

ABET
Self-Study Report
for the
Computer Science Program
at
Pontificia Universidad Javeriana
Cali, Colombia

27th of May 2015

CONFIDENTIAL

The information supplied in this Self-Study Report is for the confidential use of ABET and its authorized agents, and will not be disclosed without authorization of the institution concerned, except for summary data not identifiable to a specific institution.

Contents

A	Contact Information	11
B	Program History	11
C	Options	13
D	Program Delivery Modes	13
E	Program Locations	13
F	Public Disclosure	13
G	Deficiencies, Weaknesses or Concerns from Previous Evaluation(s) and the Actions Taken to Address Them	14
1	Students	15
1.A	Student Admissions	15
1.B	Evaluating Student Performance	15
1.C	Transfer Students and Transfer Courses	16
1.D	Advising and Career Guidance	17
1.E	Work in Lieu of Courses	17
1.F	Graduation Requirements	17
1.G	Transcripts of Recent Graduates	18
2	Program Educational Objectives	19
2.A	Mission Statement	19
2.B	Program Educational Objectives	20
2.C	Consistency of the Program Educational Objectives with the Mission of the Institution	21
2.D	Program Constituencies	21
2.E	Process for Review of the Program Educational Objectives	23
3	Student Outcomes	25
3.A	Student Outcomes	25
3.B	Relationship of Student Outcomes to Program Educational Objectives	29
3.C	Process for the Establishment and Revision of the Student Outcomes	30
3.D	Enabled Student Characteristics	30

4	Continuous Improvement	31
4.A	Student Outcomes	31
4.A.1	Program global assessment model	31
4.A.2	Planning of the program global assessment	31
4.A.3	Course assessment model	32
4.A.4	Assessment at the program level	39
4.A.5	Attainment of the student outcomes	40
4.A.6	Results in the national examinations	46
4.A.7	Overall conclusions about the global process assessment	48
4.B	Continuous improvement	49
4.B.1	Continuous improvement in the evaluation courses	49
4.B.2	Improvement plan for the results in the national examinations	52
4.B.3	Overall improvement of the Program	52
4.C	Additional Information	53
5	Curriculum	55
5.A	Program Curriculum	55
5.A.1	Plan of study of the Program	55
5.A.2	Alignment with the program educational objectives	55
5.A.3	Attainment of student outcomes	57
5.A.4	Flowchart of the courses and prerequisite structure	58
5.A.5	Meeting of general and program criteria	59
5.A.6	Cooperative education	59
5.A.7	Material available for review	60
5.B	Course Syllabi	61
6	Faculty	69
6.A	Faculty Qualifications	69
6.B	Faculty Workload	70
6.C	Faculty Size	71
6.D	Professional Development	72
6.E	Authority and Responsibility of Faculty	73
7	Facilities	81
7.A	Offices, Classrooms and Laboratories	81
7.A.1	Offices	81
7.A.2	Classrooms	81
7.A.3	Laboratories and computer rooms	82
7.B	Computing Resources	82
7.B.1	Hardware platform	82
7.B.2	Software platform	83
7.B.3	Services	84
7.C	Guidance	85

7.D	Maintenance and Upgrading of Facilities	85
7.E	Library Services	87
7.E.1	Services	88
7.F	Overall Comments on Facilities	89
8	Institutional Support	91
8.A	Leadership	91
8.B	Program Budget and Financial Support	92
8.B.1	Process to establish the programs budget	92
8.B.2	Support to teaching	93
8.B.3	Support to infrastructure, facilities and equipment	94
8.B.4	Adequacy of resources	95
8.C	Staffing	95
8.D	Faculty Hiring and Retention	96
8.D.1	Faculty hiring	96
8.D.2	Retention of faculty	99
8.E	Support of Faculty Professional Development	100
	References	105
	A Course Syllabi	107
	B Faculty Vitae	191
	C Equipment	213
C.1	Computer Science Laboratory	213
C.2	Networks Laboratory	214
C.3	Interaction and Sound Laboratory	214
D	Institutional Summary	217
D.1	The Institution	217
D.2	Type of Control	217
D.3	Educational Unit	218
D.4	Academic Support Units	219
D.5	Non-academic Support Units	220
D.6	Credit Unit	222
D.7	Tables	222
D.7.1	Program Enrollment and Degree Data	222
D.7.2	Personnel	222
E	The score system	225

F	Course evaluation	227
F.1	Course registry	227
F.2	Performance indicators	228
F.3	Numerical model for the evaluation of courses	230
F.4	Example of an evaluation balance:	232
F.5	Evaluation instruments used in the courses selected for the program assessment	233
F.6	Course report	234
F.7	Results of student outcomes in courses	235
F.8	Results in the national examinations	237
G	Bibliographic data bases	239

List of Tables

3.1	Relation between the PEO and SO. (0) means no relation and (5) means a strong relation	29
4.1	Relationship of courses with student outcomes. Relative weight in scale 0-5.	32
4.2	Comparison of relative weights of student outcomes between Table 3.1 and Table 4.1	35
4.3	Courses selected for the program assessment	36
4.4	Overall results of evaluation courses	42
4.5	Statistical characterization of the individual results on student outcomes of the evaluated courses	43
4.6	Average results on student outcomes of the Undergraduate Project	45
4.7	Overall results of the evaluation of the Professional Internship in the 2014-2	46
5.1	Curriculum of Computer Science Program	62
5.2	Curriculum of Computer Science Program - Concentration I	64
5.3	Curriculum of Computer Science Program - Concentration II	64
5.4	Subject areas and courses of the Computer Science Curriculum	65
5.5	Relationship of courses with program educational objectives.	66
6.1	Faculty qualifications - Computer Science Program	75
6.2	Faculty Workload Summary - Computer Science Program	77
6.3	Allocation of faculty members to courses in the Computer Science Area . . .	79
8.1	Budget of the Computer Science program 2010 - 2014 (in million Colombian pesos. 1 USD = \$1900)	93
8.2	Conditions for admission into the ranking of full-time faculty	97
8.3	Conditions for promotion through the ranking of full-time faculty	100
D.1	Program Enrollment and Degree Data	222
D.2	Personnel	223
F.1	Performance indicators associated with courses	228
F.2	Evaluation balance of the Formal Software Development course	232
F.3	Distribution of evaluation in an exam of the Formal Software Development course.	232

F.4	Evaluation instruments used in the courses selected for evaluation (H: Homework and workshops; E: Exams; N: News reports; P: Course project; Q: Quizzes; R: Reading report; S: Socialization; T: Presentations (talks)) . . .	233
F.5	Average results on student outcomes of the evaluated courses	235
F.6	Annual averages of results in the Saber Pro exams of 2012 and 2013	237
G.1	Data bases available for the Computer Science program	240

List of Figures

2.1	Relation between the Program and Institution Mission and the PEOs	22
4.1	Example of curriculum integration: Formal Software Development course. . .	34
4.2	Summary of the evaluation of students outcomes in courses	39
4.3	Evolution of the overall grades on student outcomes of the evaluated courses.	43
4.4	Frequency distribution of individual grades on student outcomes of the evaluated courses.	44
4.5	Semester average results in the generic competencies of the Saber Pro exams. ◦ citizenship; Δ written communication ; ∇ English; \bullet critical reading; \blacktriangle quantitative reasoning	47
4.6	Quintile distribution of results in the Saber Pro exams. Q1: first quintile; Q2: second quintile; Q3: third quintile; Q4: fourth quintile; Q5: fifth quintile . .	48
5.1	Prerequisite structure for Computer Science program curriculum	68
D.1	Organization chart of the School of Engineering	218
F.1	Template of the course registry	227
F.2	Template of the course report	234

Background Information

A Contact Information

The primary contact person is

Dr. Antal Buss
Pontificia Universidad Javeriana - Cali
Facultad de Ingeniería
Calle 18 No. 118-250, Av. Cañasgordas
Cali, Colombia.
Tel: +57 (2) 321 82 00 ext 8804
Fax: +57 (2) 555 28 23
e-mail: abuss@javerianacali.edu.co

You may also contact the general coordinator of the accreditation process:

Prof. Jorge Francisco Estela
Pontificia Universidad Javeriana - Cali
Facultad de Ingeniería
Calle 18 No. 118-250, Av. Cañasgordas
Cali, Colombia.
Tel : +57 (2) 3218200 ext 8876
Fax: +57 (2) 555 28 23
e-mail: jfe@javerianacali.edu.co

B Program History

The **Computer Science Program** program was created in 1984. It was the second academic program of the School of Engineering at Javeriana University. The name of the program was firstly “*Ingeniería de Sistemas con Énfasis en Procesos Industriales*” (System Engineering with Emphasis in Industrial Processes) and the curriculum was mainly focused on the development of (software) tools for automatizing business processes. In 1989 the program changed its name to “*Ingeniería de Sistemas y Computación*” (Computer Science). In Colombia, the term “Ingeniería de Sistemas” (System Engineering) is a traditional name for all the programs related to *computing*. However, the focus of the program is Computer

Science, and this is how it will present internationally. Since that time, the teaching, research and extension activities of the program rely on a broader sense of computing as a science and as an engineering with a special emphasis in the ability of *modeling*. We are proud of being recognized as one of the most important schools of computer science in Colombia and also for the analytical and formal reasoning skills of our alumni. The research activities of the *Electronic and Computer Science Department* (DECC) offer a fertile place to regularly host international researchers. They develop joint projects with the faculty members and also offer short courses and seminars to our students.

The current curriculum of the program was designed in 2005 based on the 14 areas of ACM Computing Curricula [5] and it comprises four components:

- Core curriculum: 128 credit hours, 75.3 percent of the total.
- Concentrations: 18 credit hours, 10.6 percent of the total.
- Minor: 12 credit hours, 7.1 percent of the total.
- General electives: 12 credit hours, 7.1 percent of the total.

Thus, the total number of credit hours is 170. A credit hour is equivalent to 48 hours of student work in a 16-week semester. Core courses correspond to the main topics of the computing discipline (including, e.g., programming languages, algorithms and complexity, discrete structures, information systems, networking, etc). In addition to this, the program includes courses in calculus, physics and humanities.

The program offers two *concentrations* of 18 credits each, starting at the 6th semester. It is mandatory for students to take at least one of them. Currently we offer:

- **Animation and Interactive Systems**, including topics related to video game design, sound processing and artificial intelligence for video games.
- **Net-Centric Computing**, including topics on multimedia, security and mobile systems.

In December 2004, the *Consejo Nacional de Acreditación* (CNA, National Accreditation Council) conferred the National Accreditation to our program for a period of 7 years. After the evaluation process in 2010 and 2011, the CNA renewed the accreditation of the program in 2012 for 6 years more.

In 2010, the Engineering School and the program started the ABET's assessment process. In 2012, the External Committee was established and the program is currently revising the outcomes of the evaluation process.

The most prominent changes in the program resulting from the evaluation processes and the accreditation are:

- Research activities of faculty members became better known to students in courses of the above mentioned concentrations.

- A new course on *system modeling* was added in the first semester.
- The networking course was augmented from 2 credits to 3 credits.
- Every course syllabus describes with precision the outcomes expected and the means to achieve them.
- We fostered the student's international experiences by means of agreements such as the design course ME310 (joint course with Stanford University) and the double diploma agreement with Turin University in Italy.

C Options

Students must take one of the above mentioned concentrations, at least one *opción complementaria* (minor) and 12 credits in elective courses. Regardless the choices taken, students obtain the same degree, Bachelor in Computer Science. The University promotes double degrees. Thus, some students pursue that option by taking all the credit hours in the minor and in the general electives in, typically, core-courses from a second program.

The core courses related directly to the discipline are offered by the *Electronic and Computer Science Department* (DECC). All the faculty of the program belongs to this department. The members of the DECC are also in charge of the *Master Program* in engineering with tracks in Computer Science and Electronic Engineering.

D Program Delivery Modes

This is an on-campus and day-time program. Courses, laboratories and other curricular activities are normally scheduled Monday to Friday. Academic activities are delivered on a traditional lecture-and-laboratory methodology. The Program operates on the university defined annual schedule of two 18-week semesters: one from late January to early June (Spring semester) and the other from late July to early December (Fall semester); depending on demand, courses are sometimes offered in the two inter-semester periods (Summer and Winter semesters).

E Program Locations

The program's location is on campus of the Javeriana University.

F Public Disclosure

As of September 2014 the Program is taking steps to ensure the public disclosure of the program educational objectives, student outcomes, and enrolment and graduation data. By the

end of October 2014 this information will be available at <http://wikiabet.javerianacali.edu.co/doku.php?id=pregrados:inicio>, and it will be available through the webpage soon thereafter.

G Deficiencies, Weaknesses or Concerns from Previous Evaluation(s) and the Actions Taken to Address Them

This is an initial ABET accreditation submission. The most significant weaknesses found in the national accreditation processes and the actions taken to address them were:

- *Report systematically the outcomes and abilities the students must attain.* The syllabi related to the discipline are now available on the DECC's wiki where students and faculty members can find the outcomes expected from each course. The wiki also reports the activities that must be followed to attain the desired outcomes.
- *Design better promotion strategies for the program.* In order to face the low number of students in the first semester, the program has designed activities related to the practice of computing aimed at high school students. Activities such as *Javatar* has attracted the attention of many of those students.
- *Provide spaces where the students may have contact with different tools and frameworks.* The program is planning to offer elective courses in different tools and frameworks to address this difficulty.

Criterion 1

Students

1.A Student Admissions

The admission of new students is defined in Title II of the Student Regulations (*Reglamento de Estudiantes*) [6] and is a two-step process: signing-up, and admission proper.

The signing-up [6, numbers 8 and 9 of the Student Rules] is the formal submission of the request for admission to the program. The most important requirements for signing-up are: filling in the corresponding format; having presented the national examinations (called the Saber 11 exams); and having graduated from secondary school, or being studying the final year of secondary school.

The admission proper [numbers 13 to 18 of the Student Regulations] is the process whereby the University accepts or rejects the request for admission of the student. In summary, the admission is decided by the Dean of the School in consultation with the Program Director, who, to that effect, constitute the Admissions Committee. Two sources of information are used to select the candidates for admission:

- The results in the Saber 11 exams and the candidate must have scores of minimum 50 in each of the areas of Mathematics, Physics, Spanish and English.
- The results in an interview, in which the candidate is examined with regards to knowledge and interest about the program, pursuit of scholarly excellence, analytical thought, communication and relational abilities, and concern about social issues.

The scores in these two sources are combined with a relative weight of 40 percent for the areas of interest of the Saber 11 exams and 60 percent for the results of the interview. Candidates for admission must have a minimum score of 65 in that combined scale.

1.B Evaluating Student Performance

Academic evaluations are defined in Title IV of the Student Regulations[6] and the following is a summary of those regulations.

At the beginning of each course, students must be informed as to how the course will be graded, i.e. the instruments of evaluations that will be used and their percentage over the global grade of the course. This information is included in the course syllabi and is also uploaded on the information system of the Academic Registrar so that the system automatically calculates course grades from grades on evaluation instruments.

To evaluate student performance in the Computer Science Program a variety of evaluation instruments are used, most commonly: exams, short exams, homework, laboratory reports, oral presentations, and course projects. Student performance is expressed in grades on the scale 0.0 to 5.0, with the pass at 3.0, i.e. 60 percent performance. The overall performance of students is calculated by the weighted average of grades in courses by the number of credit hours of those courses; two such weighted averages are calculated at the end of every semester: a semester only weighted average, and a global (all semesters) weighted average. Based upon these two averages, Program Directors monitor the progress of students through their plan of study.

According to the academic calendar of the University, exams are scheduled to take place at about one third, at two thirds and at the end of the semester. By the third-fourth week of the semester, the Secretary of the School communicates to all courses the scheduling of exams. The scheduling of the other evaluations is done by the course coordinators.

Course coordinators are required to return to students graded evaluations not later than a week after the evaluation took place, and upload those grades in the information system of the Academic Registrar.

If a student failed to take a scheduled evaluation, a supplementary evaluation is authorized by the Program Director provided a due cause. The request for a supplementary evaluation, with the supporting proof of due cause, must be submitted within five working days of the scheduled evaluation.

To enroll in courses, students must have approved all the prerequisite courses. As the prerequisite structure of the curriculum is programmed into the Registrars system, and the system records all the previously reported course grades, enrollment in courses is automatically prevented if the prerequisites are not met. Only the Program Director can exceptionally authorize enrolling in courses without prerequisites. A number of criteria Program Directors can apply to reach an informed decision whether to waive or not a prerequisite: if the prerequisite course has not been taken, or if it was taken but with a grade far below the pass, the prerequisite will not be waived; on the contrary, if the prerequisite grade was close to the pass, and if the overall performance of the student is satisfactory, the prerequisite might be waived.

1.C Transfer Students and Transfer Courses

Numbers 23 to 32 of the Student Regulations describe internal and external transfers, such that:

- Internal transfer: this is when a student changes from one academic program to another within the University. To be accepted for such procedure, the student must not have

suspended studies longer than two semesters. The Dean of the School, in consultation with the Program Director, decides upon the requests for internal transfer. If it is authorized, approved courses can be homologated into the new curriculum.

- External transfer: this is when a student comes from another institution. Three conditions have to be met: the student must take, at least, 25 percent of the credit hours in the new program; the student must not have suspended studies for more than two semesters; and the student must not have been expelled from the previous program. The Dean of the School, in consultation with the Program Director, decides upon the requests for external transfer. If authorized, approved courses can be homologated into the new curriculum.

1.D Advising and Career Guidance

The Academic Units Regulations[7] define the academic counseling as a service that academic programs offer to their students, with the purpose of contributing to the achievement of the educational objectives of the program.

Program Directors are directly responsible for academic counseling in their programs, but a number of faculty members are appointed as academic counselors to assist Program Directors in that responsibility. In the Computer Science program, the Program Director is assisted by one faculty member from the Computer Science area of the Electronics and Computer Science Department. Academic counselors help students organize their plan of study, including the choice of concentrations and minors, within a perspective of flexibility.

Students are assigned to academic counselors at the beginning of their first semester. In the Computer Science Program, attendance to academic counseling is compulsory in the following situations: being in first semester; the first semester of residence in the Program for students admitted by internal or external transfer; and, very importantly, being in academic probation. Students in any of these conditions attend counseling twice in the semester. For the rest of students, academic counseling is not compulsory.

1.E Work in Lieu of Courses

The Student Regulations do not have any provision for work in lieu of courses; therefore, such mechanism is not applicable in the Computer Science Program.

1.F Graduation Requirements

The degree awarded by the Program is Bachelor of Science in Computer Science.

According to the Student Regulations (number 104) the graduation requirements are:

- To have approved all the courses of the plan of study and met all the requirements of the programs curriculum.

- To have approved at the University at least 25 percent of the credit hours of the program.
- To be in good standing regarding academic and administrative procedures and produce all the legal and university documentation.

1.G Transcripts of Recent Graduates

Transcripts of recent graduates will be provided upon request by the evaluation team. Those transcripts will be accompanied with an explanation as to how students met the graduation requirements and how they organized their plans of study.

Criterion 2

Program Educational Objectives

2.A Mission Statement

Institutional Mission Statement

According to the University statutes, the mission statement for the Javeriana University is the same for its two branches in Bogotá and Cali (see Agreement No 576, 26/04/2013. [4]):

Pontificia Universidad Javeriana is a catholic institution of higher education, founded and run by the Society of Jesus and committed to the guiding principles of its founders. It performs with excellence teaching, research and service as a university integrated to a country of regions but with global and interdisciplinary perspectives; and seeks:

- *The integral education of individuals that outstand because of their high human, ethical, academic and professional qualifications and for their social responsibility,*
- *The creation and development of knowledge and culture within critical and innovative perspectives,*

to achieve a society that is characterized by justice, sustainability, inclusiveness, democracy, solidarity and respect for human dignity.

School of Engineering Mission Statement

Our mission, as part of Javeriana Cali, is to contribute, from the fields of science, engineering and technology, to the betterment of society by the pursuit of innovative solutions to problems of today and tomorrow.

To that end, we foster an environment of intellectual enquiry and research, scholarly excellence and social responsibility in which our students, faculty, and staff develop their potential to the fullest.

As a result, we educate our students to serve society and excel as leaders, competent professionals, and responsible citizens and do research and service of high quality and relevance.

Computer Science Program Mission Statement

The Computer Science program aims at educating engineers in the solid foundations of the discipline for them to contribute to society by building computational models and implementing computing technologies to address problems in science, engineering and industry. To that effect, we foster an environment of scholarly excellence that allows our students to know, value, and use effectively the techniques, formalisms and methods of computer science. As a result we educate our students to be professionally competent, to have a distinct sense of ethics and to be committed to the life-long quest for knowledge and the improvement of the quality of the software industry in Colombia.

2.B Program Educational Objectives

The current Program Educational Objectives (PEOs) are stated in the Program Educational Project [11] (*Proyecto Educativo del Programa*) and they were presented to the stakeholders in different meetings (see e.g., the minutes [9, 10]). The document [11] is distributed to the first semester students and it is also available on the program web page.

The graduates from the Computer Science program shall:

- Be able to solve problems in science and engineering by effectively applying computer science knowledge and techniques.
- Use their abilities, ingenuity and analytical skills and formal reasoning in computer science to identify, analyze, design, implement, maintain and operate technological solutions to solve discipline-related problems or contribute to the solution of problems in other fields.
- Be performing with excellence in promising positions in industry, academia, public service, or as entrepreneurs, and thereby to contribute to the welfare of their communities.
- Be distinguished by their sense of ethics and responsible citizenship, well-educated professionalism, awareness of the need for sustainability, strong formation in computing and teamwork and communication skills.
- Be committed to the life-long quest of knowledge by pursuing graduate studies, self-study or professional development.

In November 2014 a preliminary self-study report was submitted to the Readiness Review. The fellow engineering programs of the School of Engineering that were reviewed by the Engineering Accreditation Commission received the recommendation to revise the drafting of their program educational objectives for them not to look as student outcomes. Therefore, the Program Committee acknowledged the appropriateness of such recommendation and revised the above statements but being especially careful to preserve the spirit and aspirations declared in the original texts. Thus, the following are the Computer Science

program educational objectives (PEOs).

The graduates of the Computer Science program shall:

- EO.1 Exercise the professional practice of Computer Science.
- EO.2 Design and operate computing systems that contribute to the solution of problems related to the discipline, to other fields of science and engineering, or to other disciplines.
- EO.3 Contribute to the welfare of their communities from promising positions in industry, academia or public service, or as entrepreneurs.
- EO.4 Be distinguished by their sound foundations in computing, sense of responsible citizenship, professionalism and leadership.
- EO.5 Continue their professional development or engage in graduate studies.

2.C Consistency of the Program Educational Objectives with the Mission of the Institution

In order to establish the consistency, we decompose the mission of the institution and the mission of the program as shown in Figure 2.1. Then, we relate those statements with the PEOs.

2.D Program Constituencies

The program constituencies are listed below. In each case, we summarize how the PEOs meet their needs:

- **Students** are the natural constituencies of the program and they were consulted before giving the final definition of the PEOs in [11]. The student representative integrates the Program Committee and his/her opinion is taken into account in the different decisions the committee takes. Before graduation, students answer a survey where they are asked whether the program and its PEOs met their needs.
- **Faculty Members** participate in the different assessment processes of the program. In the Department and Program Committee, the general orientation of the Computer Science Program are discussed.
- **The university** provides the support and resources needed for the development of the program.

