difficulties encountered and delving into the origin of them, a comparative analysis with the results achieved in the UPT is provided. It is noteworthy that the number of students enrolled in the UDC is considerably higher than the UPT consequently the number of student groups is higher in the case of the UDC however, in both institutions the average number of students in each of these groups is similar (around 25-30 students).

A. Success Rate

As a first step in the analysis of results achieved by the students, we focus on grades. Fig. 1 shows a comparison of the scores achieved by the students presented in both universities. First we focus on the results of the UDC. As shown in Fig. 1 for the year 2011/2012 a 30.69% of the presented students fails to pass the course, a 40.91% achieves the minimum score (approved), a 23.29% get a remarkable and only 5.11% reaches the highest rating. During 2012/2013 the distribution of students that succeed is mainly divided between approved (32.10%) and remarkable (26.30%), only 4.50% of students reach the highest rating. Finally, in the course 2013/2014 results follow the same line; from the 37.03% students presented in the tests only 5.44% obtained the maximum rating, 31.29% a remarkable and 25.17% just approved.

As shown in Fig. 1, the main difference from the results of UPT is in the percentage of students reaching the highest rating. During 2011/2012 only 5.11% of students presented at the UDC reach the highest rating compared to 25% who get it in the UPT.

During 2012/2013 the difference is smaller but still significant, 4.50% of the UDC against 13.33% of the UPT. The biggest difference is seen in the course of 2013/2014, although in this case, in a timely manner the difference is favorable to the UDC.

Regarding the number of students who fail to pass the subject, Fig. 1 shows as these percentages are high in both institutions (between 30% and 40%) and relatively similar with the exception of 2012/2013, where a substantial difference, 53.33% in the UPT appreciated against the 37.20% obtained in the UDC.

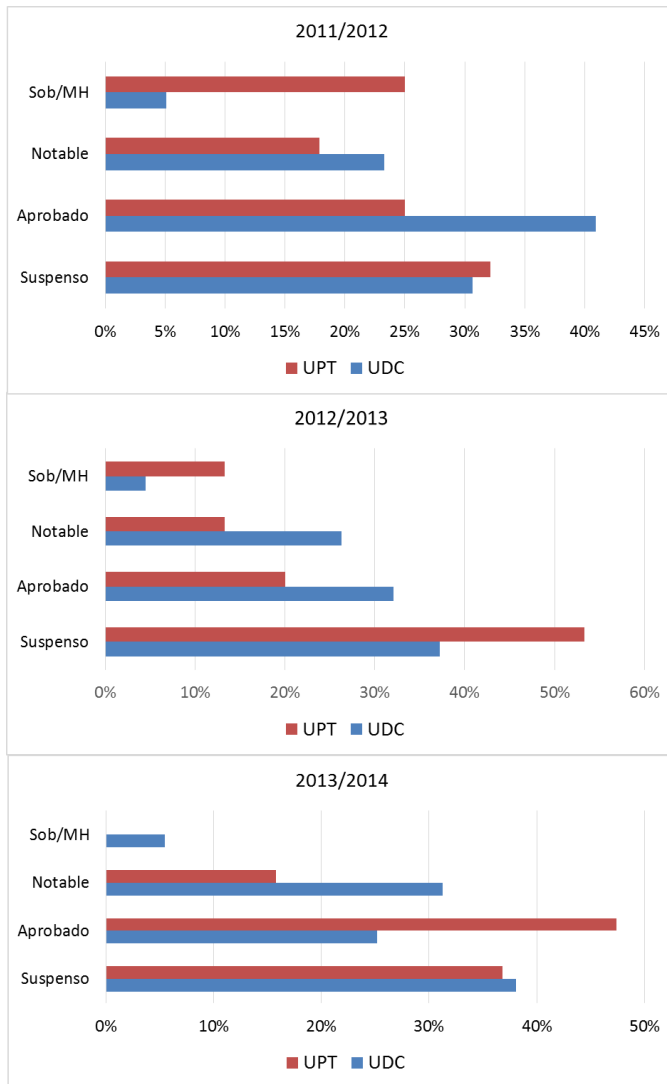


Fig. 1. Resultados académicos desglosados por calificación para los tres cursos académicos sobre el total de presentados