Institution's Mission	School of Eng. Mission	Program's Mission
<ol style="list-style-type: none"> 1. Pontificia Universidad Javeriana is a catholic institution of higher education, founded [...] 2. It performs with excellence teaching, research and service [...] 3. [...] but with global and interdisciplinary perspectives [...] 4. The integral education of individuals that outstand [...] 5. The creation and development of knowledge and culture [...] 6. [...] to achieve a society that is characterized by justice, sustainability, [...] 	<ol style="list-style-type: none"> 1. Our mission, as part of Javeriana-Cali [...] 2. [...] to the betterment of society [...] 3a [...] we foster an environment of intellectual enquiry and research 3b [...] we foster an environment of scholarly excellence [...] 3c [...] we foster an environment of social responsibility [...] 4 [...] an environment [...] in which our students, faculty and staff develop their potential to the fullest. 5a [...] we educate our students to serve society [...] 5b [...] we educate our students to excel as leaders [...] 5c [...] we educate our students to excel as competent professionals [...] 5d [...] we educate our students to excel as responsible citizens [...] 6 [...] we do research and service of high quality and relevance. 	<ol style="list-style-type: none"> 1. The Computer Science program aims at educating engineers [...] 2. for them to contribute to society [...] 3. to address problems in science, engineering and industry. 4. To that effect, we foster an environment of scholarly excellence [...] 5a. As a result we educate our students to be professionally competent [...] 5b. to have a distinct sense of ethics [...] 5c. to be committed to the life-long quest for knowledge [...] 5d. [to be committed to] the improvement of the quality of the software industry in Colombia.

PEO	Program's Mission	School's Mission	Institution's Mission
OP1	1,2,3,4	1, 2, 5a, 5c, 6	2,4,5
OP2	1,2,3,4	1, 2, 3a, 3b, 4, 5a, 5c	2,3,4,5
OP3	2,3,4,5a,5d	2, 3b, 4, 5a, 5b, 5c	1,2,3,4,5,6
OP4	1,3,4,5a,5b	2, 3b, 3c, 5c, 5d	1,2,3,4,5,6
OP5	1,4,5c	2, 3a, 4, 6	1,2,6

Figure 2.1: Relation between the Program and Institution Mission and the PEOs

- **Alumni and Computer Science Professionals** are part of the program's *Industry Board*. The coherence between the PEOs and the needs of the region are part of the agenda of such board.

In general, alumni are consulted about the quality of the program through surveys. Moreover, in the end of the *professional internship* course, the employer answers a survey where he/she explicitly evaluates: 1) the performance of the student with respect to the PEOs and the student outcomes; 2) whether the student performed well or not; and 3) whether the student met the needs of the company. Evaluations and suggestions from the employers are used by the Program Committee in order to propose changes when needed.

- **Colombian institutions** such as the MEN (Colombian Ministry of Education), ICFES (Colombian Institute for the Evaluation of the Education), CNA (national agency for the accreditation of programs) are part of the evaluation process of the PEOs through different mechanisms. The MEN and CNA, in particular, assess the quality of the program and they judge whether the program fulfills the national standards for Computer Science programs. Other Colombian agencies such as ACIS (association of Computer Science professionals) and ACOFI (association of engineering faculties) give general guidance that are followed by the Program Committee. Finally, **international associations** such as IEEE, ACM and ABET are also consulted when decisions about the orientation of the program and its curriculum must be taken.

2.E Process for Review of the Program Educational Objectives

The current definition of the PEOs took inspiration in the *Program Objectives* defined in 2004 in [1]. The initial proposal for the current PEOs came from the program director and later discussed with the Program Committee. Then, following the ABET assessment methodology, the PEOs were discussed with the different constituencies (students, alumni and Computer Science professionals). After some adjustments, the PEOs were established in the document [11].

The definition of the current PEOs coincided with the renewal of the CNA national accreditation of the program for 6 years. Hence, the Program Committee defined a window of 6 years to review the PEOs and, if needed, propose a new set of PEOs in 2018.

The Program Committee defined the following mechanisms to review the PEOs:

1. **Collecting information.** The following sources of information have been established as starting point for the assessment of the PEOs:
 - **Professional Internship Assessment:** Part of the final grade of the Professional Internship course is given by the employer of the student. In a survey, the employer evaluates, according to the performance of the student, whether the PEOs were achieved or not.

- **Industry Board Assessment:** The industry board in 2013 collected important information about the PEOs. With the aid of the human resource departments of the Industry Board's companies, the profile of the program's alumni was evaluated with respect to the definition of the PEOs. In each case, the companies were asked for the strengths and weakness of the program's alumni and whether the PEOs met the needs of the company (see the survey at [12]).
 - **Other Surveys:** As part of the continuous evaluation processes of the Javeriana University, students, faculty members and alumni answer surveys that are later used for the assessment of the PEOs.
2. **Evaluation Process:** The results of the above mentioned evaluation processes are examined by the Program Committee. Then, the Industry Board is convened and the results reported. For the time being, the industry board has not found the need of modifying the current PEOs.
 3. **Implement Changes:** As we said before, we do not foresee a new definition of the PEOs before 2018. Nevertheless, the previous evaluation processes has lead to some recommendations that the Program Committee must follow.

Criterion 3

Student Outcomes

3.A Student Outcomes

In the following we list the student outcomes (SO). For each one, we give a general description and also the performance indicators. Such definitions were used to give a precise meaning of the SO and to emphasize the aspects of interest of our program.

We note that the SO were initially proposed by the Program Committee relying on standard documents from ABET, CDIO and ACM as well as previous documents of the program, in particular, [1]. The current definition of the SO were documented in the Program Educational Project [11] (*Proyecto Educativo del Programa*) and presented to the stakeholders in different meetings (see e.g., the minutes [9, 10]). The document [11] is distributed among the first semester students and it is also available on the Program web page. Moreover, SO and performance indicators are available in the courses' syllabi.

- (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline.
Students must show an ability to effectively apply knowledge, techniques, principles and theories from continuous and discrete mathematics, logic, statistics, probability, physics as well as core computing and engineering knowledge to: (1) analyze, model and design systems and processes; and (2) propose and evaluate solutions to problems.
 - (a.1) Identify the fundamental scientific and engineering principles that govern a given process or system. (Knowledge)
 - (a.2) Solve problems related to the discipline and other areas by using knowledge, models and formalisms from computer science. (Application)
 - (a.3) Analyze data sets. (Analysis)
 - (a.4) Interpret mathematical model results to estimate accuracy and reliability. (Comprehension).
- (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

Students must show an ability to identify, analyze and formulate problems as well as an ability to identify functional and non-functional requirements. They must recognize the need for a proper and detailed problem/system definition to achieve a good design and solution.

- (b.1) Describe processes declaratively, abstracting away from implementation behavior. (Comprehension)
 - (b.2) Use mathematics, logic and engineering language to specify functional and non-functional requirements and properties of systems and processes. (Application)
 - (b.3) Synthesize the necessary information, evidence and facts to analyze a problem. (Analysis - Synthesis)
 - (b.4) Formulate hypothesis (Synthesis).
- (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
- Students must show an ability to design processes, components and computing systems to satisfy (possibly changing) requirements and needs. The design process must systematically follow a methodology and be expressed with the right artifacts and notation. Students must also show an ability to implement and evaluate computing solutions.*
- (c.1) Use standards of coding when implementing software components. (Application).
 - (c.2) Identify components, interactions, relationships and interfaces among components. (Analysis).
 - (c.3) Design processes and software artifacts by using appropriate techniques, tools and notation. (Synthesis)
 - (c.4) Evaluate a software component with respect to temporal and spatial complexities. (Evaluation).
- (d) An ability to function effectively on teams to accomplish a common goal.
- Students must be able to work collaboratively in projects and comprehend the multidisciplinary nature of a problem when appropriate. For this, they must (1) be responsible for their assignments and roles, (2) follow the directions and rules of the team, (3) accept feedbacks and criticisms, (4) value the knowledge, experience and background of other members, (5) communicate effectively to other members of the team and (6) be able to share and delegate responsibilities.*
- (d.1) Recognize the increasing role of computing in multidisciplinary settings. (Knowledge).
 - (d.2) Participate in team tasks and decision making. (Responding)
 - (d.3) Integrate points of view, information, feedbacks and criticisms to propose a solution (Synthesis).

- (d.4) Define task, roles and responsibilities (Application).
- (e) An understanding of professional, ethical, legal, security and social issues and responsibilities.
Students will recognize their professional and ethical responsibilities and they will act accordingly.
- (e.1) Identify the ethic codes related to the discipline. (Knowledge).
- (e.2) Show responsibility and professional behavior. (Valuation).
- (e.3) Identify pros and cons in ethical decisions mainly related to professional behaviors (Analysis).
- (e.4) Argument and justify ethical decisions. (Evaluation).
- (f) An ability to communicate effectively with a range of audiences.
Students must communicate effectively ideas, models, designs, and technical solutions both oral and written. They must also identify the audience and adapt the communication strategy to cope with technical and non-technical styles.
- (f.1) Produce effective communication in written with respect to: structure, coherence, flow, correct spelling, punctuation and grammar. (Application).
- (f.2) Communicate effectively with an appropriate language, style, timing, flow and non-verbal strategies according to the target audience (Application).
- (f.3) Use graphical resources to communicate and explain an idea. (Application).
- (f.4) Defend ideas with precision and clarity. (Evaluation).
- (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society.
Students must show an ability to interpret technical solutions in both in micro and macro contexts. Students must also demonstrate the ability to evaluate the socio-economic, political, and environmental implications of computing and engineering solutions.
- (g.1) Identify the world events which the computing and engineering activity likely affect. (Comprehension)
- (g.2) Use knowledge to identify impacts of computing and engineering in a given solution. (Application).
- (g.3) Analyze the local and global impact of computing and engineering. (Analysis).
- (g.4) Judge the impacts of computing and engineering in the world. (Evaluation).
- (h) Recognition of the need for and an ability to engage in continuing professional development.
Students must recognize the importance to engage in life-long learning to be aware of the new developments in the discipline. They must also show ability to process critically information and link it with prior knowledge.

- (h.1) Recognize the importance of both depth and breadth of knowledge. (Comprehension).
 - (h.2) Apply new knowledge to solve a problem or develop a solution (Application).
 - (h.3) Interpret and assess information from multiple sources and link it with prior knowledge. (Synthesis - Evaluation).
 - (h.4) Show willingness to learn new content through individual research and study. (Valuation).
- (i) An ability to use current techniques, skills, and tools necessary for computing practice. *Students will have the ability to use state-of-the-art hardware and software tools as well as modern methodologies and techniques for the practice of computing.*
- (i.1) Use software development tools. (Application).
 - (i.2) Use software design, simulation and modeling tools. (Application).
 - (i.3) Combine software and hardware tools to solve a problem. (Synthesis).
 - (i.4) Show flexibility to adapt to different programming languages and paradigms.(Valuation).
- (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. *They must recognize the importance of foundations in mathematics and computing to model and design computing artifacts. Students must identify principles and methods from computing and mathematics to tackle well-defined, open-ended or ill-defined problems in different areas and contexts. Such principles must play a fundamental role when taking decisions in the design process.*
Note: Unlike Outcome (a) (that is also related to design), the abilities described here presuppose that students are faced to problems where they have to combine knowledge and techniques from possibly different areas. A key aspect to asses j-abilities is that students must evaluate design decisions based on the principles of the discipline.
- (j.1) Recognize the importance of modeling when solving a problem. (Comprehension).
 - (j.2) Relate theoretical concepts and principles to practical problem solving. (Synthesis).
 - (j.3) Combine mathematical, computing and engineering principles that can be applied when modeling a situation. (Synthesis).
 - (j.4) Evaluate design decision based on mathematical and computing principles. (Evaluation).
- (k) An ability to apply design and development principles in the construction of software systems of varying complexity. *Students must show an ability to effectively handle software development projects. Students must choose and apply the right methodology to implement a computer-based solution according to the complexity of the situation.*

- (k.1) Follow schedules and adapt resources to meet milestones. (Application).
- (k.2) Implement and integrate software components that faithfully follow the design criteria. (Application).
- (k.3) Establish invariants and software properties. (Analysis).
- (k.4) Evaluate and verify software solutions with respect to system requirements and constraints. (Applications - Evaluation).

3.B Relationship of Student Outcomes to Program Educational Objectives

Table 3.1 shows how the SO prepare graduates to attain the Program Educational Objectives (PEO). In that figure a scale from 0 to 5 was used to represent the degree of correspondence between the PEO and SO. Later, in Section 4.A.3, we shall use the same scale to show how the SO are considered in each course, thus complementing the curricular integration matrix in Table 3.1.

Student Outcomes		EO.1	EO.2	EO.3	EO.4	EO.5	Total	%
A	Computing an mathematics knowledge	5	3		2	2	12	13.8
B	Analysis	2	4				6	6.9
C	Design abilities	2	5				7	8.0
D	Teamwork abilities			2	5		7	8.0
E	Professional responsibility			3	5		8	9.2
F	Effective communication			2	5		7	8.0
G	Ability to analyze impacts		1	3	3		7	8.0
H	Professional development			2		5	7	8.0
I	Use of modern engineering tools	2	5				7	8.0
J	Application of theory to modeling and design	5	5		2		12	13.8
K	Construction of software systems		5	2			7	8.0
Total		16	28	14	22	7	87	
%		18.4%	32.2%	16.1%	25.3%	8.0%	100.0%	

Table 3.1: Relation between the PEO and SO. (0) means no relation and (5) means a strong relation

The relative weights obtained with our method are very significant. On the one hand, the relative weights of the PEOs show that most valued are those related to the analysis and design of computing systems (PEO2) followed by the concern for ethics and professionalism (PEO4), application of knowledge (PEO1) and the excellence in contributions to society (PEO3) and finally the independent learning (PEO5). The low value given to PEO5 (in the sense of pursuing graduate studies) might be related to the culture that prevails in the professional community of Computer Science in Colombia in which, traditionally, highly rewarding positions can be reached without the need for a postgraduate degree. On the other hand, the relative weights of the student outcomes are more evenly distributed between the clearly technical SOs, i.e. A, B, C, I, J, K and those of more social character, i.e. D, E, F, G and H. Notwithstanding that, it is also satisfactory, from the point of view of an Computer

Science program, that the most valued SOs were the application of technical knowledge (A) and the application of theory to modeling and design (E). Furthermore, the relative weights of the SOs are very significant for the assessment of the Program because those weights determine how the curriculum supports the SOs, this will be explained in detail in Criteria 4 and 5.

3.C Process for the Establishment and Revision of the Student Outcomes

The current SO were initially proposed by the Program Committee drawing inspiration from previous definition of skills and abilities for the Program (see e.g., [1]) and standard documents from ACM and ABET. Following the ABET assessment methodology, the SO were discussed with the different constituencies (students, alumni and Computer Science professionals). The current version of SO is documented in [11].

As was the case for the PEOs (see Section 2.E) the definition of the current SO coincided with the renewal of the CNA national accreditation of the Program for 6 years. Hence, the Program Committee defined a window of 6 years to review the SO and, if needed, propose a new set of PEOs in 2018.

Besides the surveys described in Section 2.E, that also collect information about the SOs, the Program Committee defined as **direct mechanism of assessment** a numerical model where each course is responsible to evaluate some of the SOs. As we shall explain in detail in Section 4.A, this model relates the course's objectives with the corresponding SOs. Moreover, the numerical model requires that the course's evaluation instruments assess the performance of the students with respect to such SOs. Thus, grades are per se performance indicators and they are used by the Program Committee for decision making.

3.D Enabled Student Characteristics

As we detailed in Section 3.A, the students have the opportunity to attain the generic ABET characteristics (A-I) and also the specific student outcomes (J-K) for Computer Science programs.

Criterion 4

Continuous Improvement

4.A Student Outcomes

4.A.1 Program global assessment model

The following is the model for the global assessment of the Program:

Course level: Assessment of the contribution of courses to the student outcomes.

Program level:

- Assessment of courses.
- Assessment of the Undergraduate Project.
- Assessment of the Professional Internship.
- Assessment of results in the national examinations.

This model of program assessment is based only on direct measurements. The Program Committee took the decision to follow this strategy during the preparation of the Program for the ABET evaluation in the cycle 2015-2016. When the program assessment operates in steady state after 2016, the use of indirect measurements would be decided.

4.A.2 Planning of the program global assessment

We differentiate between two states: a transient state associated with the preparation of the Program to receive the ABET evaluation and a steady state after accreditation.

The preparation of the Program for the assessment and further evaluation for accreditation started in the first semester of 2011 (2011-1) and the cutoff semester for preparation of the Self-Study Report was 2014-2 (November 2014). The following actions were carried out up to the cutoff period:

- In 2011: definitions of: mission statements of the School of Engineering and of the Program; program educational objectives; student outcomes and their performance indicators; mapping of the student outcomes onto the curriculum; the program global assessment model. In addition, the Industry Advisory Committee was nominated.
- In the first semester of 2012: definition of the numerical model for the course assessment; selection of the courses for assessment and adaptation of their syllabi.
- In the periods 2012-2 to 2014-2: the assessment of student outcomes in courses was made. Some of the courses started the assessment in 2012-2 and the rest in 2013-1; but it was decided in 2013-1 that those courses would be assessed each semester starting in 2013-2. The assessment of the Undergraduate Project started in 2012-2 and continued thereafter. The assessment of the Professional Internship started in the 2014-2 semester.

The Program Committee decided to extend the semiannual assessment of courses, of the Undergraduate Project and of the Professional Internship up to the second semester of 2016 including the annual evaluation of the results in the national examinations. Furthermore, the Program Committee considered that this scheme of program assessment is sustainable within the capacity of the Program; therefore, this may well constitute the assessment scheme to be followed in the steady-state period after 2016, including the evaluation of the program educational objectives every five years.

4.A.3 Course assessment model

In this section the evaluation of student outcomes (SOs) through the curriculum is explained in detail. The section is organized in two components: the relationship between the SOs and courses, and an explanation of the numerical model used to evaluate courses. The results and analyses that follow apply to both the generic student outcomes A - I and to the program-specific student outcomes J and K.

Relationship between student outcomes and courses

This relationship was obtained by the same methodology used to relate the student outcomes with the program educational objectives explained in Section B, Criterion 3. Thus, for each course, the Program Committee, together with other faculty members, assigned the relationships with the SOs, and the weights indicating the levels of relevance of those relationships. The result of this exercise was the mapping of the SOs onto the whole curriculum of the Program, which is presented in Table 4.1.

Table 4.1: Relationship of courses with student outcomes. Relative weight in scale 0-5.

Course	A	B	C	D	E	F	G	H	I	J	K	Total
Fundamental Mathematics	2	2								1		5
Linear Algebra	2	2								1		5
Differential Calculus	2	2								1		5

Integral Calculus	2	2								1		5
Multivariate Calculus	2	2								1		5
Kinematics and Dynamics	2	2								1		5
Electricity and Magnetism	2	2								1		5
Numerical Analysis	5							3		2		10
Probability and Statistics	3	2							1	1		7
Discrete Mathematics	3								1	3		7
Oral and Written Expression				5		5						10
Humanities I				2	5		3					10
Humanities II				2	5		3					10
Theology I				2	5		3					10
Theology 2				2	5		3					10
English II							3					3
English III							3					3
English IV							3					3
Political Cons. and Democracy				4	5		3					12
Ethics				3	5		5					13
Organisational Management				3	5		3					11
Analysis and Design of Algorithms	3	2	3			2				5	1	16
Computer Architecture I	3		2	1					1	2	2	11
Computer Architecture II	4		4	1		4			2	2	2	19
Social and Professional Issues				1	5	5	5	2				18
Computability and Formal Lang.	4					1				4		9
Computer Graphics	5		4			4		1	3	3		20
Large Scale Software Development	1	5	5	3		1	2		3	1	5	26
Formal Methods	1	2			1		1		1	5	5	16
Web-based Services		2	1	1		1		2	1		5	13
Program. Fundam.and Data Str.	5		5			2		3	3			18
Info. Management and Modelling	2		5		1	1	2	2		5		16
Database Implementation	2		3	2				3	2	5	4	21
Artificial Intelligence	5					1	2	1		5		14
Introduction to Computer Science	3				2	2	4	2				13
Introduction to Programming	2		5	1				2	3	5		18
Introduction to Modelling		3	3			2	1			5	2	16
Laboratory of Programming			5					1	2			8
Logic in Computer Science	3									3	3	9
Objects and Middle Scale	3		5	1		1					3	13
Software Engineering Processes	4	5	5	1		2			2	1	2	22
Networks and Communication	3		1			2			2	2	2	12
Operating Systems	4							2	2	5	3	16
Fundamentals of Research		5		3	2	3	2	3				18
Undergraduate Project	2	5	5	5	5	5	5	5	2	5	3	47
Professional Internship	1	2	2	5	5	5	5	5	2	5	2	39
Engineering Economics	3						3					6
Animation and Simulation	2			3					3	4		12
Interaction and Sound	1							4	4			9
Intro. to Video Games Dev.				3		3			5			11
Artificial Intelligence for Games	4					1			5	5		15
Mobile Computing								5	5	5	5	20
Information Security	2				2		2	3	1	5		15
Multimedia Technologies	2					2		3	5		1	13

A number of conclusions are drawn from the above exercise:

- All of the student outcomes are abundantly supported throughout the curriculum and no course is left without a relationship with the SOs. Thus, the support to the SOs is redundant and every course is relevant to several of the student outcomes, to the program educational objectives and, finally, to the mission of the University.

- The connection of the curriculum with the institutional mission is tracked starting with the relationships between courses and the SOs in Table 4.1, then the connections between SOs and PEOs (Table 3.1, Section B, Criterion 3 and, finally, the relationships between the PEOs and the mission (Table 2.1, Section C, Criterion 2). An example of this exercise is presented in Figure 4.1 for the Formal Software Development course. Eventually, for every course of the Computer Science curriculum an analogous graph exists, thus ensuring the consistency of the curriculum with the mission of the University.

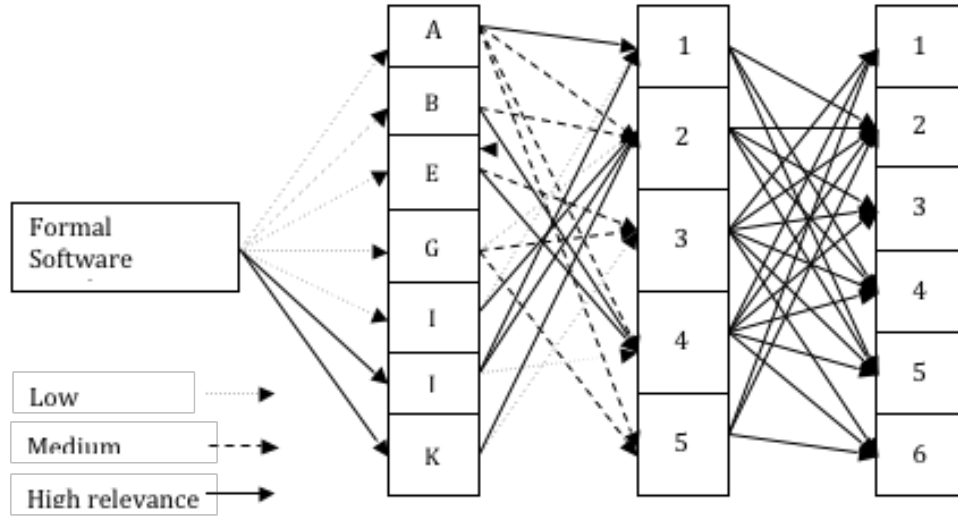


Figure 4.1: Example of curriculum integration: Formal Software Development course.

- For each course, the associated student outcomes with their levels of relevance constitute a “curricular formula” that guides the design of the syllabus and evaluation of the course.
- A necessary condition of our model of assessment is that the relative weights of the student outcomes of Table 3.1 should be preserved throughout the curriculum. Notwithstanding that, in practice, the use of integer levels of relevance between courses and student outcomes results in differences in the actual overall relative weights of the student outcomes with respect to those from the balance of Table 3.1. Thence, a tolerance range of ± 2 percentage point was established. Table 4.2 presents the results of that check of consistency, in which the first row contains the total sum of actual weights over the courses; the second row are the corresponding percentages; the third row are the expected percentages from Table 3.1; the fourth row are the expected sum of weights from the percentages of Table 3.1; and the fifth row are the difference between the actual sum of weights and the expected ones. As shown, those differences were within the tolerance, thus indicating that the actual allocation of relevance levels

to the relationships between courses and the student outcomes was consistent with the relative weights resulting from the relationship between the program educational objectives and the student outcomes.

In summary, the curriculum was shown to be consistent with the mission of the University resulting from an orderly and robust system of relationships through the program educational objectives, the student outcomes, and courses.

Table 4.2: Comparison of relative weights of student outcomes between Table 3.1 and Table 4.1

	Student outcomes											
	A	B	C	D	E	F	G	H	I	J	K	Total
Total	99	47	63	54	58	61	57	50	66	95	50	700
Percentage	14.6	6.7	9.0	7.7	8.3	8.7	8.1	7.1	9.4	13.6	71	14.1
Relative weight (Table 3.1)	13.8	6.9	8.0	8.0	9.2	8.0	8.0	8.0	8.0	13.8	8.0	13.8
Expected points	97	48	56	56	64	56	56	56	56	97	56	700
Difference(%)	0.3	-0.2	1.0	-0.3	-0.9	0.7	0.1	-0.9	1.4	-0.2	-0.9	

To carry out the program assessment, the Program Committee chose a sample of 10 courses out of the 24 discipline-specific core-curriculum courses (see Table 5.1 of Criterion 5). Those courses are located on the plan of study as follows and are presented in Table 4.3:

- Second semester: Programming Fundamentals and Data Structures.
- Fourth semester: Logic in Computer Science.
- Third semester: Discrete Mathematics for Computing.
- Fifth semester: Computer Architecture II, Computability and Formal Languages, and Information Management and Modeling.
- Sixth semester: Software Engineering Processes, Computer Graphics.
- Seventh semester: Formal Software Development.
- Tenth semester: Social and Professional Issues.

The relative weights of the student outcomes in this sample of courses do not coincide with those of Table 3.1. Thus, the relative weights of the student outcomes of technical character (A, C and I) and of social character (F) were overvalued relative to those in Table whereas those of the student outcomes of social character (D, E and H) and of technical character(B) were undervalued. This was inevitable because the courses selected for evaluation were typical technical core-curriculum courses; nevertheless, the Program Committee considered that this would not undermine the validity of the assessment model because all of the student outcomes were supported by the selected courses and the selected courses constituted a representative sample of key subject areas of the curriculum, i.e. the Mathematics and Basic Sciences area (with the Discrete Mathematics course), the Algorithms and Complexity area (with

Table 4.3: Courses selected for the program assessment

Courses	Student outcomes											
	A	B	C	D	E	F	G	H	I	J	K	Total
Prog. Fund. and Data Struct.	5		5			2		3	3			18
Discrete Mathematics	3								1	3		7
Logic in Computer Science	3									3	3	9
Computer Architecture II	4		4	1		4			2	2	2	19
Comp. and Formal Lang.	4					1				4		9
Info. Management and Mod.	2		5		1	1	2		5			16
Software Eng. Processes	4	5	5	1		2			2	1	2	22
Computer Graphics	5		4			4		1	3	3		20
Formal SW. Development	1	2			1		1		1	5	5	16
Social and Professional Issues				1	5	5	5	2				18
Total points	31	7	23	3	7	19	8	6	17	21	12	154
Relative weight (%)	20.1	4.5	14.9	1.9	4.5	12.3	5.2	3.9	11.0	13.6	7.8	100
Relative weight Table 3.1	13.8	6.9	8.0	8.0	9.2	8.0	8.0	8.0	8.0	13.8	8.0	100

the Computability and Formal Languages course), the Programming Languages area (with the Programming Fundamentals and Data Structure course), the Information Management area (with the Information Management and Modeling course), the Software Engineering area (with the Software Engineering Processes course), the Computational Science area (with the Formal Software Development course), the Social and Professional Issues area (with the course of the same name), the Graphs and Visual Computing area (with the Computer Graphics course) and the Architecture and Organization area (with the Computer Architecture II course).

The courses in Table 4.3 were required to structure their syllabi in a template called the “course registry”, an example of which is presented in Figure F.1 of the Appendix D.7.2. The course registry offers comprehensive information about the course and it is delivered to students at the beginning of the semester. Also, this template is an expanded and detailed version of the course syllabi contained in the Appendix 8.E of the Self-Study Report. Section 7 of the course registry is of critical importance because it details the whole curriculum integration pertaining to the course; in particular, it contains the following:

- Curricular formula: this is the set of student outcomes and relative weights assigned to the course indicated in Table 4.1.
- Performance indicators: these are the performance indicators assigned to each of the student outcomes of the course taken out of those that the Program Committee defined for the Program (Section C, Criterion 3). The list of performance indicators assigned to courses is presented in Table F.1 of the Appendix D.7.2.
- Integration of student outcomes, learning objectives, contents, and methodology: this is the set of connections that allow a course to develop the student outcomes associated with it. In other words, the student outcomes are associated with learning objectives, which in turn are related to specific topics that are delivered through a number of methodologies and pedagogical tools.
- Balance of evaluation: this describes the balance of the evaluation of the course through

the instruments of evaluation (exams, homework, projects, etc.) such that the relative weights of the student outcomes are preserved. This balance results from the numerical model used in our method of program assessment and that is explained in Section 3b ahead.

Evaluation of student outcomes in courses

A summary of the method used to evaluate the attainment of student outcomes in courses is presented in this section whereas the mathematical fundamentals of the method are presented in Section F.3 of the Appendix D.7.2.

As a result of the curriculum integration described in the above Section 4.A.3, the basis of the evaluation of the attainment of student outcomes must come from the results of the evaluation instruments of the course. In the classical grading of our courses, the overall grade of the student is the weighted average of the grades obtained in the evaluation instruments, which are assigned a percentage of the overall grade. In that method, the evaluation instruments are applied along the schedule of the course and follow very closely the progression of the contents. Thus, that is a contents-oriented grading and the grades in the individual evaluation instruments are usually obtained from the assessment of a percentage performance in solving problems, developing projects, and so forth. On the other hand, to assess the attainment of student outcomes a different perspective of the evaluation is needed. Such student outcome - oriented evaluation must account for the assessment of student outcomes but have to be done through the evaluation instruments that are normally used in the University. This is a very important constraint because the platform whereby faculty members upload course grades into the academic registry is parameterized in terms of conventional evaluation instruments, i.e. exams, projects, homework, etc., and not in terms of student outcomes; moreover, irrespective of the initiative to assess student outcomes, the academic registry still requires the grading of courses.

Therefore, from the above emerged the solution and model we devised to relate the evaluation of student outcomes and the standard grading of courses. Our solution was twofold: replacing the contents-oriented evaluation of courses with the ability-oriented evaluation, and basing the evaluation of the course entirely on the evaluation of the student outcomes. As a result, the evaluation of student outcomes and the grading of the course are no longer separate efforts but an integrated effort to evaluate the course only from the attainment of student outcomes and learning of the students. To avoid confusion with the traditional grading, hereinafter we will not refer to grades but to “results” of students in courses.

The mechanism to implement this scheme of evaluation is the evaluation balance of Section 7 of the course registry (see Figure F.1). The evaluation balance is the distribution of the evaluation of the student outcomes through the instruments of evaluation such that the relative weights of the SOs and of the instruments are simultaneously satisfied in the overall evaluation of the course. The relative weights of the student outcomes are those of the curricular formula of the course and those of the instruments of evaluation are defined in Section 8 of the course registry. An example of an evaluation balance is that of the Formal Software Development course presented in Table F.2 of the Appendix D.7.2. To facilitate the

understanding of the balance, integer “points” were used. Thus, from the curricular formula of Table 4.1, the course has the student outcomes A, E, G, and I, each with 6.25 percent of the total evaluation; outcome B with 12.5 percent; and outcomes J and K with 31.25 percent each; on the other hand, the instruments of evaluation are two exams, each of 30 percent, the course project with 25 percent and homework of 15 percent. Thence, one of the many possible ways to achieve this distribution is the one presented in Table F.2, in which it is confirmed that the relative weights of the student outcomes and of the instruments are very closely satisfied in the overall evaluation. However, the weights of the SOs need not be satisfied at the level of individual evaluation instruments. Once the evaluation balance is achieved, the next step is to structure the evaluation instruments accordingly. Again, there are many possibilities to that effect and this is left to an informed choice by the course coordinator, the only condition is that instruments follow the evaluation balance. An example of the distribution of evaluation is that of the an exam of the Formal Software Development course in the 2014-2 semester, presented in Table F.3 of the Appendix D.7.2 with the corresponding explanation. A wide variety of evaluation instruments are used by the courses selected for the program assessment, which is summarized in Table F.4 of the Appendix D.7.2. Exams are the most widely used instrument, followed by projects, homework and workshops. Finally, whatever the evaluation instrument is used, the actual evaluation of performance is guided by the performance indicators associated with the course (see Table F.1). For instance, in the Stochastic Processes course, student outcome A is evaluated with exams and homework (see Table F.4) with all of the performance indicators associated with that outcome (see Table F.1); thence, those instruments have to be structured to evaluate the application of knowledge of sciences, mathematics and core-curriculum topics (see Section C, Criterion 3).

At the end of the semester, course coordinators submit to the Program administration a “course report” and the listing of course results, to be reviewed by the Program Committee. The template of the course report is presented in Figure F.2 of the Appendix D.7.2. Course reports provide comprehensive information as to the operation of courses: the attainment of student outcomes and learning objectives, the mechanisms of evaluation, the appropriateness of processes and resources, and, very importantly, changes implemented and recommendations.

There are several advantages derived from this method of course evaluation:

- It is systematic and numerically consistent with the integration from the curriculum to the mission of the University, i.e. the course results are consistent with the relative weights of student outcomes and program educational objectives.
- It avoids the use of two separate evaluation systems: one for student outcomes and the other for traditional grades, therefore reducing the workload on faculty members and mitigating resistance to change.
- In practice, the method reduces significantly the amount of documents and information the Program administrators need to archive about the attainment of student outcomes. As indicated in the above paragraph, this is reduced to the course report and the listing of results in the student outcomes of the course.

- As the course results are expressed in the same scale of ordinary grades, this allows comparisons with the rest of evaluations in the Program.
- Very importantly, the information that eventually reaches the stakeholders, in the guise of traditional grades, is the evaluation of abilities. This is fundamental because very important decisions for students are taken upon grades: permanence in the Program, scholarships, admission to postgraduate studies, and recruitment, to mention but a few. Clearly, those decisions would be better informed if they were made upon the abilities of students for professional practice, i.e. student outcomes. Conversely, if the evaluation of student outcomes were made separately from the grading of courses, that very valuable information most certainly would not reach the stakeholders. Thus, the system is fundamentally fair and beneficial to students by giving useful value to their performance in student outcomes. Therefore, we are confident that our method is an alternative to the traditional practice of grading courses only on a contents-oriented perspective.

In summary, the evaluation of student outcomes in our model of program assessment proceeds through the steps shown in Figure 4.2.

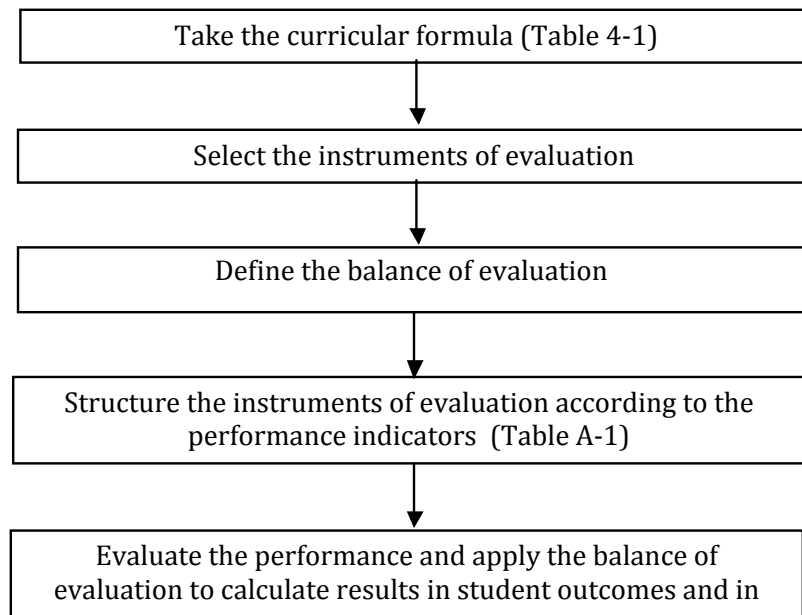


Figure 4.2: Summary of the evaluation of students outcomes in courses

4.A.4 Assessment at the program level

Recalling Section 4.A.1, the four sources of information for the program assessment are the results in the courses selected for evaluation, in the Undergraduate Project, in the Profes-

sional Internship and in the national examinations.

The results in the assessment courses were obtained through the mechanisms explained in Sections 4.A.3. Each of those courses produced an overall result for each of the corresponding student outcomes. Thus, the contribution of the courses to the student outcomes was calculated by the weighted average of each course on each student outcome. The individual relative weights are those from Table 4.3.

Following our method, the evaluation of the Undergraduate Project and of the Professional Internship was made with their corresponding curricular formulae of Table 4.1. In that numerical sense, the evaluation of these two courses is no different from that of the ordinary courses selected for evaluation; nevertheless, because of their location at the end of the plan of study, these courses bear an special significance as capstone activities, and this is why the Program Committee decided to treat their results as a separate source of the program assessment.

The national examinations are called the “Saber Pro” exams and are administered by an agency of the Ministry of Education twice a year. In this Report we present the results for 2012 and 2013, which are the latest available for analysis. All graduation candidates are required to take the Saber Pro exams as a graduation prerequisite, but there is no passing grade. These exams are constructed in terms of generic and program-specific competencies. In that context the concept of competencies is closely related to skills. The generic competencies are: citizenship, written communication, English, critical reading, and quantitative reasoning. The specification of the program-specific competencies changed from 2012 to 2013, so that a description of those is deferred to Section 4.A.5. The results on the exams are reported as averages and standard deviations of the Programs students, and only for the generic competencies comparisons are made with the national averages. Because of the particular structure of the Saber Pro exams it is difficult to establish a direct correspondence with the student outcomes and the results of courses. However, we proposed the following qualitative correspondence with the generic competencies:

- Student outcomes E and G: related to the citizenship competency.
- Student outcome F: it is clearly related to the written communication, English, and critical reading competencies.
- Student outcomes A, B, C, J, and K: related to the quantitative reasoning competency.

Thus there is no correspondence for the student outcomes D and H. Regarding the program-specific competencies; we proposed that the technical student outcomes A, B, C, I, J, and K be related to all the program-specific competencies.

4.A.5 Attainment of the student outcomes

The attainment of student outcomes is discussed in this section for the evaluation courses, for the Undergraduate Project, for the Professional Internship and for the results in the national examinations. The Program Committee set the following minimum performance criteria:

- 60 percent, i.e. a score of 3.00 in our grading scale, for the overall performance in the evaluation courses, in the Undergraduate Project, and in the Professional Internship.
- 50 percent pass ratio in each student outcome.

These results and analyses presented in this section correspond to both the generic characteristics A-I and to the program-specific student outcomes J and K.

Attainment of student outcomes in evaluation courses

The results of student outcomes in the evaluation courses are presented in Table F.5 of the Appendix D.7.2 for the five semesters of observation, 2012-2 to 2014-2. A great deal of information is reported in this table:

- Individual cell figures are the mean results, over the number of students, in each student outcome assigned to a course.
- The overall results of student outcomes are the weighted averages through the set of courses according to the relative weights of Table 4.3 and to the number of students in the courses.
- Cell figures in the “overall” panel are the weighted averages of the results of courses in the corresponding student outcomes in each semester according to the relative weights of Table 4.1, i.e. averaging by rows.
- The overall results in each semester in the “overall” panel are the weighted averages of the course results according to the relative weights of the courses (Table 4.3) and to the number of students in the courses, i.e. these are the overall attainment of student outcomes of the set of evaluation courses.

The discussion and analysis of the attainment of student outcomes was made with four criteria: the evolution of overall course results (last panel of Table F.5), the evolution of overall results in student outcomes of Figure 4.3, the statistical characterization of individual results in student outcomes of Table 4.5, and the frequency distribution of those individual results of Figure 4.4. For convenience and given the importance of the evolution of course results, the “overall” panel of Table F.5 is reproduced here as Table 4.4.

From Table 4.4, seven of the ten evaluated courses showed increasing overall results over the five semesters of observation. To The only courses with decreasing overall results were Programming Fundamentals and Data Structures, with an important overall decrease of -17.3 percent; Social and Professional Issues, with an overall decrease of -5.1 percent; and Software Engineering Processes, with a slight decrease of -1.7 percent. On the contrary, the courses with significant overall increase of results, of more than 10 percent, were Discrete Mathematics, Computer Architecture II, Computer Graphics, and Logic in Computer Science. On the other hand, the means of the semester overall results were higher than the pass at 3.00 for nine of the evaluated courses; only the Programming Fundamentals and

Courses	Academic periods				
	2012-2	2013-1	2013-2	2014-1	2014-2
Prog. Fund. and Data Struct.	2.96	2.86	2.27	3.25	2.45
Discrete Mathematics	3.06	3.26	3.17	3.31	3.63
Logic in Computer Science		3.15	3.38	3.50	3.47
Computer Architecture II		2.94	3.36	3.43	3.37
Comp. and Formal Lang.		3.05	3.17	3.14	
Info. Management and Mod.	2.75	3.10	3.36	3.53	2.91
Software Eng. Processes	3.52	3.84	3.00	3.40	3.46
Computer Graphics		4.12	4.10	4.55	4.83
Formal SW. Development	3.17	2.84	3.06	3.04	3.26
Social and Professional Issues	3.98	3.57	3.94	4.48	3.78
Overall	3.22	3.26	3.26	3.53	3.29

Table 4.4: Overall results of evaluation courses

Data Structures course had a mean lower than the pass, i.e. 2.76. Regarding the program overall results along the five semesters of observation, those increased from 3.22 in 2012-2 to 3.53 in 2014-1, but decreased to 3.29 in 2014-2; thus representing an overall increase of 2.26 percent. However, the Computability and Formal Languages course was not evaluated in 2014-2, which might have contributed to the decreasing overall results in that semester.

With regards to the evolution of overall results in student outcomes, Figure 4.3 shows outcomes with important variability, such as B, C, H, and I; whereas the outcomes A, E, G, J, and K showed a more orderly evolution through the observation. The student outcome B, C, and H showed decreasing trends in overall results, whereas the other showed the opposite behaviour. The Programming Fundamentals and Data Structures course contributed in 2014-2 to decreasing overall results of the student outcomes C, F, and H. Also, the abnormally low results in student outcomes B and D in 2013-2 were determined by low overall results in the Formal Software Development course, for the former, and Computer Architecture II and Software Engineering Processes courses, for the latter. On the other hand, the means of the overall student outcomes results over the five semesters of measurements were all above the pass at 3.00; and the mean of the overall program results was 3.31. As a result of the numerical consistency of our program assessment, the increase in the overall results of student outcomes was 2.26 percent, coinciding with the result reported in the above paragraph. Therefore, those results are satisfactory as they show a slow but consistent improvement of student outcomes results through the period of observation.

The statistical characterization of individual student results of Table 4.5 allows a number of comments. First, the overall mean of individual results was 3.28 over the period of observation, which almost coincides with the value of 3.31 for the simple mean of the overall results of Table 4.4. Thus, the mean attainment of student outcomes in the 10 courses selected for evaluation was equivalent to a percentage performance of 66 percent, which met the objective of minimum performance of 60 percent; and all of the student outcomes had average

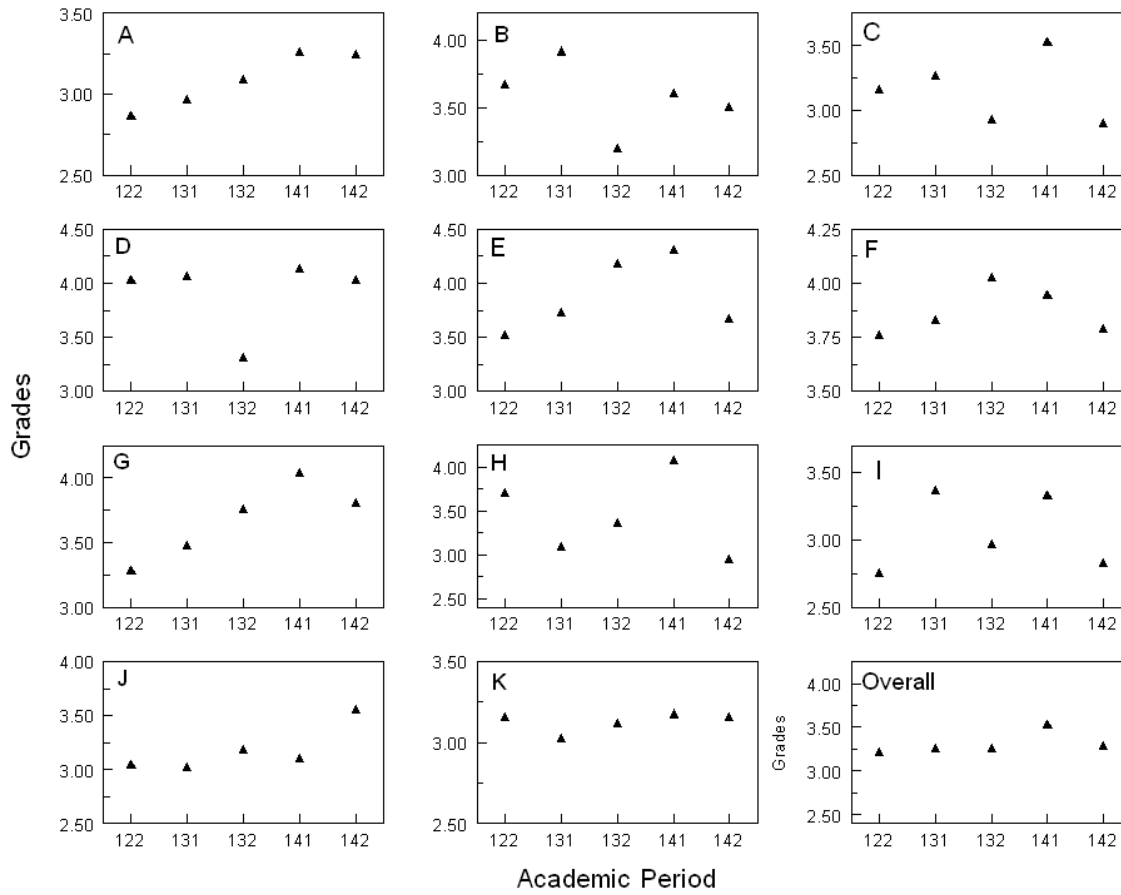


Figure 4.3: Evolution of the overall grades on student outcomes of the evaluated courses.

results higher the pass at 3.00. Second, the overall pass ratio in student outcomes was 67 percent, and the percentage pass ratio on each student outcome was higher than the minimum of 50 percent.