After analyzing this result it is found that the value of UPT is punctual and that the trend is shown for the other two courses. Therefore, the fundamental difference between the two institutions is in the percentage of students attending the exams reaching the highest rating, this being much higher in the case of UPT. For all the other categories, although with some specific differences, the percentages follow the same line. Here comes in the evaluation method applied in UDC where the weight of the written exam in the final grade is high. Studying in detail the results it can be observed as the UDC students who get a good grade in practices and troubleshooting problems do not reach the highest possible grades in the examination and that makes them stay in the remarkable and not reach the highest marks. This leads us to reflect and rethink the weights of the various activities in the assessment method: it seems reasonable to reduce the weight of the theoretical test and increase the weight of other activities that objectively evaluate the ongoing work of students.

B. The problem of non-presented students

The second thing to note is the number of students presented for evaluation. Fig. 2 shows the percentages of students submitted and not submitted for evaluation considering the total enrolled since the academic year 2011/2102 to 2013/2014. In the graphs it can be seen as in the year 2011/2012 in UDC a 39.31% of the students enrolled were not submitted to examination; in 2012/2013 this percentage is even higher reaching 54.65% being this value exceeded in the 2013/2014 course assuming 62.97% of the total. We are therefore faced with a crucial problem that we need to address and try to tackle as far as possible. To this end, we have tried to delve into the reasons for the abandonment, understood as the renunciation of evaluation submission. Many of the waivers occur even before starting the quarter, since there are students who no longer are incorporated as such and in principle are out of any evaluation process. Other students stop attending class, either because they prefer to concentrate their efforts on other more affordable subjects according to their level of competence or their training is not enough to successfully meet the course.

Few studies shed light on such student types [19], [20] however, reports that clarify this problem attribute, surprisingly and in light of the students opinion, mainly to external causes such as organizing schedules subjects and, to a lesser extent to the teaching methodology [21].

The study on the possible causes of abandonment reveals that one of the main reasons is the inability to achieve the competencies and skills that students must acquire in ProII, to solve the proposed practical cases. We considered that the possibility that the problem also lies in not having achieved the competencies and skills required and which are supposed to be previously acquired.

Teachers believe that the situation of ProII in the curriculum (second half of the first year) could not be correct because at this point the students have poor base from the point of view of programming. The heavy reliance on the subject of ProI is a major handicap.

For this reason we are now interested in reflecting on the results achieved in UPT. In this case, due to the structuring of the curriculum, students have a first full academic year where they receive training with regard to matters relating to programming. Specifically Algorithms and Programming (first semester of first year) and Object-Oriented Programming (second semester of first year). In our opinion, this situation allows students to get better programming skills. Thus, we compared the results based on the percentage of students not presented in ProII and DS at both universities.

Fig. 3 shows the percentages of student's not present, succeeding and failing at both institutions for the academic years 2011/12 to 2013/14.

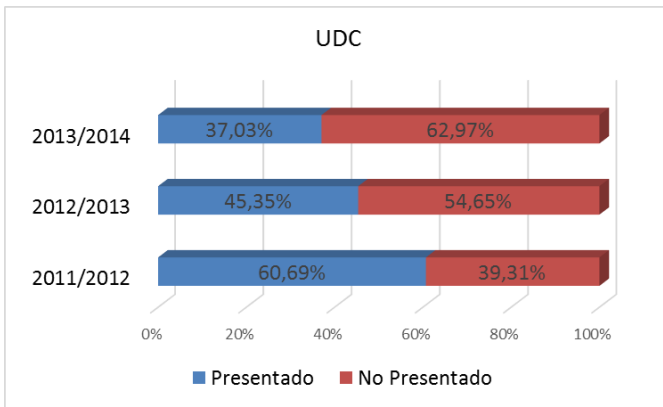


Fig. 2. Porcentaje de alumnos presentados y no presentados a examen en la UDC durante los cursos 2011/2012, 2012/2013 y 2013/2014