Table 4.5: Statistical characterization of the individual results on student outcomes of the evaluated courses

Courses	Student outcomes											
	A	B	C	D	E	F	G	H	I	J	K	Σ
Number of entries	391	102	195	122	135	271	135	113	306	291	176	425
Mean	3.05	3.41	3.17	3.97	3.85	3.67	3.65	3.56	3.22	3.25	3.09	3.28
Standard deviation	0.84	0.74	0.89	0.72	0.84	0.75	0.82	0.90	1.15	0.94	0.92	0.71
Coeff. of variation	0.28	0.22	0.28	0.18	0.22	0.20	0.22	0.25	0.36	0.29	0.30	0.22
Minimum	0.00	1.80	0.90	0.90	1.36	0.94	1.04	0.00	0.30	0.13	0.45	1.53
Maximum	5.00	4.88	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	4.95	4.96
Pass percentage	52	71	55	92	86	82	81	72	61	58	60	67

On the other hand, the whole distribution of individual results was characterized by a relatively small coefficient of variation of 0.22, thus indicating a narrow dispersion of results as confirmed by the frequency distributions of results of student outcomes of Figure 4.4. As these plots show, the distributions for the student outcomes of technical character A, B, C, and J, and to a lesser extent K, and of social character F and G conformed well to a normal-like distribution; whereas the distributions were more biased for the rest of student outcomes; also, the overall distribution also conformed to a normal distribution. A likely cause of these findings is that the outcomes of technical character were evaluated over a much larger number of students (as confirmed by the number of entries reported in Table 4.5; and that faculty members might be much more skilled at evaluating technical abilities; nevertheless, the overall distribution allows claiming the appropriateness of our instruments to evaluate the performance on student outcomes.

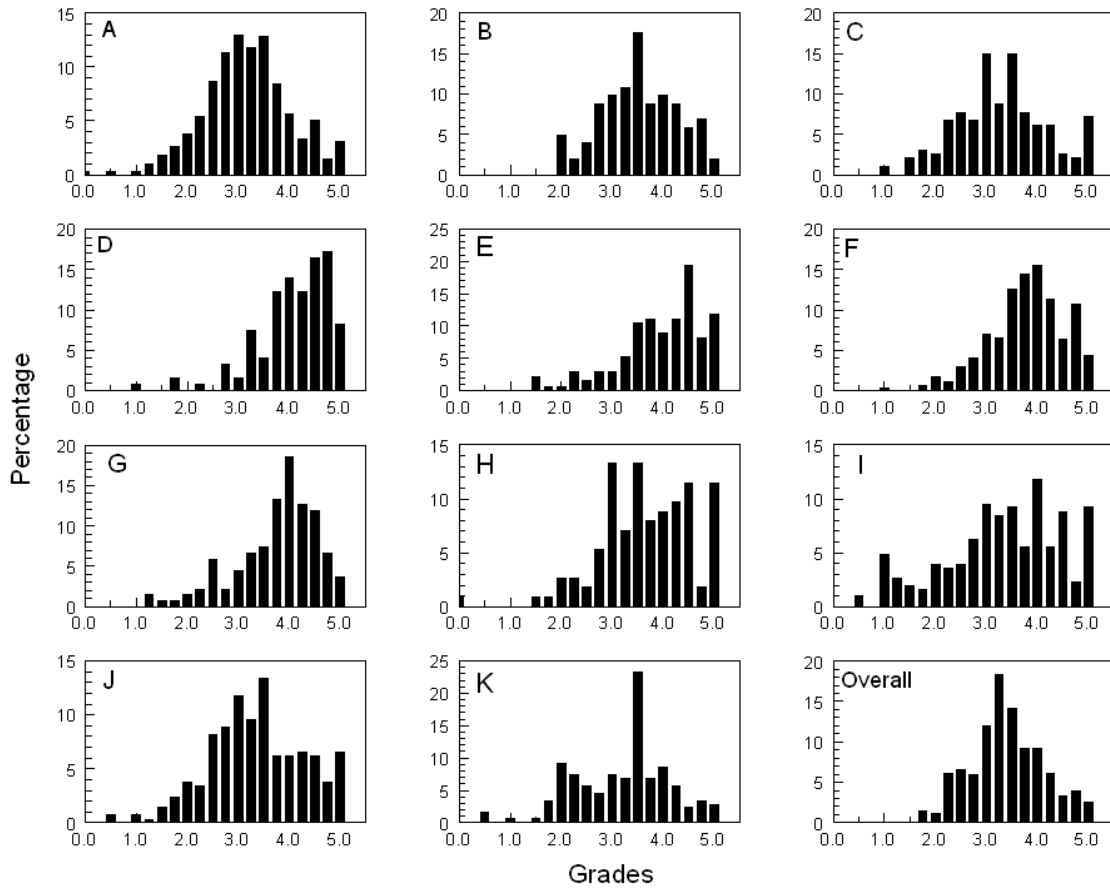


Figure 4.4: Frequency distribution of individual grades on student outcomes of the evaluated courses.

Attainment of student outcomes in the Undergraduate Project

The results of the evaluation of the Undergraduate Project are presented in Table 4.6. The evaluation of this course started on the first semester of 2013 and continued to 2014-2.

Table 4.6 shows that the student outcome results changed, in cases significantly, from one semester to the other, but the overall trends were increasing for nine of the student outcomes; only for the outcome G the overall trend was decreasing and for the outcome I the overall change was negligible. On the other hand, the overall results of the Undergraduate Project increased by 7.6 percent over the observation, with a mean of 4.50. Therefore, the overall performance in student outcomes in this capstone course was 90 percent, well above the minimum for acceptance of 60 percent; and it must be highlighted that the Undergraduate Project is developed around realistic design problems and the evaluation involves, in cases, external evaluators.

Student outcomes									
A					B				
2012-2	2013-1	2013-2	2014-1	2014-2	2012-2	2013-1	2013-2	2014-1	2014-2
4.33	4.07	4.16	5.00	4.66	4.17	4.63	4.56	4.52	4.40
C					D				
2012-2	2013-1	2013-2	2014-1	2014-2	2012-2	2013-1	2013-2	2014-1	2014-2
4.50	4.21	4.50	4.91	4.89	4.33	4.83	4.05	4.70	4.65
E					F				
2012-2	2013-1	2013-2	2014-1	2014-2	2012-2	2013-1	2013-2	2014-1	2014-2
4.17	4.76	4.21	5.00	4.86	4.00	4.44	3.90	4.74	4.40
G					H				
2012-2	2013-1	2013-2	2014-1	2014-2	2012-2	2013-1	2013-2	2014-1	2014-2
4.92	4.64	4.42	4.58	4.77	4.33	4.83	4.25	4.48	4.80
I					J				
2012-2	2013-1	2013-2	2014-1	2014-2	2012-2	2013-1	2013-2	2014-1	2014-2
4.93	4.45	4.21	5.00	4.92	4.33	4.12	4.08	4.52	4.55
K					Overall				
2012-2	2013-1	2013-2	2014-1	2014-2	2012-2	2013-1	2013-2	2014-1	2014-2
4.00	4.05	4.13	4.70	4.69	4.35	4.50	4.23	4.71	4.68

Table 4.6: Average results on student outcomes of the Undergraduate Project

To explain the difference in average performance between the evaluation courses, of 66 percent as discussed in Section 4.A.5, and that of 90 percent in the Undergraduate Project, two hypotheses were put forward. First, the populations that were evaluated pertained to different stages along the plan of study; with the evaluation courses being located between the fourth and the eighth semesters, whereas the Undergraduate Project is evaluated after the completion of the plan of study. Thence, the evaluation actually measured students at different stages of maturity and of development of knowledge, abilities and skills. Second, the evaluation of the Undergraduate Project is made over the Undergraduate Project proper, which receives a good amount of feed-back from the project supervisor and other advisors; on the contrary, the evaluation of ordinary courses is largely based on individual work. Therefore, all of the above supports the important conclusion that our scheme of evaluation actually measured the progress of the attainment of student outcomes along the plan of study.

Attainment of student outcomes in the professional Internship

The overall results of the evaluation of the Professional Internship started in the 2014-2 semester, and are presented in Table 4.7.

As it will be explained in Section 5.A.7 of Criterion 5, the evaluation of the Professional Internship is carried out by a faculty member plus an industry supervisor, with relative weights of 60 percent for the former and 40 percent for the latter. As seen in Table 4.7, the average percentage performance was 94 percent. Although the results reported here were only for one semester, for a class of 5 students, this average percentage performance appears to coincide very closely with that reported in Section 4.A.5 for the Undergraduate Project. Therefore, these two pieces of information confirm that the performance in the student outcomes is about 90-94 percent when measured about realistic problems and workplace situations.

Student outcomes	Overall results	Student outcomes	Overall results
A	5.00	G	4.50
B	5.00	H	4.83
C	5.00	I	5.00
D	4.78	J	4.00
E	5.00	K	4.50
F	4.50	Overall	4.68

Table 4.7: Overall results of the evaluation of the Professional Internship in the 2014-2

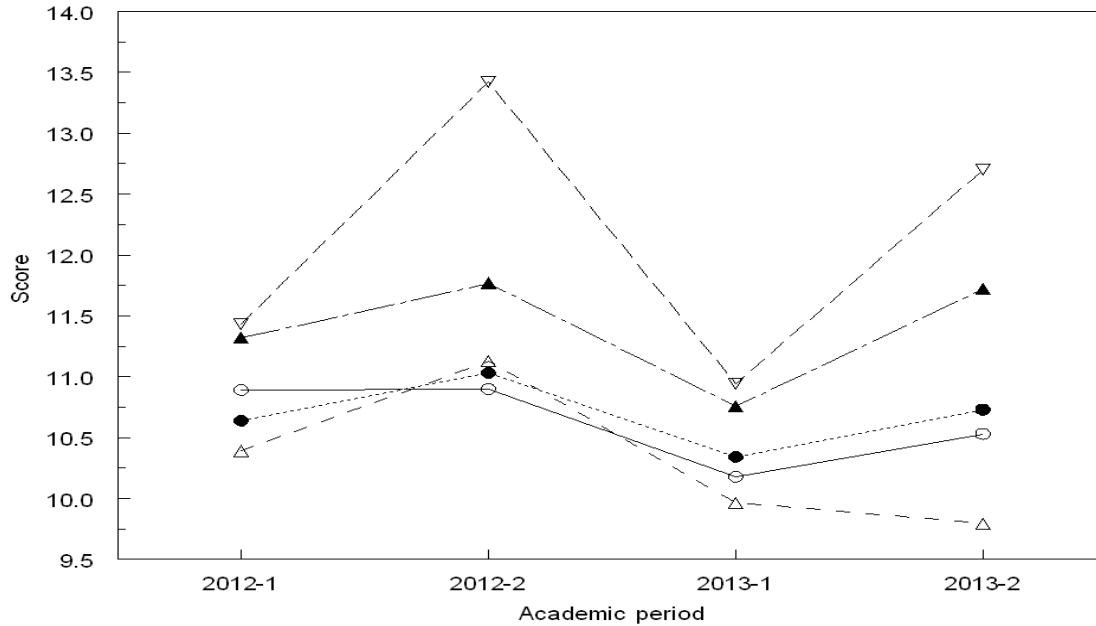
4.A.6 Results in the national examinations

The results in the national examinations are the fourth input to the global program assessment. With regards to generic competencies, Table F.6 of the Appendix presents the annual averages in 2012 and 2013, the number of students evaluated, the associated standard deviations and the averages of the national reference. The evolution of semester averages underlying those annual averages is presented in Figure 4.5.

The results of Figure 4.5 and Table F.6 show that the overall trends were decreasing for the five generic competencies, although those were very slight for the English, critical reading, and quantitative reasoning competencies. On the other hand, except for the written communication competency in 2013, the annual averages of the other competencies were higher than those of the national reference, yet within the standard deviation of the studied population.

A number of considerations are important. First, it is not possible to compare these results with those from the evaluation of courses, not only because the scales are different and the equivalences between the generic competencies and the student outcomes are only qualitative, but because the students examined in the exams of 2012 and 2013 pertained to cohorts that had no previous experience with the systematic evaluation of competencies, let

Figure 4.5: Semester average results in the generic competencies of the Saber Pro exams. \circ citizenship; \triangle written communication ; ∇ English; \bullet critical reading; \blacktriangle quantitative reasoning



alone student outcomes. Notwithstanding this, it was satisfactory that the results were higher than the group of reference in all cases but one, and by virtue of the proposed equivalences mentioned in Section 4.A.4, this conclusion could be extended to the student outcomes A, B, C, E, F, G, J, and K.

Regarding the program-specific competencies, those are also reported in Table F.6. Those results are reported here for three modules: Module 1: formulation of engineering projects; Module 2: software design; and Module 3: scientific thought. From the results of Table F.6, the results increased over the two periods of observation for Module 1 and decreased for Module 3. To qualitatively relate the program-specific competencies to the student outcomes, it was proposed that Modules 1 and 2 bear a relationship with the outcomes B, C, I, J, and H; and that Module 3 clearly corresponds to the outcome A.

This distribution by quintiles of the results in generic and program-specific competencies is presented in Figure 4.6.

The quintiles were defined as: first: low; second: low; third: average; fourth: good; and fifth: excellent. The evolution of quintiles was largely consistent with the results of Figure 4.5 and Table F.6, i.e. the increase in the first two quintiles all the generic competencies but the quantitative reasoning competency. The opposite was observed regarding the program-specific competencies: the upper two quintiles increased importantly in the three competencies studied. On the other hand, it was satisfactory that the population in the

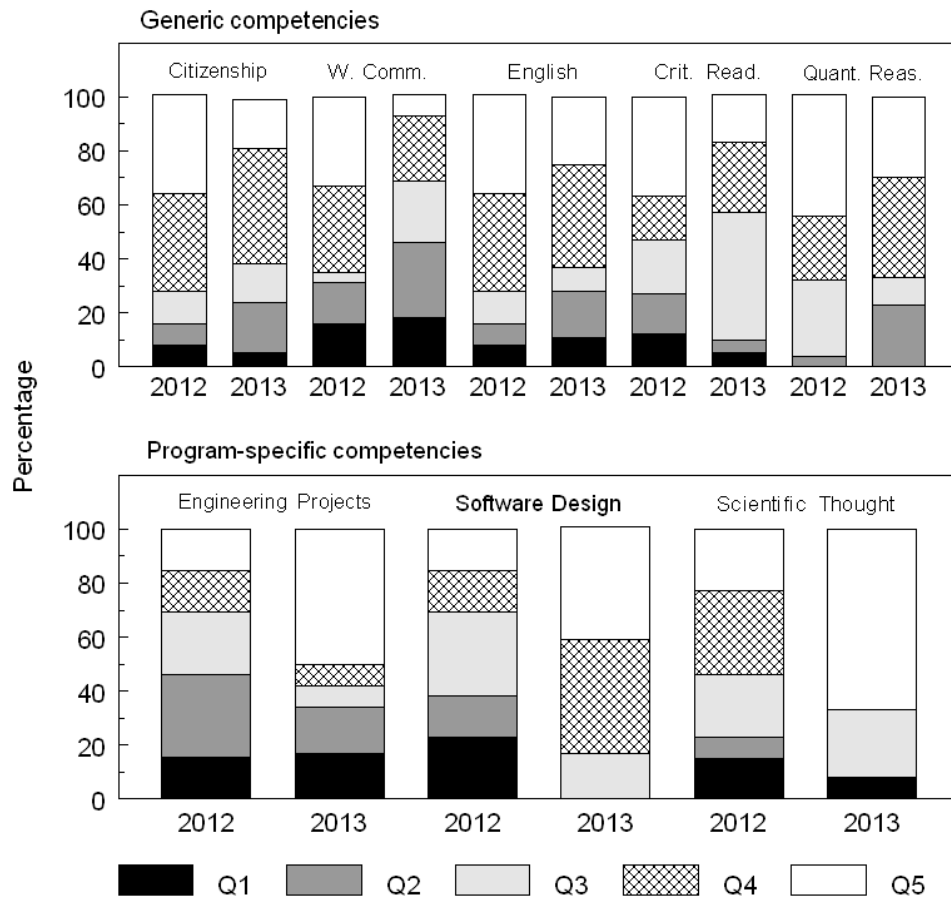


Figure 4.6: Quintile distribution of results in the Saber Pro exams. Q1: first quintile; Q2: second quintile; Q3: third quintile; Q4: fourth quintile; Q5: fifth quintile

upper two quintiles ranged between 30 and 90 percent for the generic and program-specific competencies; and, with due caution, this might be consistent with the already discussed average performance in the evaluation courses and in the Undergraduate Project and the frequency distribution of individual results.

4.A.7 Overall conclusions about the global process assessment

The findings presented thus far are summarized as follows:

- Through a system of orderly relationships, the curriculum was shown to support the student outcomes, the program educational objectives and the mission of the University.
- Student outcomes were evaluated in courses with a numerically consistent model that ensured the abovementioned system of relationships. This system was shown to offer

distinct advantages for the administration of the program assessment. The courses selected for evaluation constituted a sample sufficiently representative of the discipline-specific core curriculum.

- The overall performance in the student outcomes over five semesters in the representative sample of courses was equivalent to 66 percent, which met the minimum acceptance performance of 60 percent. These results appear to improve steadily with time.
- The minimum pass ratio of 60 percent in each student outcome was met.
- The overall attainment of student outcomes in the Undergraduate Project was 90 percent, and was shown to be relatively stable through the evaluation; and it was 94 percent in the Professional Internship.
- Although weakly related to the student outcomes, the results in the national examinations were largely higher than the national reference over two years of measurement.

Based on the above, our conclusion was that the Industrial Engineering met the objectives set out for the program assessment and satisfies the institutional mission and the interests of its stakeholders.

4.B Continuous improvement

This Section presents the continuous improvement actions regarding the evaluation courses, the results in the national examinations and a summary of the proposed improvement of the program.

4.B.1 Continuous improvement in the evaluation courses

From the course reports, the following is the summary of significant strengths and weaknesses found by course coordinators, of the recommendations they made, and of the improvement actions approved by the Program Committee to be implemented in the those courses.

Programming Fundamentals and Data structures: In general, the results of the course appear to have been impaired by students insufficiently motivated to the course, who do not dedicate the study time required by the courses credit hours, and by others who do not submit all of the assigned homework. Since 2014-1 a laboratory was included and it was highly recommended to dedicate more laboratory practices to programming portable devices and to use the hardware of the Animation and Sound laboratory.

In 2014-2 it was recommended to reduce the time dedicated to the topic of complexity from 3 to 1 week to allow more time for other topics and for laboratory practices. It was also proposed that the course coordinator completes a new version of the course notes.

Logic for Computer Science: A revision of the course formula was requested in 2013-1, with the current formula being introduced since 2013-2.

The performance in learning objectives was adequate, although students showed difficulties at translating propositions from the natural language to the formal language of the logic of propositions.

Workshop sessions were introduced since 2013-2. In 2014-2, a workshop every fortnight was implemented to practice formal demonstrations in LP and LPPO with COQ.

Discrete Mathematics: In 2013-1 it was requested to change the level of relevance of the student outcome J from 3 to 2 and increase the number of homework related to outcome I. Those changes came into effect in 2013-2 and improved the attainment of learning objectives.

In 2014-1 and 2014-2 the time dedicated to problem solving was increased. In 2014-2 it was requested to review the balance of evaluation to adjust it better to the course formula.

In general, the learning objectives are met, i.e. to describe the properties and operations with integer numbers, to use the counting techniques and recursive relationships, and to use Booles algebra.

Computer Architecture II: It was requested in 2013-1 to change the relative weight of the student outcome I from 1 to 2; and the change was approve to come into effect in 2013-2. This allowed a better balance between concepts, designs and the use of implementation tools.

The attainment of learning objectives is adequate, but students appear to have difficulties organizing their study time and duties in the semesters.

Computability and Formal Languages: In 2013-1 it was requested to increase the relative weight of the student outcome F from 1 to 2; and that change came into effect in 2013-2.

In general, the learning objectives are reached adequately, but it appears that students do not dedicate the study time required by the credit hours and by the proper submission of the course project. This has been observer all along the course observation. Starting in 2014-2, the revisions of the course project started earlier in the semester.

Information Management and Modeling: In 2013-1 it was requested to revise the relative weights of the student outcomes A, C, and I. Those changes were approved to start in 2013-2, thus constituting the reported formula of the course.

In general, the performance in learning objectives was only partially satisfactory. Starting in 2014-1 the time for practices was increased, but seems not to be enough. The students suggested including more examples of databases using Object Oriented designs.

It was recommended in 2014-2 to change the sequence of the end topics of the course to PL/SQL, functional dependencies, and normalization; this might allow for more time to complete workshops before submitting the course project.

Software Engineering Processes: The performance in learning objectives is adequate. Along the whole observation it appears that student do not dedicate the study time required by the credit hours, it is common that student do not submit all of the homework and course projects; thus impairing the course overall results.

Since 2013-1 it was recommended to implement visits to software companies. In 2014-2 the topic of Personal Software Process was introduced, and it was recommended to implement the use of UML as a modeling language.

Computer Graphics: In 2013-1 it was requested to distribute the evaluation of the course semester along the semester; this change was approved to come into effect starting in 2013-2. Also, starting in 2013-2, more interactive tools were introduced in homework and laboratory reports.

The attainment of learning objectives was very satisfactory; in particular, the use of transformation of objects in space, the use of graphic data structures, and the use of illumination and textures.

In semesters before 2014-2, students were free to choose the subjects of the course project, but in 2014-2 the subject was the modeling of the students own rooms, with satisfactory results.

In 2013-2 and 2014-1 it was requested to have access to ACM Digital Library, this came into effect starting in 2014-2.