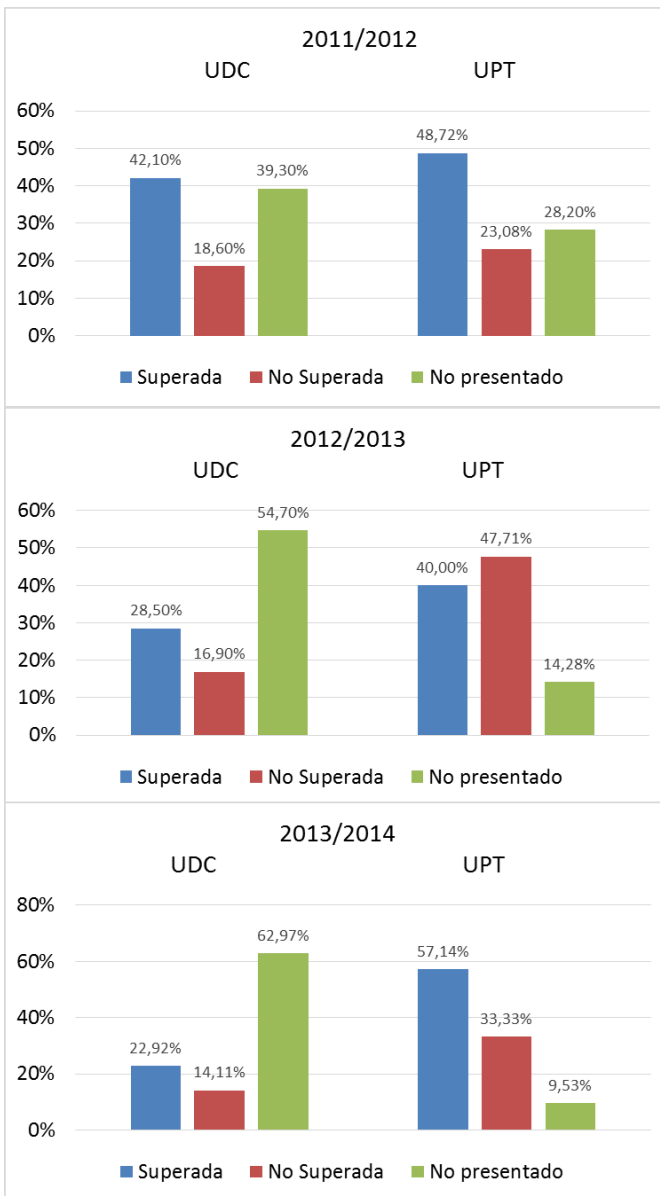


Fig. 3. Resultados académicos desglosados por calificación para los tres cursos académicos sobre el total de presentados

As it can be seen the rate of non-presented students is, in general, lower in UPT. A detail that should be noticed is the increasing trend of non-presented students in UDC over the three academic years (39.30% in 2011/2012, 54.70% in 2012 / 2013 and 62.97% in 2013/2014). However, the trend is reversed in the case of UPT where the percentage is 28.20% in 2011/2013 decreasing, progressively, reaching 9.53% in 2013/2014. The main difference between the two institutions is that in UPT the delivery of the main subjects of interest is performed in the second year after the study of subjects related to programming throughout the first academic year. In fact, despite the UPT teaches the subjects in the second year, the teachers involved indicate that most of the students who have not studied the subjects considered as a prerequisite leave, not being able to follow the content. Teaching the contents of interest in the second course allows students to have more relevant basic knowledge, unlike what happens in UDC where strong dependence from ProI in the first course translates into a large number of students without required knowledge and not enough time to assimilate it. This is a structural problem, for which there is no short-term solution, since it affects the curriculum established in the university. However, aware of these limitations, the teacher poses some alternatives to alleviate, as far as possible, its consequences. Well worth mentioning, the implementation of a plan of student motivation since the beginning of the semester, discussing what are the myths about overcoming matter, adding reinforcement classes to strengthen basic concepts, and reinforcing personalized tutoring work allowing small group classes. As an attempt to avoid losing the students in the areas of programming, it is drawn jointly with the ProI (first half) teaching staff a continuing education program that allows students to address, in the best way possible, content associated with programming matters. The aim is to avoid, at least in part, absenteeism or abandonment by students in the field.

V. CONCLUSIONS

Reflecting on the results achieved by students should be standard practice, not only to detect strengths and weaknesses in our educational project but to seek solutions to correct deficiencies. In order not only to improve success rates but to better educate our students, we have made a small work of introspection in the teaching of the subject of ProII that has led us to research how the teaching of such contents in other institutions is. This fact has led to a continued collaboration with the UPT teaching staff through teacher mobility programs.

As for the main problems we face in the delivery of content associated with ProII and after a comparison with the approach used in UPT we can conclude that:

(1) As for the success rate, the main differences are located in the percentage of students reaching the highest rating, with much lower values in the case of UDC. The main reason is the difference in weight assigned to each of the items that are part of the evaluation process. In the case of UDC the weighting associated with the theoretical single written test is high, 75%, and this makes that students with the highest rating in the

practical and problem solving items (20% and 5% respectively of qualifying total), not reach the highest rating. Consequently, the vast majority of the scores achieved by these students are concentrated between the approved and remarkable.

(2) With regard to the problem of non-presented students and based on the results shown in Section IV.B. it can be concluded that teaching this matters in the second half of the first course has a number of drawbacks that are apparent when students face the contents of the subject: lack of basic knowledge or limited time available to assimilate knowledge. The teaching methodology, based on active student learning, requires students to perform a continuous work involved in delivering various works and laboratory practice and sometimes fail to cope with the tasks conferred. ProI strong dependence explains the high dropout rates of the subject. The curriculum in UPT, where the contents are offered in second year after a first course with matters related to programming promotes student learning and this is reflected in the results where the rate of non-presented students in UPT differs significantly with respect to UDC, being much less. Therefore, we find our idea of the need for a complete pre-programming course for students to entrench their knowledge base and thus can adequately deal with the topics that concern us. In this way, it could be avoided, at least in part, absenteeism or abandonment by students in the field.

REFERENCES

- [1] A. Gomes, C. Areias, J. Henriques y A. J. Mendes, "Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte", *Revista Portuguesa de Pedagogia*, vol. 42, no. 2, pp. 161-179, 2008.
- [2] A. Gomes, "Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução" Tese de Doutoramento, Faculdade de Ciências e Tecnologia, Departamento de Engenharia Informática, Universidade de Coimbra, 2010.
- [3] W. F. Archison, S. D. Conte, J. W. Hamblen, T. E. Hull, T. A. Keenan, W. B. Kehl, E. J. McCluskey, S. O. Navarro, W. C. Rheinboldt, E. J. Scweppe, W. Viavant y D. M. Young, "Curriculum 68: Recommendations for Academic Programs in Computer Science: A Report of the ACM Curriculum Committee on Computer Science", *Commun. ACM*, 11(3), 151-197, 1968.
- [4] R. H. Austing, B. H. Barnes, D. T. Bonnette, G. L. Engel y G. Stokes, "Curriculum 78: Recommendations for the Undergraduate Program in Computer Science. A Report of the ACM Curriculum Committee on Computer Science", *Commun. ACM* 22 (3), 147-166, 1979.
- [5] R. Celso, M. Correia, R. E. Garcia, C. O. Junior, A.C. Brandi y G. P. Cardim, "A Methodological Approach to Use Technological Support on Teaching and Learning Data Structures", in *Proc. Frontiers in Education Conference (FIE)*, pp. 1-8, 2014.
- [6] T. D. Hendrix, J. H. Cross y L. A. Barowski, "An extensible framework for providing dynamic data structure visualizations in a lightweight IDE", *SIGCSE Bulletin*, ACM, vol. 36, no. 1, pp. 987-391, 2004.
- [7] A. S. S. Nair y T. Mahalakshmi, "Conceptualizing Data Structure: a Pedagogic Approach", *SIGCSE Bulletin*, ACM, vol. 36, no. 4, pp. 97-100, 2004.
- [8] R. Lister, I. Box, B. Morrison, J. Tenenberg y D. S. Westbrook, "The dimensions of variation in the teaching of data structures", *SIGCSE Bulletin*, ACM, vol. 36, no. 3, pp. 92-96, 2004.
- [9] J. Tenenberg, "A framework approach to teaching data structures", *SIGCSE Bulletin*, ACM, vol. 35, no. 1, pp. 210-214, 2004.
- [10] A. T. Duchowski, "τέχνη photons: Evolution of a course in data structures", *Computer Graphics Forum*, vol. 33, no. 1, pp. 294-304, 2014.
- [11] X. Wang, J. Fan, G. Sun y S. Nan, "The Exploration and Practice of Data Structure Teaching Based on CDIO Model", in *Proc. 7th International Conference on Computer Science & Education (ICCSE)*, pp. 1403-1407, 2012.
- [12] M. J. Guzdial y B. Ericson, *Problem Solving with Data Structures Using Java: A Multimedia Approach*, Prentice Hall Press, 2010.
- [13] M. A. Weiss, "Experiences teaching data structures with Java", *SIGCSE Bulletin*, ACM, vol. 29, no. 1, pp. 164-168, 1997.
- [14] F. Carrano y T. Henry, *Data Abstraction & Problem Solving with C++: Walls and Mirrors*, Pearson, 2016.
- [15] F. Carrano y J. Prichard, *Data Abstraction & Problem Solving with Java: Walls and Mirrors*, Pearson, 2015.
- [16] M. A. Weiss, *Data Structures and Problem Solving Using Java*, Pearson, 2013.
- [17] M. A. Weiss, *Data Structures and Problem Solving Using C++*, Addison-Wesley, 2014.
- [18] J. Crosby, "Learning in small groups", *Medical Teacher*, vol. 18, no. 3, pp. 189-202, 1996.
- [19] A. Cernuda, S. Hevia, M. C. Suárez y D. Gayo, "Un estudio sobre el absentismo y el abandon en asignaturas de programación", en *Actas de las XIII Jornadas de Enseñanza Universitaria en Informática (JENUI)*, pp. 487-494, 2007.
- [20] R. Rodríguez, J. Hernández, A. Alonso y E. Díez-Itza, "El absentismo en la universidad: resultados de una encuesta sobre motivos que señalan los estudiantes para no ir a clase", *Aula Abierta*, vol. 82, pp. 117-146, 2003.
- [21] I. Sanz, M. Aramburu, L. Museros, M. Pérez y C. Barrachina, "En busca del estudiante perdido: caracterización de los no presentados", en *Actas de las XVII Jornadas de la Enseñanza Universitaria de la Informática (JENUI)*, pp. 403-410, 2011.