Formal Software Development: In general, the learning objectives are developed adequately; however, the greatest difficulties are related with the strategies to implement interactive proofs.

In 2013-1 it was recommended to increase the amount of homework to improve the practical component of the course. As a result, in 2013-2 the amount of laboratory work was increased. Also in 2013-2 it was recommended to introduce topics related to programmer-oriented verification. In addition to that, in 2014-1 the contents included the modeling of software and non-software systems, and automatic proof systems. Also, in 2014-1 it was recommended to introduce the methodology of distributing the several components of a problem amongst groups of students, which was continued in 2014-2.

In 2014-2 it was recommended to implement the hybrid methodology of translating an EventB system to Java-JML, and to ensure the availability of Rodin and ProB in the computer rooms. It was reported in 2014-2 that students appear not to dedicate the study time required by the credit hours; in particular, students fail to submit all the assigned homework, and this is not trivial because all the practical component of the course (using Rodin) is developed through homework.

Social and Professional Issues: In general, the performance in learning objectives was adequate. Starting in 2014-1 the time dedicated to debating ethics and professional topics was increased, and one session was dedicated to improving writing skills.

In 2014-2 two topics were introduced: client service, and information security. Also in 2014-2, class attendance improved significantly and students requested inviting experts in information law.

4.B.2 Improvement plan for the results in the national examinations

By end of 2014-2, the Academic Vice-rector required every academic program to submit a plan to improve their results in the Saber Pro exams. Those plans should be implemented starting in 2015-1 and were structured around four strategies: students, faculty, curriculum, and culture. The following is the summary of the improvement plan submitted by the Computer Science Program:

- Students: although the national exams are a graduation requisite, there is no passing grade; therefore, it is presumed that students might not be committed enough to perform to the best of their abilities. Therefore, the strategy is to encourage students to their best performance; familiarize them with the structure of the exams and the sort of questions and problems; include, in selected courses, evaluations resembling those of the national exams and to schedule a number of drills in the months leading to the exams.
- Faculty: the basic strategy is to train faculty members in the development and evaluation of student outcomes and competencies but is also important to familiarize faculty members with the structure of the exams and the sort of questions and problems and with the relationships between the generic and specific competencies of the Saber Pro tests and their relationships with the student outcomes associated with courses.
- Curriculum: it is necessary to continue adjusting the relationships between courses and the student outcomes based on the results of the evaluation of courses. In this regard, the structure of the program assessment is considered as the appropriate tool to benefit the overall performance of students and, thence, the results in the national examinations.
- Culture: this is entirely related to the continuous improvement process derived from the program assessment. It is also important to promote in faculty and student awareness about the renewal of the national accreditation, the results in the national exams and the results of the program assessment.

4.B.3 Overall improvement of the Program

From all of the above, the most important weakness and need of improvement are the training of faculty members in the development and evaluation of student outcomes. The program assessment carried out in preparation for the ABET evaluation was a new endeavour for the Computer Science Program and it was expected that faculty members would face a new

paradigm of course delivery and evaluation. The School of Engineering, jointly with the Office of the Academic Vice-rector will start in the inter-semester period of 2015 a continued program of faculty training to address this weakness and need.

On the other hand, the Council of the School of Engineering approved in November 2014 extending to all core-curriculum courses the course evaluation method used in the program assessment thus far. The goal is to have the extension completed in the 2016-1 semester and the Program Committee will define in 2015-1 the scheduling of the extension. This will allow having a uniform method of course evaluation; improving the development and attainment of student outcomes; improving the statistical distribution of results; adding flexibility to the set of evaluation courses and bringing more courses to support the student outcomes with very few courses assigned, such as outcomes I and J.

4.C Additional Information

Copies of representative samples of the assessment instruments or materials referenced in 4A and 4B will be available at the time of the visit, as well as the minutes of the meetings where the assessment results were evaluated and where recommendations for action were made.

Criterion 5

Curriculum

5.A Program Curriculum

5.A.1 Plan of study of the Program

The plan of study of the Program is presented in Table 5.1 and in the additional Tables 5.2 and 5.3 (at the end of this chapter).

Terms are based on a semestral system. Each term, called semester, last approximately four months. There are normally two terms, Spring and Fall each year. Some courses may be also scheduled for a summer term of two months. These are made intensive in order to cover exactly the same material as their regular counterpart.

The unit that characterizes courses in the plan of study is the credit hour. According to the Academic Units Rules [7] one credit hour equals 48 hours of study in a semester of 16 weeks, with a distribution of two hours of independent study per hour of class work. Nominally, one credit hour corresponds to a course of one hour per week of class work. In Table 5.1, S13 and F13 denote Spring and Fall 2013 respectively, and S14 and F14 denote Spring and Fall 2014 respectively.

The total amount of credit hours required for graduation is 170 and, from Section B of the Background Information chapter, they are distributed into 134 credit hours of core-curriculum courses, 18 credit hours of concentration (9 credits of them is the degree project), 12 credit hours of the minor and 12 credit hours of elective courses. The minimum number of approved credit hours to start the minor and the concentration are 36 and 66, respectively. Also, and as indicated in section C of Background Information, students can take two concentrations and no minor. In Table 5.1 does not include detailed elective and minor courses, since these are selected from the whole university offering. Table 5.2 and Table 5.3 present the courses for the Animation and Interactive Systems, and Net-centric Computing concentrations respectively. The grouping of courses by subject areas is shown in Table 5.4.

5.A.2 Alignment with the program educational objectives

Recalling from Section 2.B, the Computer Science program objectives are:

1. Be able to solve problems in science and engineering by effectively applying computer science knowledge and techniques.
2. Use their abilities, ingenuity and analytical skills and formal reasoning from Computer Science to identify, analyze, design, implement, maintain and operate technological solutions to solve discipline- related problems or contribute to the solution of problems in other fields.
3. Be performing with excellence in promising positions in industry, academia, public service, or as entrepreneurs, from which to contribute to the welfare of their communities.
4. Be distinguished by their sense of ethics and responsible citizenship, well-educated professionalism, awareness of the need for sustainability, sound foundations in computing and teamwork and communication skills.
5. Be committed to the life-long quest of knowledge by pursuing graduate studies, self-study or professional development.

The curriculum has been aligned with the program educational objectives in the following ways:

The initial courses prepare our students on theoretical and fundamental concepts required to address engineering problems from a computer science perspective. Courses such as differential, integral and multivariable calculus, dynamics and kinematics, electricity and magnetism, and linear algebra provides a strong mathematical foundation to understand and propose solutions to engineering problems. Introduction to programming, introduction to system modeling, programming laboratory, data structures and programming fundamentals and object oriented and mid-scale programming provides the tools and formalism to solve problems applying computer science. All of these courses and many others contribute to the achievement of the first objective.

The second objective is addressed in courses such as probability and statistics, computability and formal languages, management and data modeling, numerical analysis, computer networks, computer graphics, formal software development, large-scale software development, analysis and design algorithms, operating systems and artificial intelligence, among others. These courses provided more deeper knowledge in computer science topics on how to effectively apply that knowledge in problems solutions in different fields.

Some courses have been designed to foster student performance in different professional environments. Oral and written expression, politics and Colombian democracy, humanities I and II, organization management, economic engineering, fundamental of research, professional and social aspects, degree project and professional internship, are courses oriented to address the third objective. These courses present different professional aspects: research, social, community and industry. The concentration and minor courses help the students on focus areas based on their professional interests.

Objective four is addressed in some of the previous referenced courses, but mainly in courses such as theology I and II, humanities I and II, politics and Colombian democracy,

social and professional aspects, ethics, oral and written expression, professional internship, and degree project.

In general, the courses provide knowledge and help to improve the student's skills for their professional development. However fundamental research, degree project and professional internship are courses that are oriented to promote pursuing graduate studies, self-study and professional development.

A more detailed alignment between the curriculum and the program educational objectives is presented in Table 5.5. The table shows the results from the alignment of the PEOs with the student outcomes (see Table 3.1, Criterion 3), and of those with the curriculum (see Table 4.1, Criterion 4). Thus, to each relationship of a course with a student outcome in Table 4.1 it is possible to assign the PEOs related to that student outcome. In general, each program educational objective is generously supported through the curriculum. Furthermore, through the relationships of the PEOs with the institutional mission (see Table 2.1, Criterion 2), it is possible to ascertain that the mission is fully supported by the curriculum.

Beyond the systematic result of the above paragraph, the fact that each course supports all of the program educational objectives is very important qualitatively because it indicates a balanced and robust curriculum in which all courses are equally important for the purposes of the program. Thus, every course is concerned with the application of knowledge, the pursuit of excellence in serving society, and the promotion of responsible citizenship and independent learning.

5.A.3 Attainment of student outcomes

The relationships of the student outcomes with the curriculum were presented in Table 4.1 of section 4.A.3, with an explanation as to the method used to establish those relationships.

The Computer Science program curriculum starts with foundations on mathematics (Fundamental of Mathematics, Differential Calculus, Integral Calculus, Multivariable calculus, Linear Algebra, Discrete Mathematics, Logic for CS) and physics (Dynamics and Kinematics, Electricity and Magnetism). These courses provide the mathematical fundamental knowledge to support advance courses with mathematical foundation such as Numerical Analysis and Computer Graphics, among others. At the same time, on the fundamentals in computer science are supported for courses such as, Introduction to Computer Science, Data structures and Programming fundamentals, Object Oriented and mid-scale programming, and Computer Architecture I. These courses allow students to obtain the necessary computing and mathematical knowledge, and analysis skills required to continue with deeper and topic specific courses in computer science.

After the foundations courses, students can pursue courses requiring more analysis and decision making. Courses such as Probability and Statistics, Computability and Formal Languages, Numerical Analysis, Analysis and Design of Algorithms, complements the foundation with more tools and skill to approach problems in computer science and engineering, in general. There are courses focused on improve design and implementation skills, starting from Introduction to System Modeling and Object Oriented and mid-scale programming and later complemented with Management and Data Modeling, Computer Architecture II, Software

Engineering Processes, Database implementation, Operating Systems. These course give the students the ability to apply concepts and theories in computer science to model and design systems and solutions, through concepts, theories and tools, working in a collaborative teamwork environment.

The curriculum provides a set of courses to discuss and approach different software system complexities, from a design and development perspective. Students starts with Object Oriented and middle scale programming, followed for Management and Data Modeling, Software Engineering and continuing with Large-scale Software Development. After these courses, the students are able to model, design and construct software systems, that are common to find in the industry, and how these systems affect the society.

In most courses, working in groups of 2 or three students is used as pedagogical method. This pedagogy allows the student to develop skills of leadership, responsibility and teamwork. Usually, final course projects are developed in teams. Starting from Introduction to Computer Science, the students are in contact with professional, ethical, legal and social issues related to computer science. These initial knowledge is later expanded in courses such as Humanities I and II, Ethic and Professional and Social Aspects.

The Oral and Written expression course is the first course where students are exposed to the improvement in communication skills. These skills are continually exercised during the semesters on different courses. One of those courses is Undergraduate Project, where the students are required to produce a document and oral presentation showing their research process and results. These skills are also tested in the professional internship, where the students need to interact within a professional environment. In these two courses, Undergraduate Project and Professional Internship, plus fundamentals of research, the student has the opportunity to practice and show the knowledge gained during the earlier semesters, in more complex and real problems.

5.A.4 Flowchart of the courses and prerequisite structure

Figure 5.1 shows the plan of study and the prerequisite structure of the core curriculum and concentration courses. As indicated in section 5.A.1, the minor and elective courses are not depicted on this flowchart because the precise location of their courses entirely depends on the choices of students.

In Figure 5.1, courses are identified by their name, catalog code, credit hours and the audience they are offered to: open to all students or closed to engineering students. Also, in Figure 5.1 there are additional instructions regarding: the maximum amount of 18 credits to register in each semester, the maximum number of credit hours in the concentrations, minor and electives and the minimum amount of approved credit hours to start the concentration and the minor.

5.A.5 Meeting of general and program criteria

From Table 5.1, the total number of credit hours in the subject areas corresponds to the following percentages (relative to a total of 170 credit hours): 21.2 percent in the mathematics and basic sciences component; 44.1 percent for Computing related topics; 11.7 percent for the general education component and 26.5 percent for other areas. The percentage for electives and concentration courses is 14.2 percent distributed equally (7.1 percent electives, 7.1 percent concentration). However, it must be highlighted that this is the result of the nominal offerings of the curriculum but that the actual balance of credit hours students get is slightly different due to how they administer the minor and concentrations (see section 5.A.1).

The curriculum provides two and a half years of work in science and mathematics. Students take courses in mathematics such as Differential calculus, Integral calculus, Multivariate calculus, Linear Algebra, Numerical Analysis, Probability and Statistics, Discrete Mathematics and Logic for Computer Science. On topics related to science the student take courses such as Introduction to Computer Science, Dynamics and Kinematics, Electricity and Magnetism and Programming Laboratory. These courses involve the student in laboratory work where they explore how to apply the scientific method and engineering design to understand the different studied phenomena.

In topics related directly to computer science, the curriculum includes courses to cover fundamentals concepts on algorithms, data structures, software design, programming languages and computer organization and architecture. Courses such as Introduction to Programming, Programming Fundamentals and Data Structures, Object Oriented and Mid-scale programming, Computer Architecture I, Computability and Formal Languages, and Computer Networks are some of the courses that cover these topics. More advance topics in computer science come after the fundamentals and introductory courses are taken. Courses such as Software Engineering Processes, Computer Architecture II, Formal Software Development, Large-scale Software Development, Analysis and Design of Algorithms, Computer Graphics and Operating Systems, among others, require fundamental knowledge to discuss the various topics treated in them. In general, more fundamental and advance topics in computer science are covered in approximately two and a half years. In this period, the students face different programming languages, starting from Python to learn the basics of programming, moving to C and C++ to understand more deeply how a program works in a machine and how to structure data and programs, to finally cover languages more industry oriented such java and web oriented languages.

The earlier mentioned times meet the general criteria about minimum required time for science, mathematics and computer science, and the topics covered on this time.

5.A.6 Cooperative education

The major cooperative education activity of the Program is the Professional Internship course. This is a course of the tenth semester of the plan of study and bears the important value of 6 credit hours. In this course students work as interns in industry and organizations

for a minimum of 20 weeks and have to undertake tasks and projects that are commensurate with the professional practice of Computer Science but that, also, bear a clear academic challenge. Thus, the purpose of the course is to make students familiar with standard working environments in their discipline in which they are expected to apply the knowledge, abilities and skills proper of their curriculum.

Each student is assigned two supervisors; one is a faculty member, and the other is the person who oversees the students work in the organization. The two supervisors jointly evaluate the course according to the curricular formula of the course (see Table 4.1) with relative weights of 60 percent for the faculty member and 40 percent for the industry supervisor. The role of the faculty member is to ensure that the tasks assigned to the intern bear an academic value appropriate to the curricular formula of the course. The importance of this course is expressed in the curricular formula evaluating the 11 student outcomes, placing the highest relevance on teamwork abilities (D), ethical and professional responsibility (E), communication (F), analyze computing impact (G), continue professional development (H), and ability to apply fundamental concepts to model and design computer-based systems (J).

In addition to the above, the Professional Internship is very important for students as it is a unique opportunity for them to get their first employment. Normally, students are hired as interns with contracts for the duration of the course and some are offered the renewal of their contracts in those organizations.

Apart from the Professional Internship, the Program offers two additional cooperative education opportunities: the ME310 course with Stanford University and an agreement with the Turin Polytechnic Institute.

The ME310 course is offered by the Department of Mechanical Engineering of Stanford University to students of international universities. The course is based on design problems posed by corporations and organizations. The duration of the course is about nine months (September to June) and teams of students are formed from the participating universities. Each team is assigned a design project. The programs of the School of Engineering have participated in this course since 2007. As the course requires a dedication of time equivalent to 18 credit hours, the course is valid for the Degree Project, the Professional Internship and two electives.

With the Turin Polytechnic Institute we have an agreement whereby students from Javeriana in their fifth year go to Turin to start the Masters of Engineering program, with duration of two academic years. On completion of that program, the students are awarded the corresponding Bachelor of Science degree at Javeriana.

5.A.7 Material available for review

The materials prepared for the ABET visit include course syllabi (cic.javerianacali.edu.co/wiki/), textbooks, sample student work, and other artifacts that will demonstrate achievement related to this criterion.

5.B Course Syllabi

In Appendix 8.E we present the syllabi of the core-curriculum, including mathematics and basic sciences, and concentration courses.

Table 5.1: Curriculum of Computer Science Program

Course (Department, Number, Title) List all courses in the program by term starting with first term of the first year and ending with the last term of the final year.	Indicate whether course is Required, Elective or a Selected Elective by an R, an E or an SE	Subject Area (Credit Hours)				Last Two Terms the Course was Offered: Year and, Semester, or Quarter	Average Section Enrollment for the Last Two Terms the Course was Offered
		Math & Science	Computing Topics Mark with an F or A for Funda- mental or Advanced	General Education	Other		
Semester 1							
CNM, 300MAG018, Fundamentals of Mathematics	R	3				S14, F14	13
CJP, 300CSP003, Politics and Colombian democracy	R			2		S14, F14	N/A
DECC, 300CIG001, Introduction to Computer Science	R		3(F)			S14, F14	23
DECC, 300CIP001, Introduction to Programming	R		3(F)			S14, F14	25
COM, 300EPG002, Oral and Written expression	R			2		S14, F14	N/A
HUM, 300EIH001, Humanities I	R			2		S14, F14	N/A
DECC, 300IGO001, Introduction to System Modeling	R		2(F)			S14, F14	26
Semester 2							
CNM, 300MAG002, Differential Calculus	R	3				S14, F14	2
CNM, 300MAG006, Linear Algebra	R	3				S14, F14	2
DECC, 300CIP002, Prog. Fund. and Data structures	R		4(F)			S14, F14	13
DECC, 300CIP007, Laboratory of Programming	R		2(F)			S14, F14	14
HUM, 300EIH002, Humanities II	R			2		S14, F14	N/A
COM, 300LEI002, English II	R			2		S14, F14	N/A
Semester 3							
CNM, 300MAG007, Integral Calculus	R	3				S14, F14	2
CNM, 300MAG031, Discrete Mathematics	R	4				S14, F14	9
DECC, 300CIS004, Object Oriented and Mid-scale Prog.	R		3(F)			S14, F14	10
HUM, 300TEG001, Theology I	R			2		S14, F14	N/A
COM, 300LEI003, English III	R			2		S14, F14	N/A
Semester 4							
CNM, 300MAG008, Multivariable Calculus	R	3				S14, F14	3
CNM, 300FIF003, Dynamics and Kinematics	R	4				S14, F14	2
DECC, 300CIG005, Computer Architecture I	R		3(F)			S14, F14	9
DECC, 300CIG002, Logic for Computer Science	R	3				S14, F14	9
GO, 300ANG007, Organizational Management	R				3	S14, F14	5
COM, 300LEI004, English IV	R			2		S14, F14	N/A
Semester 5							
CNM, 300MAE005, Probability and Statistics	R	3				S14, F14	4
CNM, 300FIF002, Electricity and Magnetism	R	4				S14, F14	3
DECC, 300CIG006, Computer Architecture II	R		3(A)			S14, F14	9
DECC, 300CIG007, Computability and Formal Languages	R		3(A)			S14, F14	17
DECC, 300CID001, Inf. Management and Data Modeling	R		3(A)			S14, F14	9
HUM, 300TEG002, Theology II	R			2		S14, F14	N/A
Semester 6							

CNM, 300MAG015, Numerical Analysis	R	3				S14, F14	10
DECC, 300CIS006, Networks and Communications	R		3(A)			S14, F14	8
DECC, 300CIS005, Software Engineering Processes	R		3(A)			S14, F14	13
DECC, 300CIG008, Computer Graphics	R		3(A)			S14, F14	6
Semester 7							
DECC, 300CIS002, Formal Software Development	R		3(A)			S14, F14	14
DECC, 300CIS001, Large-scale Software Development	R		3(A)			S14, F14	6
DECC, 300CIG004, Analysis and Design of Algorithms	R		3(A)			S14, F14	8
DECC, 300CID002, Databases Implementation	R		2(A)			S14, F14	11
DECC, Concentration I	SE						
Semester 8							
DECC, 300CIG002, Fundamentals of Research	R				3	S14, F14	8
DECC, 300CIS003, Web Services Development	R		3(A)			S14, F14	7
DECC, 300CIG011, Operating Systems	R		3(A)			S13, S14	12
DECC, 300CIG009, Artificial Intelligence	R		3(A)			S14, F14	11
DECC, Concentration II	SE						
Semester 9							
DECC, 300IGG004, Undergraduate Project	R				6	S14, F14	6
HUM, 300FRG001, Ethic	R			2		S14, F14	N/A
DECC, Concentration III	SE						
Semester 10							
DECC, 300CIG037, Professional Internship	R				6	S14, F14	7
DECC, 300CIG010, Professional and Social Aspects	R		2(A)			S14, F14	8
DICI, 300CIG003, Engineering Economics	R				3	S14, F14	6
DECC, Concentration IV	SE						
Minor and Elective Courses							
Minor courses	SE				12		
Elective courses	E		6		6		
Totals-ABET Basic-Level requirements		36	75	20	39		
Overall total credits hours for completion the program : 170							

Acronyms of names of departments.

DICI: Department of Civil and Industrial Engineering	COM: Department of Communication and Language
DECC: Department of Electronics and Computer Science	CJP: Department of Legal and Political Science
CNM: Department of Natural Science and Mathematics	GO: Department of Organization Management
HUM: Department of Humanities	

Table 5.2: Curriculum Concentration: Animation and Interactive Systems Computer Science Program

Course (Department, Number, Title) List all courses in the program by term starting with first term of the first year and ending with the last term of the final year.	Indicate whether course is Required, Elective or a Selected Elective by an R, an E or an SE	Subject Area (Credit Hours)				Last Two Terms the Course was Offered: Year and, Semester, or Quarter	Average Section Enrollment for the Last Two Terms the Course was Offered
		Math & Science	Computing Topics Mark with an F or A for Funda- mental or Advanced	General Education	Other		
Concentration: Animation and Interactive Systems							
DECC, 300CIG032, Concentration I: Animation and Simulation	SE		3(A)			S14	9
DECC, 300CIG034, Concentration II: Interaction and Sound	SE		2(A)			F14	9
DECC, 300CIG033, Concentration III: Introduction to Video Games Development	SE		2(A)			S14	8
DECC, 300CIG035, Concentration IV: Artificial Intelligence for Video Games	SE		2(A)			F14	2
Totals-ABET Basic-Level requirements			9				
Overall total credits hours for completion the program : 170							

Table 5.3: Curriculum Concentration: Net-Centric Computing

Course (Department, Number, Title) List all courses in the program by term starting with first term of the first year and ending with the last term of the final year.	Indicate whether course is Required, Elective or a Selected Elective by an R, an E or an SE	Subject Area (Credit Hours)				Last Two Terms the Course was Offered: Year and, Semester, or Quarter	Average Section Enrollment for the Last Two Terms the Course was Offered
		Math & Science	Computing Topics Mark with an F or A for Funda- mental or Advanced	General Education	Other		
Concentration: Net-centric Computing							
DECC, 300CIS014, Concentration I, Multimedia technologies	SE		3(A)			F14	12
DECC, 300CIG030, Concentration II, Information Security	SE		3(A)			F14	8
DECC, 300CIG031, Concentration III, Mobile computing and Mobile agents	SE		3(A)			S14	11
Totals-ABET Basic-Level requirements			9				
Overall total credits hours for completion the program : 170							

Table 5.4: Subject areas and courses of the Computer Science Curriculum

Subject area	Courses
Mathematics and Basic Science (8 courses, 26 credit hours)	Fundamental Mathematics, Linear Algebra, Differential Calculus, Integral Calculus, Multivariate Calculus, Kinematics and Dynamics, Electricity and Magnetism, Probability and Statistics
General Education (10 courses, 20 credit hours)	Oral and Written Expression, Humanities I, Humanities II, Theology I, Theology 2, English I, English II, English III, Political Cons. and Democracy, Ethics
Basic Engineering (2 courses, 6 credit hours)	Organizational Management, Engineering Economy
Fundamentals of Programming (2 courses, 5 credit hours)	Introduction to Programming, Laboratory of Programming
Algorithms and Complexity (2 courses, 6 credit hours)	Analysis and Design of Algorithms, Computability and Formal Languages
Programming Languages (2 courses, 7 credit hours)	Programming Fundamentals and Data Structures, Object Oriented and Mid-scale Programming
Information Management (2 courses, 5 credit hours)	Database Implementation, Inf. Management and Data Modeling
Operating Systems (2 courses, 6 credit hours)	Operating Systems, Linux for Engineering (Elective)
Software Engineering (2 courses, 6 credit hours)	Software Engineering Processes, Large-Scale Software Development
Network-Centric Computing (3 courses, 9 credit hours)	Web-based Services, Information Security, Mobile Computing
Human-Computer Interaction (2 courses, 5 credit hours)	Interaction and Sound, Multimedia Technologies
Graphic and Visual Computing (3 courses, 8 credit hours)	Computer Graphics, Animation and Simulation, Introduction to Video Games Development
Intelligent Systems (2 courses, 5 credit hours)	Artificial Intelligence, Artificial Intelligence for Video Games
Computational Science (3 courses, 8 credit hours)	Introduction to Computer Science, Introduction to System Modeling, Formal Software Development, Numerical Analysis
Social and Professional Issues (1 courses, 2 credit hours)	Social and Professional Issues
Architecture and Organization (2 courses, 6 credit hours)	Computer Architecture I, Computer Architecture II
Data Communication (1 courses, 3 credit hours)	Networks and Communication,

Projects (3 courses, 15 credit hours)	Fundamentals of Research, Undergraduate Project, Professional Internship
--	---

Table 5.5: Relationship of courses with program educational objectives.

Course	Program Educational Objectives
Fundamental Mathematics	EO1, EO2, EO4, EO5
Linear Algebra	EO1, EO2, EO4, EO5
Differential Calculus	EO1, EO2, EO4, EO5
Integral Calculus	EO1, EO2, EO4, EO5
Multivariate Calculus	EO1, EO2, EO4, EO5
Kinematics and Dynamics	EO1, EO2, EO4, EO5
Electricity and Magnetism	EO1, EO2, EO4, EO5
Oral and Written Expression	EO3, EO4
Humanities I	E02, EO3, EO4
Humanities II	E02, EO3, EO4
Theology I	E02, EO3, EO4
Theology 2	E02, EO3, EO4
English I	EO3, EO4
English II	EO3, EO4
English III	EO3, EO4
Political Cons. and Democracy	E02, EO3, EO4
Ethics	E02, EO3, EO4
Organizational Management	E02, EO3, EO4
Analysis and Design of Algorithms	EO1, E02, EO3, EO4, EO5
Computer Architecture I	EO1, E02, EO3, EO4, EO5
Computer Architecture II	EO1, E02, EO3, EO4, EO5
Social and Professional Issues	E02, EO3, EO4, EO5
Computability and Formal Lang.	EO1, E02, EO3, EO4, EO5
Computer Graphics	EO1, E02, EO3, EO4, EO5
Large Scale Software Development	EO1, E02, EO3, EO4, EO5
Formal Software Development	EO1, E02, EO3, EO4, EO5
Web-based Services	EO1, E02, EO3, EO4, EO5
Program. Fundam.and Data Str.	EO1, E02, EO3, EO4, EO5
Inf. Management and Data Modeling	EO1, E02, EO3, EO4, EO5
Database Implementation	EO1, E02, EO3, EO4, EO5
Artificial Intelligence	EO1, E02, EO3, EO4, EO5
Introduction to Computer Science	EO1, E02, EO3, EO4, EO5
Introduction to Programming	EO1, E02, EO3, EO4, EO5
Introduction to System Modeling	EO1, E02, EO3, EO4
Laboratory of Programming	EO1, E02, EO3, EO5
Logic in Computer Science	EO1, E02, EO3, EO4, EO5
Obj. Oriented and Mid-scale Prog.	EO1, E02, EO3, EO4, EO5

Professional Internship	EO1, E02, EO3, EO4, EO5
Software Engineering Processes	EO1, E02, EO3, EO4, EO5
Networks and Communication	EO1, E02, EO3, EO4, EO5
Operating Systems	EO1, E02, EO3, EO4, EO5
Fundamentals of Research	EO1, E02, EO3, EO4, EO5
Undergraduate Project	EO1, E02, EO3, EO4, EO5
Animation and Simulation	EO1, E02, EO3, EO4, EO5
Interaction and Sound	EO1, E02, EO3, EO4, EO5
Intro. to Video Games Dev.	EO1, E02, EO3, EO4
Artificial Intelligence for Games	EO1, E02, EO3, EO4, EO5
Mobile Computing	EO1, E02, EO3, EO4, EO5
Information Security	EO1, E02, EO3, EO4, EO5
Multimedia Technologies	EO1, E02, EO3, EO4, EO5
Numerical Analysis	EO1, E02, EO3, EO4, EO5
Engineering Economy	EO1, E02, EO3, EO4, EO5
Probability and Statistics	EO1, E02, EO4, EO5
Discrete Mathematics	EO1, E02, EO4, EO5

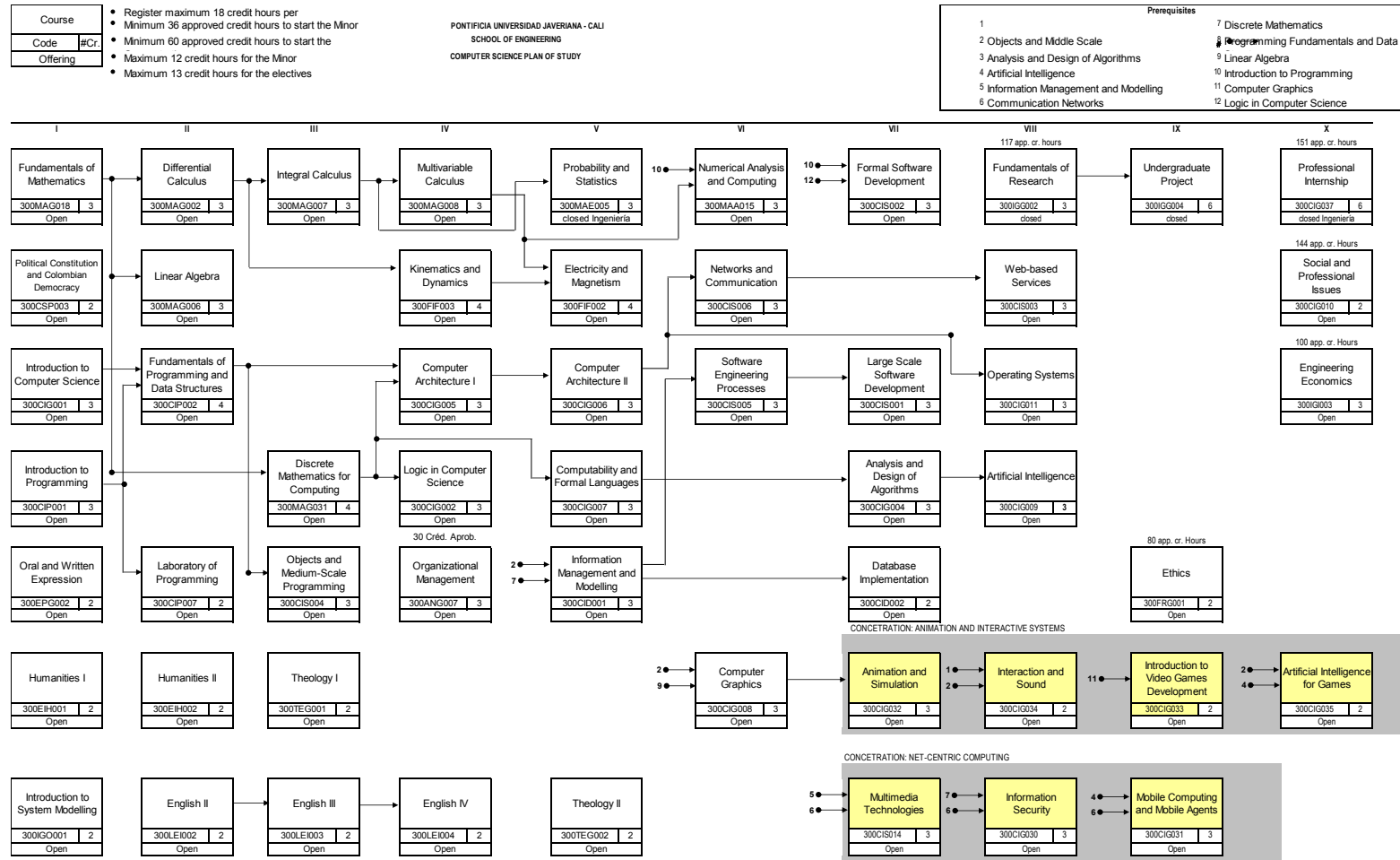


Figure 5.1: Prerequisite structure for Computer Science program curriculum

Criterion 6

Faculty

6.A Faculty Qualifications

The Academic Units Rules [7] define the academic departments as units composed of faculty members that carry out teaching, research and service (continued education and consultancy). Thus, the Department of Electronics and Computer Science is in charge of providing those services for the support of the Computer Science Program and the Electronic Engineering Program and of the Postgraduate Programs of the School of Engineering.

The Department comprises 24 full-time faculty positions associated with the two areas that support the programs indicated above: the Computer Science area with 10 full-time faculty positions and the Electronic Engineering area with 14 full-time faculty positions. In addition, the Department hires 14 part-time academics to support courses of the Computer Science and Electronic Engineering programs

Table 6.1 presents the credentials, ranks, time of service, professional registration and external experience of the 10 full-time faculty members of the Computer Science area. These faculty members support the core-curriculum and concentration discipline-related courses that are mainly associated with the computing-topics courses of Table 5.1. Thus, the following analyses as to the qualifications, workload, size and adequacy of the support given to the Program are made only upon those faculty members.

From Table 6.1, the average time of experience of the full-time faculty members is 11 years, with five faculty members reaching more than 15 or more years of experience in the Program. With regards to the bachelor degrees, all of the full-time faculty members hold degrees in Computer Science or Systems Engineering (which is the common denomination in Colombia). Regarding the highest earned degrees, five are PhDs, three are Masters and two still hold bachelor degrees. As of June 2014, two full-time faculty members are completing their PhD studies; therefore, in the short term the Computer Science area would have 70 percent of full-time faculty members with PhD degrees. This is remarkable in the context of engineering schools in Colombia. In terms of ranks, one is a Professor, four are Associate Professors, three are Assistant Professors and two are Instructors. Regarding the origin of the doctorate degrees, one was earned in the United States (Massachusetts Institute of

Technology), one in France (Ecole Polytechnique Paris), one in Spain (Universidad Politécnica de Valencia), one in New Zealand (Otago University) and one in Colombia (Universidad del Valle, Cali).

Therefore, considering the time of experience, the academic areas of the degrees, and their origin the overall conclusion is that faculty qualifications are adequate for the attainment of the program educational objectives and student outcomes of the Computer Science Program.

6.B Faculty Workload

Table 6.2 presents the allocation of workload of the 10 full-time and 10 part-time faculty members of the Computer Science area. A full-time position corresponds to a weekly dedication of 40 hours and this is expected to be distributed as follows: between 13 and 30 percent to teaching; between 7 and 60 percent to preparation of classes and evaluation; between 7 and 15 percent to attention of students and the remainder can be dedicated to other activities related to teaching, research, service (continued education, consultancy), professional development, academic administration and institutional initiatives. In the Department the nominal workload is six courses through the year for faculty members who do not participate in research, and for those who carry out research, the yearly workload is four courses. In the Computer Science area the average percentage workload of full-time faculty members over the semesters 2013-2 and 2014-1 was distributed as follows: 14 to teaching; 19 to preparation of classes and evaluation; 7 to attention of students; 9 to supervision of degree projects and interns; 1 to academic counseling; 19 to research; 2 to service; 10 to professional development; 13 to administration and 6 to institutional initiatives.

At this point it is important to highlight that six full-time faculty members (see Table 6.2) report time dedicated to research. Research in the Computer Science area is articulated by the AVISPA and DESTINO research groups. The research interest of the AVISPA group is the development of formal methods for computational modeling, in particular the modeling of concurrent processes. On the other hand, the research interests of the DESTINO group are: pattern recognition, graphical computing, human-computer interaction and software engineering.

In summary, the full-time faculty members of the Computer Science area dedicate 49 percent of the week to activities related with teaching out of which 40 percent involves close contact with students. Thus, this allocation of workload meets in excess the abovementioned minimum expectations. On the other hand, the dedication to academic counseling and service are very low. Considering that the University seeks to promote research and service, the current allocation of workload to teaching should not be expected to increase significantly with the available resources.

All in all, the current situation of workload distribution is considered to be adequate for the attainment of the program educational objectives and student outcomes because students have opportunities to benefit from close contact with full-time faculty members in a variety of activities including classes, workshops, laboratories, supervision and counseling.

6.C Faculty Size

The Computer Science area supports, on average over the 2013-2 and 2014-1 semesters, 28 sections per semester, out of which the full-time faculty members support 16, i.e. 57 percent. By comparison (see section A), the current proportion of full-time faculty members in the Computer Science area is 50 percent. On the other hand, Table 6.3 presents the allocation of faculty members to the core-curriculum discipline-related and concentration courses of the Computer Science curriculum. Thus, full-time faculty members are allocated to 24 out of 29 courses in 13 out of the 14 subject areas of the Computer Science Program. Thence, the allocation of faculty members to sections and courses far exceeds the proportion of full-time faculty members, and this is possible because the courses of the Computer Science Program are single-section courses and five full-time faculty members were allocated to more than one course. On the other hand, from Table 6.3, only six core-curriculum courses were assigned to part-time faculty members. It must be considered as well that this area contributes two full-time faculty members to the academic administration: the Director of the Computer Science Program (Antal Alexander Buss) and the Head of the Electronics and Computer Science Department, DECC, (Dr. Gloria Inés Álvarez), both with reduced teaching workload. Also, and as said in section A, the allocation of time to research is important in this area. All in all, the conclusion is that the faculty size is adequate to cover the subject areas of the curriculum and support the attainment of student outcomes.

It must be noticed at this point that Table 6.3 does not include discipline-related courses of the Computer Science curriculum that are offered from areas and departments other than the Computer Science area. That is the situation of the courses Discrete Mathematics for Computing (from the Natural Sciences and Mathematics department); the Engineering Economics course (from the Industrial Engineering area of the Civil and Industrial Engineering department); and the Organizational Management course (from the Organizational Management department). The course Logic in Computer Science was assigned to a full-time faculty member from the Mathematics area of the Natural Sciences and Mathematics department because Dr. Carlos Alberto Olarte was in postdoctoral leave in the semesters 2013-2 and 2014-1.

The Professional Internship course is assigned to one full-time faculty member as shown in Table 6.2, who supervises, on average, 15 – 20 students per semester. The methodology of the course requires that supervisors visit the organizations where the interns are based at least two or three times during the semester, thus the course is very demanding on faculty time.

Another activity that demands faculty time is academic counseling (see section D, Criterion 1). Currently the program director is appointed as counselors.

The student group of the Computer Science Program, the Student ACM Chapter, is advised on a permanent basis by one full-time faculty member along the semester, but other faculty members might get involved if needed.

At the moment of preparation of this report (June 2014), only one full-time faculty member is involved in a consultancy project and another is engaged in continued education. This illustrates the very low engagement of faculty members of the Industrial Engineering

area in service activities.

6.D Professional Development

A variety of programs and initiatives are available for faculty members. Those programs are mainly offered and administered by the Office of the Academic Vice-rector and are intended to promote and strengthen abilities related to teaching, research and service. A brief description of those programs is the following:

- Program of academic development: its purpose is to provide, total or partial, financial support for faculty members to pursue postgraduate studies either in Colombia or abroad. As of June 2014, five full-time faculty members of the Computer Science area have benefited from this program for PhD studies; three of them with completed studies and another two faculty members are next to finishing.
- Program of competencies development: it is intended to strengthen the development of competencies related to teaching, research, innovation, transfer of knowledge, service and academic administration. It operates through seminars and diploma courses.
- Institutional training: its purpose is to train faculty members and staff in institutional issues such as statutes, rules and procedures and to promote the institutional identity.
- Training in technologies of information and communication (TICs): it offers training in TICs for faculty members to use adequately those resources in support of teaching to enhance learning and also in support of research and service.
- Program of second language: it offers second-language courses for faculty members and staff. Regarding English, the program offers the possibility for faculty members to have short stays at the University of San Francisco and Gonzaga University (Spokane).
- Counseling: it offers personalized counseling and advice on career planning, academic and professional development, academic and scientific writing, training as tutors/mentors and it also offers psychological and spiritual advice.

The Office of the Academic Vice-rector, through the Office of Professoral Affairs, organizes the annual schedule of those programs and their advertising. The promotion of activities, courses and seminars is made through the University webpage, through the individual email and via individualized promotion at the Departments. The activities of those programs are usually scheduled on the inter-semester periods of June – July and December - January. The funds to cover the costs of the individual participation on those programs are included in the annual budget of the Departments upon demands from faculty members.

6.E Authority and Responsibility of Faculty

Given the academic organization of the University, the programs administer the curricula and students whereas the departments are comprised of the faculty that support teaching, research and service. Thence, the essential function of the departments is to support the programs in the attainment of the program educational objectives and student outcomes.

The mechanism that allows faculty members participating in the academic administration of the programs is their membership of the Program Committees. Those Committees are advisory bodies to the Program Director. As of June 2014, the members of the Committee of the Computer Science Program are Antal Alexander Buss (Director of the Program), Prof. Camilo Rueda, Dr. Andrés Navarro, Dr. Gerardo Sarria, Juan Carlos Martínez, Maribel Sacanamboy (from the Computer Science area), Jorge Figueroa (from the Natural Sciences and Mathematics department) and the student Alejandro Cardona; upon need, Dr. Gloria Inés Álvarez, Head of the DECC department attends the meetings of the Committee. On the other hand, the Head of the Department is advised by a Department Committee, which includes the following faculty from the Computer Science area: Dr. Gloria Inés Álvarez, Prof. Camilo Rueda, Antal Buss, and María Constanza Pabón. Thus, purposely, there is a high level of redundancy in the involvement of faculty members in those two bodies to facilitate a fluid communication between these two areas of the academic administration of the Program. From the above, the role of the Program Director is to ensure the overall consistency of the program with the institutional mission and the attainment of the program educational objectives and student outcomes, whereas the role of the Head of the Department is, primarily, to ensure that courses are delivered consistently with the student outcomes and other resources are provided adequately, such as laboratory spaces, equipment, teaching assistants, library resources and faculty development.

As it was mentioned in sections B and D, Criterion 2, the faculty members that support the Computer Science Program are a constituency of the Program, and as such, the full-time faculty members of the Computer Science area participated in the definition of the program educational objectives. It was also mentioned in section B, Criterion 3 that the Program Committee consulted with a wide number of faculty members about the relationships between the program educational objectives and the student outcomes and of those with the curriculum. Those consultations were made at the start of the preparation of the assessment of the Program but it is expected that they will be included in the continuous revision of the program educational objectives (see section E, Criterion 2).

In addition to the above paragraph, faculty member also play an important role in the preparation for the national accreditation. The usual procedure is that the writing of the self-study document is distributed amongst faculty members, an editor is appointed and the whole document is subjected to a peer-review stage before it is studied and approved by the Program Committee.

The reforms of the curriculum are regulated by the Academic Units Rules [1] and directives issued by the Office of the Academic Vice-rector. Thus, proposals of curriculum reforms are presented by the Program Director to the Dean of the School. Those proposals must have been studied and approved by the Program Committee from propositions made by the

Committee members. Therefore, this is the core mechanism whereby faculty members participate in the development of the program because, it is assumed, it is the faculty members who intimately know the necessities and issues that affect the program. The proposals of curriculum reform must be supported by a document that gives a detailed account of necessities, justification and benefits for both the students and the quality of the program. By directive of the Academic Vice-rector, modifications to contents of courses only require the approval of the Program Committee but must be communicated to the Academic Vice-rector through the Dean of the School; the rest of curriculum reforms must be approved by the Academic Vice-rector. Depending on the extent of a proposed curriculum reform there are two distinct procedures: if the number of affected credit hours is up to 5 percent of the total amount of credit hours in the program, the procedure is called a “curriculum adjustment”; if that amount exceeds 5 percent, the procedure is a “curriculum reform”.

From the above, the role of the Dean of the School is to link the program with the Office of the Academic Vice-rector. If the process in question is a curriculum adjustment, that link is immediate; when the process is a curriculum reform, it is customary that the Dean of the School submits the proposal to the study and approval of the Council of the School to add an important message of endorsement for the proposed reform.