Beatriz Pérez-Sánchez Ingeniera y Doctora en Informática por la Universidade da Coruña desde los años 2005 y 2010, respectivamente. Desde el año 2007, ha desempeñado diferentes puestos investigadores y docentes en el Área de Ciencias de la Computación e Inteligencia Artificial de la Universidade da Coruña. Con respecto a su actividad investigadora, ha publicado diversos trabajos como artículos en revistas científicas, contribuciones a congresos científicos y capítulos de libros, siendo la mayor parte de las aportaciones de ámbito internacional. Asimismo, ha participado en diferentes proyectos financiados en convocatorias públicas de carácter competitivo y en contratos de especial relevancia con empresas privadas. A lo largo de estos años ha ganado varias becas de carácter competitivo para la realización de diferentes actividades de formación, entre las que cabe destacar las ayudas para movilidad tanto con fines investigadores como docentes. A mayores, ha recibido premios de carácter competitivo como son el Premio José Cuena de la Asociación Española para la Inteligencia Artificial al mejor artículo Doctoral Consortium en la XII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2007) o el Premio Extraordinario de Doctorado en Ingeniería Informática por la Universidade da Coruña por su tesis doctoral defendida en el año 2010.



Paula Morais profesora asociada del Departamento de Economia, Gestão e Informática da Universidade Portucalense, Porto, Portugal. Doctorada en Tecnologías e Sistemas de Informação, por la Universidade do Minho en 2002. Su labor docente está ligada a las disciplinas de Ingeniería de Software y Gestión de Conocimiento. Su investigación se enmarca en las áreas de sistemas de información en las organizaciones y gestión de conocimiento organizacional. Participa como orientadora en proyectos de fin de curso de las licenciaturas en Informática e Tecnologías e Sistemas de Informação y en tesis de master y doctorado en el área de Informática. En los últimos años ha participado en los proyectos "Erasmus +, KA2, 3C - Creating Competences for a Circular Economy" y, "Necesidades específicas de RRHH para el sector de las TIC". Además, ha sido coordinadora de las estancias pedagógicas de la Licenciatura en Informática y evaluadora de proyectos en el área de SI/TI y también de las candidaturas del Programa ERASMUS+.