Table 6.1: Faculty qualifications - Computer Science Program

Faculty Name	Highest Degree Earned - Field and Year	Rank ¹	Type of Academic Appointment ² T, TT, NTT	FT or PT ³	Years of Experience			Professional Registration/ Certification	Level of Activity ⁴ H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summerwork in industry
Gloria Inés Álvarez	Doctor, Pattern Recognition and Artificial Intelligence, 2008.	ASC	TT	FT	3	20	15	17255-118865-CLD	L	H	L
Andrés Becerra	Bachelor of Science, Computer Science, 2004	I	NTT	FT	3	12	10	76255118487-VLL	L	L	L
Antal Alexander Buss	Master in Engineering, Computer Science, 2008.	AST	TT	FT	1	10	6	76208-118500-VLL	L	H	L
Juan Carlos Martínez	Master in Engineering, Computer Science, 2012.	AST	NTT	FT	17	7	6	252551-60278-CND	L	H	M
Andrés Navarro	Doctor of Philosophy, Graphical Computing, 2010.	ASC	TT	FT	6	20	18	76208-54387-VLL	M	H	L
Carlos Alberto Olarte	Doctor, Computer Science, 2009.	ASC	TT	FT	2	11	11	762081-18506-VLL	L	H	L
María Constanza Pabón	Master, Business Administration, 2000.	AST	NTT	FT	9	15	13	762085-1095-VLL	L	H	L
Luisa Fernanda Rincón	Bachelor of Science, Informatics, 2010.	I	NTT	FT	3	0.5	0.5	17105012348-CLD	L	L	L
Camilo Rueda	Doctor of Philosophy, Electrical Engineering, 1976.	P	T	FT	3	27	20	N/A	M	H	L
Gerardo Sarria	Doctor in Engineering, Computer Science, 2008.	ASC	TT	FT	3	12	10	760281-21241-VLL	M	H	M
Katherine Cancelado	Bachelor of Science, Informatics			PT	4	1	1	N/A	M	M	M
Pablo Miguel Grech	Master in Engineering, Computer Science, 1970			PT	0	50	25	N/A	L	L	L
Diego Fernando Loaiza	Bachelor of Science, Computer Science, 2010			PT	10	3	2	762082-09947-VLL	L	H	L
Carlos Andrés Olarte	Bachelor of Science, Electronic Engineering, 2000			PT	17	11	10	VL2065-1070	L	H	H
Diego Fernando Polo	Specialist, Management of Organizational Informatics, 2007			PT	12	11	11	762081-22166-VLL	L	H	H
Juan Camilo Rada	Bachelor of Science, Computer Science, 2013			PT	3	1	1	76208-272700-VLL	L	H	M
Juan Manuel Reyes	Master in Engineering, Computer Science, 2010			PT	7	5	2	762551-18401-VLL	L	H	L

Frank Valencia	Doctor in Computer Science, 2003			PT	18	18	2	N/A	L	H	L
Jairo Andrés Velasco	Bachelor of Science, Electronic Engineering, 2009			PT	7	5	2	VL20-673946	L	H	L
Jorge Hernn Victoria	Master in Engineering, Cinnema and Communication, 2013			PT	3	2	1	76208-270750-VLL	L	H	L
Alexander Yela	Master in Engineering, Computer Science, 2013.			PT	16	9	2	762551-45380-VLL	L	H	H

1. Code: P = Professor ASC = Associate Professor AST = Assistant Professor I = Instructor A = Adjunct O = Other
2. Code: T = Tenured TT = Tenure Track NTT = Non Tenure Track
3. Code: FT = Full-time PT = Part-time Appointment at the institution.
4. The level of activity (high, medium or low) should reflect an average over the year prior to the visit plus the two previous years.

Table 6.2: Faculty Workload Summary - Computer Science Program

Faculty Member (name)	PT or FT ¹	Classes Taught (Course No./Credit Hrs.) Term and Year ²	Program Activity Distribution ³			% of Time Devoted to the Program ⁵
			Teaching	Research or Scholarship	Other ⁴	
Gloria Inés Álvarez	FT	Computability and Formal Languages (300CIG007/3) 2014-1 Artificial Intelligence (300CIG009/3) 2014-2	33	38	29	25
Andrés Becerra	FT	Operating Systems (300CIG011/3) 2014-1, 2014-2 Object and Mid-Scale Programming (300CIS004/3) 2014-2	100	0	0	50
Antal Alexander Buss	FT	Objects and Med-Scale Programming (300CIS004/3) 2014-1 Introduction to Programming (300CIP001/3) 2014-2	16	8	76	25
Juan Carlos Martínez	FT	Software Engineering Processes (300CIS005/3) 2014-1, 2014-2 Professional Internship (300CIG037/6) 2014-1, 2014-2 Introduction to Programming (300CIP001/3) 2014-1, 2014-2	59	8	33	80
Andrés Navarro	FT	Computer Graphics (300CIG008/3) 2014-1, 2014-2 Animation and Simulation (300CIG032/3) 2014-1 Artificial Intelligence for Video Games (300CIG035/2) 2014-2	57	33	10	100
Carlos Alberto Olarte	FT	In postdoctoral leave, 2014-1, 2014-2	N/A	N/A	N/A	N/A
María Constanza Pabón	FT	Information Management and Modeling (300CID001/3) 2014-1, 2014-2	19	0	81	100
Luisa Fernanda Rincón	FT	Programming Laboratory (300CIP007/2) 2014-1, 2014-2 Large-scale Software Development (300CIS001/3) 2014-1, 2014-2 Web-based Services (300CIS003/3) 2014-1, 2014-2	100	0	0	100
Camilo Rueda	FT	Formal Methods (300CIS002/3) 2014-1, 2014-2 Introduction to Systems Modeling (300IGO001/2) 2014-1, 2014-2	72	24	4	100
Gerardo Sarria	FT	Programming Fundamentals and Data Structures (300CIP002/ 4) 2014-1, 2014-2 Artificial Intelligence (300CIG009/3) 2014-1 Interaction and Sound (300CIG034/2) 2014-2	60	14	26	100
Katherine Cancelado	PT	Information Security (300CIG030/3) 2014-2	100	0	0	100
Pablo Miguel Grech	PT	Introduction to Computer Science (300CIG001/3) 2014-1, 2014-2	100	0	0	100
Diego Fernando Loaiza	PT	Introduction to Video Games Development (300CIG033/2) 2014-1	100	0	0	100

Carlos Andrés Olarte	PT	Networks and Communication (300CIS006/3) 2014-1, 2014-2	100	0	0	100
Diego Fernando Polo	PT	Social and Professional Issues (300CIG010/2) 2014-1, 2014-2	100	0	0	100
Juan Camilo Rada	PT	Mobile Computing and Mobile Agents (300CIG031/3) 2014-1	100	0	0	100
Juan Manuel Reyes	PT	Analysis and Design of Algorithms (300CIG004/3) 2014-1, 2014-2	100	0	0	100
Frank Valencia	PT	Computability and Formal Languages (300CIG007/3) 2014-2	100	0	0	100
Jairo Andrés Velasco	PT	Computer Architecture I (300CIG005/3) 2014-1, 2014-2	100	0	0	100
Jorge Hernn Victoria	PT	Multimedia Technologies (300CIS014/3) 2014-2	100	0	0	100
Alexander Yela	PT	Database Implementation (300CID002/2) 2014-1, 2014-2	100	0	0	100

1. FT = Full Time Faculty or PT = Part Time Faculty, at the institution
2. For the academic year for which the Self-Study is being prepared.
3. Program activity distribution should be in percent of effort in the program and should total 100%.
4. Indicate sabbatical leave, etc., under "Other".
5. Out of the total time employed at the institution.

Table 6.3: Allocation of faculty members to courses in the Computer Science Area

Subject area	Courses	Full-time faculty members	Part-time faculty members
Fundamentals of Programming	Introduction to Programming	Gerardo Sarria, Juan Carlos Martínez, Antal Buss	
	Laboratory of Programming	Luisa Fernanda Rincón	
Algorithms and Complexity	Analysis and Design of Algorithms		Juan Manuel Reyes
	Computability and Formal Languages	Gloria Inés Álvarez	Frank Valencia
Programming Languages	Object-Oriented and Mid-Scale Programming	Antal Buss, Andres Becerra	
	Programming Fundamentals and Data Structures	Gerardo Sarria	
Information Management	Database Implementation		Alexander Yela
	Information Management and Modeling	María Constanza Pabón	
Operating Systems	Operating Systems	Andrés Becerra	
Software Engineering	Software Engineering Processes	Juan Carlos Martínez	
	Large-Scale Software Development	Luisa Fernanda Rincón	
Network-centered Computing	Web-Based Services	Luisa Fernanda Rincón	
	Information Security		Katherine Cancelado
	Mobile Computing and Mobile Agents		Juan Camilo Rada
Human-Computer Interaction	Interaction and Sound	Gerardo Sarria	
	Multimedia Technologies		Jorge Hernn Victoria
Graphic and Visual Computing	Computer Graphics	Andrés Navarro	
	Animation and Simulation	Andrés Navarro	
	Introduction to Video Games Development		Diego Fernando Loaiza
Intelligent Systems	Artificial Intelligence	Gerardo Sarria	
	Artificial Intelligence for Games	Andrés Navarro	
Computational Science	Formal Software Development	Camilo Rueda	
	Introduction to Computer Science		Pablo Grech
	Introduction to System Modeling	Camilo Rueda	
Social and Professional Issues	Social and Professional Issues		Diego Fernando Polo
Architecture and Organization	Computer Architecture I		Jairo Andrés Velasco
	Computer Architecture II	Maribel Sacanamboy (Electronic Engineering area)	
Data Communication	Networks and Communication		Carlos Andrés Olarte

Criterion 7

Facilities

7.A Offices, Classrooms and Laboratories

7.A.1 Offices

The building of the School of Engineering houses offices and spaces for the entire faculty associated with the three departments of the School and for all the administrative and clerical staff. Out of a total area of 1807 m^2 , 979 m^2 are dedicated to faculty and administrative offices. Those offices have good illumination and air-conditioning, have a web-connected personal computer and a telephone extension, are fully furnished (desk, desk chair, two auxiliary chairs, and book shelves), and ensure a private, safe and quiet work environment. In addition, there are six meeting rooms with video-beam projectors, a common room with a kitchenette, an archive room and two cellars, eight restrooms, and six secretary stations.

Full-time faculty members have offices with an area of 9 m^2 , which ensure good privacy for the attention and counseling of students. Part-time faculty members of the Civil and Industrial Engineering and the Electronics and Computer Science departments have a room with 10 personal computers. For the administrative staff, the areas of offices are: 32 m^2 for the Dean; 14 m^2 for program directors, the secretary of the School and the coordinator of professional internships; 12 m^2 for department heads, and 9 m^2 for the assistants to the dean and to the secretary of the school.

7.A.2 Classrooms

For students of all the academic programs of the University, 98 spaces are available for academic activities such as lessons, lectures, examinations, workshops, and teamwork; out of which 68 are classrooms, 24 are auditoria, and 6 are projections rooms. All of those facilities are fully furnished, air-conditioned, well illuminated and provided with desk computers and video-beam projectors. The total area dedicated to those facilities is 7199 m^2 .

The capacities of the above facilities range between 24 and 40 students in classrooms, between 30 and 200 students in auditoria, and between 24 and 60 students in projection

rooms. Course sections are assigned to any of those facilities, according to size and methodologies. Checking with Table 5.1 of Criterion 5 shows that the capacities of those facilities are enough to accommodate course sections of the Industrial Engineering curriculum.

Those facilities provide to all occupants comfortable, safe and quiet work conditions and allow the use of a variety of audio and visual resources to support course methodologies.

7.A.3 Laboratories and computer rooms

The total area dedicated to the laboratories of the School of Engineering is 1902 m^2 , distributed into 272 m^2 for the physics and chemistry laboratories and the rest, i.e. 1630 m^2 , for the program-specific laboratories. On the other hand, the total area of program-specific computer rooms is 318 m^2 .

For the Computer Science program, there are three laboratories with a total area of 198 m^2 . Including the two 70 m^2 physics laboratories, a total area of 338 m^2 is available for Computer Science students. The following is the description and dedication of the three program-specific laboratories.

Computer Science Laboratory: this is an 112 m^2 laboratory with 31 computers and 17 tablets and is used to support a wide number of courses: Introduction to System Modeling, Laboratory of Programming, Fundamentals of Programming and Data Structures, Computer Architecture I, Computability and Formal Languages, Information Management and Modeling, Computer Graphics, Large Scale Software Development, Database Implementation, Formal Software Development, Web-based Services, Operating Systems, Artificial Intelligence, Introduction to Video Games Development, Artificial Intelligence for Games, Information Security.

Networks Laboratory: this is a 56 m^2 laboratory with 25 computers and is used to support the Networks and Communication course.

Interaction and Sound Laboratory: this is a 30 m^2 laboratory with seven computers and is used to support the following courses: Animation and Simulation, Interaction and Sound, Multimedia Technologies, and Mobile Computing and Mobile Agents.

The detailed specifications of the laboratory equipment and software licenses are presented in Appendix 8.E.

7.B Computing Resources

This section describes the computing resources that support services available to all students, faculty and staff of the University. These resources and services are administered and maintained by the Center for Computer Services (CSI).

7.B.1 Hardware platform

The main servers are six ProLiant DL380 G7 servers and eight ProLiant DL380 G8 servers. The G7 servers operate with processors Intel Xeon of 24 cores at 2.9 GHz with the operating

system ESXi 5.0 and support 230 virtual servers; whereas the G8 servers operate with Intel Xeon processors of 32 cores at 2.9 GHZ with the Citrix operating system and support 460+ virtual servers. About 85 percent of the virtual servers operate with Windows Server operating systems, and the rest operate with Linux. For backup, about 600+ LTO5 tapes are held in custody.

The total number of computers for general use is 2100, out of which 1056 are for use by students, and 1044 are for use by faculty and administrative staff. The ratios of computers to students and staff are 1:6 and 1:1, respectively.

The computers for use by students are distributed into 441 in 12 general-purpose computer rooms and 615 in program-specific laboratories. The general-purpose computer rooms comprise a total area of 1277 m^2 and service is available from 7:00 to 21:30 Monday to Saturday. Classes in those computer rooms are supported by the application Net Support School. A variety of hardware specifications is available: the processors include Core 2 Duo, Dual Core, Core i3, Core i5, Core i7, iMac, Quad-Core Intel Xeon, and Intel Core 2 Duo; processing speeds from 2.0 to 3.4 GHz; 2-8 Gb in RAM; and hard disks of 40 Gb to 1 Tb. Four laser printers and two scanners support the 12 computer rooms. Computer rooms are serviced by four technicians. In general, computers for faculty and administrative staff have hardware configurations very much the same as computer-room computers.

7.B.2 Software platform

The basic database search engine is Oracle, in versions 10g, 11g, and 9i, being supported by two of the ProLiant DL380 G7 servers described above. In addition, the search engine Microsoft SQL Server was added to support the Blackboard and Sharepoint applications.

General-purpose computers for students and those for faculty and administrative staff are loaded with Windows XP Professional, Windows 7, Windows 8, Linux or Mac OS.

Licenses are renewed and supported permanently. Those include Microsoft, Oracle, Corel, Adobe, AutoDesk, to mention but a few. The number of licenses is:

- Unlimited for Windows 7 Professional, Office 2010 Professional, Internet Explorer 9.0, Microsoft Project 2010, Microsoft Visio 2010, and Acrobat Reader 10.
- 1800 for Nod32 Antivirus.
- 50 for IBM SPSS 20.0 Basic Module.
- 3 for each of IBM SPSS Tables, IBM SPSS Regression, IBM SPSS Categories, and IBM SPSS Advanced Models.

In addition, open access is available for Mozilla Firefox, FTP-Commander, Google Chrome, and for CMAP Tools.

7.B.3 Services

The following are the services available through the computing infrastructure of the University:

- Connection to the Internet: it is supported by a 160-Mbps channel with the provider Media Commerce. Wireless access covers 95 percent of the campus through 140+ access points and it is available 24 hours a day 7 days a week. In addition, there is a connection to the Advanced Technology National Academic Network through a 100-Mbps channel, which allows access to Internet 2 and Geant 2. The intranet is at 1 Gbps.
- Email: it is available to all students, faculty and administrative staff. For students this is supported by Microsoft Office 365, with storage of 50 Gb and attachments up to 20 Mb, and there are about 26000 active email accounts. For faculty and administrative staff, email is supported by Microsoft Exchange 2007, with about 3270 active email accounts.
- On-line services for students: those allow them to carry out a vast variety of formalities and procedures, such as: initial registration for freshmen, on-line enrollment in courses, checking course time schedules, declaration of concentrations and minors, registration for professional internship, dropping out courses, access to Blackboard, booking of classrooms and computer rooms, evaluation of faculty, access to the library catalog and databases, application for study loans, request of transcripts and study certificates, access to virtual servers, downloading of regulations, registration for study exchanges, registration to student groups and associations, and registration to activities offered by the University Medium Vice-rectory.
- Services in support of teaching: those include lecture capture; streaming; and video- and teleconference by means of Tandberg Edge 95MPX and Polycom hardware, and Polycom Software PVX, Webex and Elluminate applications.

With regards to the stability and availability of the computing services, the target and commitment of the service providers is an average availability of 99.60 percent. Over the period June 2014 to March 2015, the average availability was 99.88 percent.

From all of the above, we conclude that the computing infrastructure is adequate in terms of technology and capacity, it is robust, reliable and sufficient backup is maintained, and, very importantly, it is flexible to accommodate the preferences of the users of Windows, Linux or Mac OS. Also, a number of services are directly available to support teaching through interactive methodologies. On the other hand, whereas the computer/faculty and computer/staff ratios are both 1:1, the computer/student ratio is 1:6, which might seem low but it is necessary to consider that not all students demand computer services simultaneously and most of them use their laptops or tablets for light-duty work. Therefore, we deem this infrastructure adequate for the attainment of student outcomes and the support to the needs of the Industrial Engineering program.

7.C Guidance

Student guidance regarding the use of facilities (laboratories, equipment, and computing resources) is shared between the faculty members whose courses use those facilities and the corresponding technicians.

Regarding laboratories and workshops, at the start of the semester, course coordinators inform students about the safety elements they have to acquire, i.e. lab coat, gloves, helmet, safety goggles, hearing protectors, and safety boots. Then, at the first visit to the laboratory the course coordinator and the technician present students with the safety regulations and with the laboratory itself, i.e. its equipment, the evacuation routes, and the use of the safety elements (fire extinguishers, fire alarm, stretchers). Regarding computing resources, during their sessions of insertion to the University, new students receive from the Center of Computer Services complete instructions as to the use of computer rooms and university-wide computing services. On the other hand, when needed, instructions as to the use of specific applications are given by course instructors.

The laboratory safety regulations include a complete set of recommendations, a sample of which is:

- Identify the nearest safety elements and always use the personal protection elements.
- Always follow the technicians instructions about the use of equipment, and never use equipment for which instructions are not available.
- Always behave appropriately, never work alone, never eat nor drink and always use the stools adequately.
- Always use appropriate clothing.
- Always verify the physical condition of equipment before working.
- Never work with electronic equipment while having hands or clothing wet.
- Never attempt to repair or tamper with any equipment.
- Never place liquids near electric equipment.
- After working with electronic equipment, first shut down, then disconnect by pulling the plug, never the cord.

7.D Maintenance and Upgrading of Facilities

In general, the maintenance and upgrading of facilities are a responsibility shared between the departments supporting the discipline-related core curricula and two administrative offices: the Office of Operational Services and the Center for Computer Service. As indicated in Section B3 of Criterion 8, this has a direct influence on the organization of the programs budget.

Laboratory equipment: For the Computer Science program, the Electronics and Computer Science Department is responsible for maintaining and upgrading the equipment of the laboratories described in Section A3 above. On its turn, the Natural Sciences and Mathematics Department is responsible for the maintenance and upgrading of the physics, chemistry and biology laboratories. The policy of the Civil and Industrial Engineering Department is to provide annual maintenance and calibration every other year; however, if a failure occurs unexpectedly, the repair is provided immediately.

Physical infrastructure: The Office of Operational Services is responsible for the maintenance and upgrading of the whole physical infrastructure of the University, including furniture. The following is a summary of policies, procedures and service statistics:

- The entire infrastructure is checked monthly to identify risky situations.
- The maintenance of classrooms, auditoria, projection rooms, laboratories and computer rooms is scheduled for the two inter-semester periods of the year. Walls are painted, floors are sealed, and rest-rooms are maintained (where applicable).
- The air-conditioning infrastructure is maintained and serviced monthly, every other month or quarterly according to the type of equipment; and this service is outsourced.
- Requests for emergent service are uploaded through a hotline administered by the Center for Computer Services. In the year 2014, 14,992 service requests were responded.
- The breakdown of the response time is: 26 percent within one day; 41 percent within two days; 12 percent within three days; 8 percent within five days; 7 percent within 10 days; 4 percent within 20 days; and only 0.5 percent takes longer than 30 days.
- In the year 2015 the budget of the Office of Operational Services, in Colombian pesos (\$), includes \$1,903 million in expenditures associated with infrastructure maintenance plus \$1,843 million in salaries and labor expenses. The exchange rate is 1 USD = \$2,400.

Computing infrastructure and services: As indicated in Section B above, the Center for Computer Services is responsible for the maintenance and upgrading of all the computing infrastructure and services of the University. The policies for maintenance and upgrading of hardware and software are summarized as:

- Hardware maintenance is provided permanently.
- Servers are replaced in cycles of three to four years; whereas the network hardware, storage and firewall are replaced in cycles of five years.
- The replacement cycles of computers are: five years for rotations of 30 percent of computers of computer rooms; five to six years for computers of program-specific computer

rooms and laboratories; and four years for faculty and administrative staff. A stock of 10 spare computers is held continuously for replacement of computer-room computers.

- The computers of faculty and administrative staff are replaced in cycles of four years.
- Much of the troubleshooting of software and applications is carried out by remote assistance through a hotline.
- With regards to software licenses, the policy is never to lapse on their renewal. Various schemes of renewal are used: many licenses are renewed annually; others are renewed every other year; and others are bought under perpetual licenses and upgrades are bought when available.

7.E Library Services

The main Library occupies an area of 918 m^2 and houses spaces for the collections; 2 common reading rooms with capacity of 106 users; 7 group reading rooms with capacity of 52 users; 32 individual reading modules; 6 stations for consultation of the on-line catalog (OLIB); and offices for the administrative staff.

Collections The total size of the collection changed from 54945 titles (134702 volumes) in 2010 to 67385 titles (with 168064 volumes) in 2014. The major components of the collection in 2014 were 57552 titles of books (comprising 80000 volumes), 1603 titles of serial publications (comprising 74024 volumes), 5302 undergraduate projects, and 74 data bases. The minor components of the collection comprise materials such as: maps, computer files, artifacts, videos, sound recordings, and psychological tests.

In 2014, the sizes of the mathematics and basic sciences collections were

- Mathematics: 838 titles (1686 volumes), with growth of 13 percent both in titles and in volumes since 2010.
- Physics: 228 titles (265 volumes), with growth of 11 percent in titles and 17 percent in volumes since 2010.
- Chemistry: 79 titles (87 volumes), with growth of 20 percent in copies and 33 percent in volumes since 2010.

Also in 2014, the technical collection associated with Computer Science comprised 1633 book titles (2361 volumes) with growth of 10 percent both in titles and in copies since 2010. In addition, the collection contained 116 volumes of undergraduate projects. The following are the numbers of book titles and volumes allocated to subject areas:

- Systems: 76 titles and 113 volumes.
- Data processing: 336 titles and 452 volumes.

- Programming: 809 titles and 1175 volumes.
- Artificial intelligence: 293 titles with 450 volumes.
- Computing engineering: 119 titles with 171 volumes

In addition, the technical collection includes the 10 databases, comprising 2580 journal titles that are presented in Table G.1 of the Appendix F.8.

Therefore, from the above, including the technical and mathematics/sciences collections and for an enrollment in 2014 of about 120 students, there was a ratio of about 38 volumes/student. This appears to be sufficient for an adequate service to students. On the other hand, the technical collection and the data bases cover the gross subject areas of the curriculum.

Over the period 2010 to 2014, the expenditure of the Electronics and Computer Science Department in library materials was (in Colombian pesos, \$) \$13.1 million in books, \$88.8 million in data bases, and \$5.4 in serial publications and data bases (exchange rate: 1 USD = \$1900).

7.E.1 Services

The Library offers services that are standard in the context of university libraries, i.e. on-line consultation of collections, acquisitions, loans, archiving of undergraduate projects and masters theses, exchange of materials with fellow libraries, donations of duplicate or defective materials, and informative talks to new students about the library services. In addition, there are 120 laptops and 60 iPads for loan.

The acquisition of library materials results from the needs of programs and faculty that departments consolidate in their annual budgets. Upon the preparation of the next annual budget, heads of departments survey their faculty about their request for library materials (see Section B1 of Criterion 8). When the budget is approved and enabled, then heads of departments pass onto the Library their requests for new materials, which does not preclude that faculty might request additional materials through the year provided the department can allocate the funding. Thus, the Library is placing purchase orders through the year. The response time depends on the capacity of the purveyor and on whether the materials are in stock or have to be imported; also, to expedite the process, the Library usually resorts to on-line purveyors. After the materials arrive in the Library, they pass through cataloging and then become available in the collections. New acquisitions bulletins are issued periodically and reach faculty through the heads of department. All in all, the process runs continuous and smoothly and new materials are available within four to six weeks after issuance of the requests.

On the other hand, access to serial publications and electronic materials is made through the data bases, by exchange with fellow libraries and, in particular, the retrieval of scientific papers is made through the Celsius platform. Celsius was created by Universidad Nacional de la Plata, Argentina, and its provider is ISTEAC (Ibero-American Science & Technology Education Consortium). Access to Celsius is through celsius.javerianacali.edu.co, the user

simply has to create an account and upload requests for fully-referenced papers. Usually, the response time is very short: it might be within 24 hours, and very rarely extends more than a couple of days. Therefore, access to publications and research papers is very robust and efficient.

Loans of materials are a typical service of the Library. The standard return times for general-collection materials are 8 days for regular students (up to 7 materials simultaneously) and 15 days for faculty and students in undergraduate projects (up to 12 materials simultaneously). Fines are issued for delayed return of lent materials, at the rate of \$3500/day for books and journals. Also, an inter-library loan service is available with 15 local and 8 national institutions; up to 2 books at a time can be borrowed for up to 8 days, with the possibility of renewal.

From all of the above, by and large, the Library services are enough in size, efficient and technologically founded to provide adequate support to academic programs, faculty and students.

7.F Overall Comments on Facilities

Ensuring the safety of facilities, tools and equipment used by the program is a responsibility shared between the program and the supporting department, and offices of the central administration, i.e. the Office of Operational Services and the office of Work Health and Safety of Human Resources.

The safety of general-purpose facilities such as classrooms, auditoria and projection rooms is the responsibility of the central administration. In general, for their purposes of congregating people, those facilities are safe, i.e. evacuation routes are labeled; doors are never locked while those spaces are in use; smoking, eating, drinking or activities other than academic are never allowed; and fire cabinets are available nearby.

The safety of equipment is a joint responsibility of the program and of the supporting department, in this case the Electronics and Computer Science Department, and of the two offices of the central administration indicated above. This also applies to the laboratories themselves. In general, safe equipment is acquired, but the safety of installations is ensured by the central administration upon recommendations from the program and the department. In particular for the Computer Science laboratories, safety conditions are not different from those of a typical computer room; thus fire extinguishers are available as well as voltage protection.

Criterion 8

Institutional Support

8.A Leadership

To assess the leadership of the Program, it is necessary to review the nature and organization of academic programs defined by the Academic Unit Regulations[2]:

- Academic programs are the units in charge of developing a curriculum leading to a university degree in a profession or discipline. To that effect, the main responsibility of academic programs is to develop the curriculum to achieve scholarly and professional excellence within a perspective of comprehensive and balanced education.
- The Program Director is the person in charge of directing the academic, university medium and administrative activities of the academic program and is responsible for the operation, academic quality, social relevance, economic viability, promotion of the educational processes, and wellbeing of students and faculty members supporting the program.
- Program Directors are appointed by the Academic Vice-rector from candidates nominated by the Dean of the School, and are accountable directly to the Dean.
- Program Directors are members of the Council of the School.
- The Program Committee is a collegiate body that assists and advises the Program Director as to the planning, devising, following up, evaluation and reform of the curriculum. Program Committee members are appointed by the Council of the School from the faculty supporting the core curriculum such that the major curricular areas are represented. In addition (see Section E of Criterion 6), the School of Engineering operates a system of cross memberships between Program and Department Committees.

From all of the above result the following conditions and mechanisms through which academic authorities exert leadership on the Program:

- There is an explicit regulatory framework that empowers agents to exert authority and be accountable for the Program.
- The Program Director represents the Program at the Council of the School of Engineering thus ensuring a two-way communication between the Program and the School and higher authorities.
- Besides the regulatory provisions, the cross memberships between Program and Department Committees allow for a close link between programs and departments. Thus, the Head of the Electronics and Computer Science department is invited to the Program Committee, and two of its members are members of the Department Committee (see Section E, Criterion 6).
- Finally, the academic credentials of the current Program Director, i.e. PhD degree plus 10 years of teaching and research experience, allow for an illustrated conduction of the Program.

Therefore, the leadership of the program is academic in origin and driven mainly by academic considerations, and is flexible and plural. For instance, these strengths have allowed the programs authorities to oversee and implement a number of critical processes in the last four years, i.e. the renewal of the national accreditation, the program assessment leading to the ABET evaluation, and the design and preparation of an ambitious curriculum reform to be submitted for approval by the end of the first semester of 2015. Thence, we consider that the programs leadership is adequate to ensure academic quality and the attainment of the program educational objectives.

8.B Program Budget and Financial Support

8.B.1 Process to establish the programs budget

The annual budget of the program is established following the guiding principles and parameters issued by the Office of Accounting and Budget for the general budget of the University. The guiding principles of the general budget are: transparency, overview, viability, sustainability and equity. The budget parameters are: the tuition fees; the salaries of all personnel; the fees for academic services; the per diem expenses; and an estimation of the exchange rate to be used in imports. As a private university, Javerianas major source of income is tuition fees, whereas minor sources are consultancy, sponsored research, donations and academic.

The programs budget follows the academic organization of academic programs and departments. Thus, the income for tuition and academic services is counted on the academic program, as well as the expenses associated directly with activities carried out by the Office of the Program Director. Academic services comprise the fees for supplementary exams, registration, course drops, and graduation duties. On the other hand, all the personnel expenses and investments on equipment, library, software licenses, and maintenance are counted on

the departments supporting the academic program. Also, the income from consultancy and sponsored research is counted on the departments.

The sources of information used to establish the programs budget are:

- The forecast of enrollment. This is done by the Program Director and is used to estimate the income and by the Secretary of the School to estimate the demand of course sections.
- Also, the Program Director defines the activities to be implemented, i.e. support to student groups, meetings, hiring of assistants, for instance.
- The Program Director and the Head of the Electronics and Computer Science department defines the budget for laboratory equipment, computers and software licenses.
- The Heads of Departments use the demand of course sections to allocate full-time faculty members and then to estimate the hiring of part-time faculty, and of student teaching assistants.
- Also, the Heads of Departments survey their faculty to estimate the demands for library, computers and software, and travel expenses.

The consolidated budget of the Program in the period 2010–2014 is presented in Table 8.1. The figures are in million Colombian pesos (\$), and the exchange rate to the US dollar over that period was largely stable at 1 USD = \$1,900. The budget includes the income from tuition fees, the expenditure in full-time and part-time faculty salaries, the investment in laboratory equipment and other expenditures such as support to student groups, promotional activities and miscellaneous activities organized by the Program Director.

	2010	2011	2012	2013	2014
Income	\$1105	\$1048	\$1127	\$1072	\$1110
Full-time faculty salaries	\$690	\$743	\$704	\$716	\$719
Part-time faculty salaries	\$21	\$0.5	\$13	\$24	\$42
Equipment	\$7		\$4	\$7	
Other expenditures	\$6	\$2	\$3	\$6	\$6

Table 8.1: Budget of the Computer Science program 2010–2014 (in million Colombian pesos. 1 USD = \$1900)

8.B.2 Support to teaching

Graders, other than faculty members, are not used in the University because grading is the sole responsibility of faculty members.

The support to teaching in terms of teaching assistants is regulated by the Academic Vice-rectory through a directive[3]. To the effects of this section, the most relevant topics of that directive are:

- Teaching assistantship is understood both as a contribution to the development of professional abilities and as the acknowledgment of the academic performance of students.
- Teaching assistantship must be carried out under the immediate supervision of the course coordinator.
- The Program Director, with the assistance of the Program Committee, identifies the courses that require teaching assistants upon the following considerations: enrollment, methodologies, percentage of students failing to pass the course and percentage of students dropping the course. Upon those criteria, the Head of the concerned department appoints the teaching assistants.
- To be appointed as teaching assistants, students must: be registered as active students; have a global weighted performance at least of 3.80; have had a grade of at least 3.80 in the concerned course; and not have received disciplinary sanctions.
- Before starting their assignments, teaching assistants must take a training course offered by the Academic Vice-rectory through the Office of Student Development.
- The maximum weekly workload of student assistants is 10 hours.

On the other hand, and as indicated in Section 6.D of Criterion 6, the Academic Vice-rectory through the Office of Faculty Development carries out various programs to support faculty development, a number of which are intended to support teaching: the program to develop teaching competencies, training in TICs, and the program of second language.

8.B.3 Support to infrastructure, facilities and equipment

The resources to acquire, maintain and upgrade infrastructures, facilities and equipment are assigned either to the programs budget or to the general budget according to their nature and intended use.

As indicated in Section 8.B.1 above, the budget of the Electronics and Computer Science Department includes the resources to acquire, maintain and upgrade program-specific laboratory equipment, computers and software licenses, i.e. those indicated in Section 7.A.3 of Criterion 7. On its part, the Department of Natural Sciences and Mathematics administers the resources to acquire, maintain and upgrade the equipment and hardware of the physics, chemistry and biology laboratories that support all the programs of the School of Engineering. These two departments administer the expenditures related to laboratory equipment; however, the Center of Computer Services administers the expenditures related to computers and software licenses although the resources are budgeted in the corresponding departments.

On the other hand, the Office of Operational Resources administers the resources assigned to the general budget that are related to the maintenance and upgrading of physical infrastructures in general, i.e. classrooms, meeting rooms, projection rooms, auditoria, laboratories and workshops, and offices.

8.B.4 Adequacy of resources

From the above, a number of conclusions are drawn:

- There is a regulatory framework and the mechanisms exist for the allocation of resources to the program.
- The budget of the program is defined through a well-informed process led by the Program Director in close consultation with the Head of the Electronics and Computer Science Department.
- The budget assigned to the program was shown to be sufficient to support the needs of the program.
- Units of the central administration provide important support for faculty development, training of teaching assistants, maintenance and upgrading of facilities, and computing services.

As a result of the above, our conclusion is that the budget and financial support are adequate to the needs of the Computer Science program.

8.C Staffing

The staff of the program is presented in Table D.2 of Appendix C.3. In summary, it comprises 10 full-time faculty members; 11 part-time faculty members (4 full-time equivalents); 10 student teaching assistants (5 full-time equivalents); the Program Director (who is a full-time faculty member), 1 full-time technician and 1 auxiliary secretary. The salaries of only the Program Director and the auxiliary secretary are included in the program's budget; the salaries of the rest of staff are included in the budget of the Electronics and Computer Science department.

The qualifications of the full- and part-time faculty members were discussed in Section 6.A of Criterion 6 and were deemed to be adequate. Thus, with regards to full-time faculty members, 50 percent hold PhD degrees and 30 percent hold Master degrees; with regards to the part-time faculty, 55 percent hold postgraduate degrees. On the other hand, the technicians hold professional degrees as technicians/technologists.

On the other hand, the size of the faculty was discussed in Section 6.C of Criterion 6 and it was shown that the allocation of full-time faculty members to course sections slightly exceeds the proportion of those faculty members in the Computer Science area of the Electronics and Computer Science department. Another consideration about the size of the faculty is the ratio of the number of students to the full-time equivalents: from the data of Tables D.1 and D.2 of Appendix C.3, for the year 2014 that ratio was 15 students/FTE. By and large, this ratio largely coincides with the average enrollment in the core-curriculum courses presented in Table 5.1 of Criterion 5, thus indicating that the size of course sections does not exceed the capacity of the program.

Upon demand from faculty members, in the 2014-2 semester, student teaching assistants were assigned to five core-curriculum courses, with a total weekly workload of 50 hours, thus corresponding to a 4 full-time equivalents.

From the above, the conclusion is that the staffing of the program is adequate both in quality and in size because it provides enough support to the programs needs.

8.D Faculty Hiring and Retention

8.D.1 Faculty hiring

In general, the hiring of new faculty is regulated by the Faculty Regulations, and in the School of Engineering this is implemented through a directive[8].

The most relevant provisions of the Faculty Regulations (numbers 37 to 39) regarding the hiring of new faculty are:

- Departments carry out the evaluation and selection of candidates according to the rank approved for the full-time position in question.
- The rank of the selected candidate must be determined according to the Faculty Regulations.
- The appointment is made by the Dean of the School.
- The documents of the appointed candidate must be sent to the Director of Faculty Development and to the Chief of Human Resources for the admission into the ranking of faculty and the signing of the employment contract.
- The Council of the School will specify the procedures to organize the call for candidates and the criteria of evaluation.

The Faculty Regulations (numbers 32 to 36) define the ranking of full-time faculty members, which in ascending order are: Instructor, Assistant Professor, Associate Professor, and Professor. In summary, the ranking is defined as:

- Instructor: it is the faculty member with minimum a Master degree, less than three years of academic experience and less than 120 points of academic production.
- Assistant Professor: it is the faculty member with minimum a Master degree, or with a second undergraduate degree, with academic experience, either at teaching or research, and academic production higher than those of Instructors.
- Associate Professor: it is the faculty member with a PhD degree, with academic experience and production higher than those required of Assistant Professors. The requisite of the PhD degree can be waived in favor of 400 additional points of academic production and performance evaluation.

- Professor: it is the faculty member with a PhD degree who has passed the minimum time of residence as Associate Professor and has accredited outstanding teaching and research.

Faculty members are assigned a score (number 82 of the Faculty Regulations), that qualifies and quantifies their performance and outcomes. The score is formed from various sources: performance evaluation, academic production, second degrees, distinctions and awards, and proficiency in non-native languages. The maximum scores of these sources are presented in the Appendix at the end of this chapter.

The admission of new full-time faculty members into the ranking is regulated by the Faculty Regulations (number 42) and by the Accord 89/2013[]of the Council of the Cali campus, and is summarized in Table 8.2. Salary levels were introduced to facilitate the improvement of salaries during the required time of residence at each rank. The rank of Professor was not included because that is achieved not by admission but by promotion from Associate Professor. It must be noticed that the requirements of degree, experience and academic production have to be met jointly. In addition, to promote proficiency in a second language, the following minimum levels of proficiency are required: A2 for Instructors, B1 for Assistant Professors and B2 for Associate Professors. The time equivalences are: 1 year as full time = 1 FTE; 1 year as half time = 0.5 FTE; 1 year as part time = 0.25 FTE.

Rank	Salary level	Degree required	Experience required (years)	Academic production (points)
Instructor	I	Master	None	None
	II	Master	None	None
Assistant Professor	I	Master	3	120
	II	Master	3	180
	III	PhD	3	180
Associate Professor	I	PhD	8	350
	II	PhD	8	400
	III	PhD	8	450

Table 8.2: Conditions for admission into the ranking of full-time faculty

On the other hand, the following is the summary of the directive applied in the School of Engineering:

- The principles governing the selection of new faculty are: equity, transparency, and objective assessment of the merits of candidates.
- All new faculty members must have, at least, a Master degree in areas relevant to the position sought.
- In practice, the selection of new faculty is only for the ranks of Assistant Professor and Associate Professor, i.e. the School is not interested in hiring Instructors.

- The criteria of evaluation and their relative weights are: PhD degree (15 percent); publications (20 percent); teaching and research proposals (20 percent); didactic presentation (10 percent); teaching and research experience (20 percent); and interview with the Selection Committee (15 percent). Each of these criteria is evaluated in a scale from 0 to 100 points, and the score of the candidate is the resulting weighted average.
- The didactic presentation is brief talk about a subject of interest for the candidate within the scope of the position sought.
- The candidacy of a candidate is invalidated if the evaluation were less than 60 points in any of the following criteria: teaching and research proposals, didactic presentation; or the interview.
- The Dean of the School will publish the call for candidates with a deadline of six weeks. This call will provide the areas of interest and all the requirements and procedures for candidates.
- The evaluation and selection of candidates will take up to four weeks after the closure of the call for candidates.
- Once a decision is reached, the Dean of the School will communicate the decision to the selected candidate within two weeks.
- The Selection Committee is integrated by the Dean, the Head of Department, the Director of the concerned program, and three full-time faculty members.

In addition, each of the evaluation criteria is graded according to the following instructions:

- PhD degree: it must be relevant to the academic area of the position sought.
- Publications: have to be given the points determined in number 93 of the Faculty Regulations but the overall score is calculated with the model presented in the Appendix D.7.2.
- Teaching proposal: it is evaluated according to: academic motivation of the candidate (35 points); academic profile of the candidate (35 points); and potential of contribution to the department (30 points).
- Research proposal: it is evaluated according to the following criteria each with 20 points: scientific and academic quality; scientific and social relevance; contribution of the candidate; methodology; and use of information sources.
- Didactic presentation: it is evaluated according to the following criteria each with 20 points: command of the subject; pedagogical ability; order and clarity of the presentation; resolution of questions; and use of the language and handling of the audience.

- Interview: it is evaluated according to the following criteria: consistency of the candidacy (30 points); resolution of questions (20 points); potential of retention (30 points); professional experience (20 points).
- Time equivalences: the same as indicated above but the overall scores is calculated with the model presented in the Appendix.

Finally, the admission of part-time faculty members is far less regulated; with the only provisions being numbers 52 to 54 of the Faculty Regulations. In summary: part-time faculty members are hired only for the duration of courses; they are not admitted into the faculty ranking; and are appointed by the Dean upon selection and recommendation of the Head of Department. In the School of Engineering the number of part-time faculty members has been fairly stable and Heads of departments are well acquainted with their credentials, experience and performance; in case new hiring is needed, the selection criteria are that the credentials and academic/professional experience are commensurate with the courses for which they are hired.

8.D.2 Retention of faculty

The retention of qualified faculty is impossible to ensure because it ultimately depends on a personal decision based on a pros and cons balance. However, both the University and the School of Engineering consider this as a critical issue and a number of strategies are in place.

A very important strategy is the improvement of salaries. The University strives to increase salaries annually, at least, in the same percentage as the inflation rate. However, faculty members can improve their salaries by promotion through the salary levels and ranks. The promotion through the ranking is regulated by the Faculty Regulations (number 47) and the Accord 89/2013, as summarized in Table 8.3.

As indicated in Section 8.D.1, the requirements of experience and academic production must be met jointly. Therefore, the promotion from Instructor to Assistant Professor requires a minimum of 3 years and a score of 150 points; from Assistant to Associate Professor requires a minimum of 5 years and 300 points, and from Associate to Professor requires a minimum of 7 years and 600 points. However, the effect of the intermediate salary levels is to shorten the times for the improvement of salaries. In addition, the scores required for promotion cannot be accumulated from one level to the next, so that the figures given in Table 8.3 are the scores that must be accumulated at each level. On the other hand, at least 30 percent of the scores for promotion must be earned by academic production. Finally, proficiency in a second language at level B2 is required for promotion to Associate Professor and to Professor.

As a second strategy, full-time faculty members earn bonuses linked to academic production. Those bonuses are paid once and do not constitute salary. There are two mechanisms to receive bonuses:

- A bonus is paid for the score earned by the academic production submitted and assessed in the previous calendar year. The Rector of the University establishes the monetary

Rank	Salary level	Degree required	Experience required (years)	Score (points)
Instructor	II	Master	None	None
Assistant Professor	I	Master	3 as Instructor	150
	II	Master	2 in the rank	250
	III	PhD		
Associate Professor	I	PhD	5 as Assistant	300
	II	PhD	2 in the rank	200
	III	PhD	4 in the rank	400
Professor	I	PhD	7 as Associate	600
	II	PhD	5 in the rank	300

Table 8.3: Conditions for promotion through the ranking of full-time faculty

value of each score point. This mechanism is administered by the Office of Faculty Development.

- In addition, a bonus is paid for each high-quality publication, i.e. papers in journals ranked by ISI. These bonuses are quantified in multiples of the legal minimum monthly salary, such that: 12 for journals in Q1, 6 for journals in Q2, 5 for journals in Q3 and 4 for journals in Q4. The bonuses are paid right after the faculty member submits the published paper to the Director of Research, Development and Innovation.

In addition to the above, full-time faculty members can receive stipends from consultancy and continued education and royalties for patented inventions or technological developments (numbers 107 and 108 of the Faculty Regulations). With regards to consultancy, faculty members can include those activities in their approved workload, but the stipends must come from the budgets of those projects; which must also include the cost the departments incur in replacing the concerned faculty members. Regarding continued education, full-time faculty members can undertake continued education courses apart from their approved workload.

8.E Support of Faculty Professional Development

The program supporting faculty professional development was presented in Section 6.D of Criterion 6 and was also mentioned in Section 8.B.2 above. This program comprises the support to faculty development regarding postgraduate studies, competencies development, training in TICs, learning of non-native languages, and counseling. As indicated in Section 8.D of Criterion, five full-time faculty members of the Computer Science program benefited from the support for postgraduate studies and from Appendix ??, another six faculty members have benefited from the activities supporting competencies development.

Other activities and needs of faculty members are supported through the budget of departments. It was mentioned above in Section 8.B.1 that as part of the preparation of the

budget, Heads of department carry out surveys to identify those requests. Thus, that budget supports the requests for attending seminars and workshops provided those are relevant to the departments needs and interests. Usually, the support for the presentation of talks at scientific conferences is provided by the budgets of research projects; although, occasionally, departmental funds can be used to those purposes.

Program Criteria

For Computer Science programs, the Computing Accreditation Commission program-specific criteria refer to program-specific student outcomes, curriculum, and faculty.

Program-specific student outcomes The discussion about the attainment of the student outcomes J and K was included in Section 4.A.5 of Criterion 4; however, it is worth recalling at this point that the average results in both student outcomes were 3.25 and 3.09, respectively, thus meeting the objective of minimum of 3.00, and that the pass ratios were 58 and 60 percent, respectively, thus meeting the objective of minimum 50 percent.

Curriculum The discussion was included in Section 5.A.5 of Criterion 5, where the curriculum was shown to meet the program-specific criteria.

Faculty Table 6.1 in Criterion 6 showed that five out of 10 full-time faculty members have PhD degrees in Computer Science, or closely related fields; also, it was mentioned in Section A of Criterion 6 that, as of December 2014, two additional full-time faculty members were to finishing their PhD studies. Therefore, we consider that the Program meets satisfactorily with the program-specific criterion of having some faculty members with PhD degrees.

References

- [1] 004A. Plan de estudios sistemas.
- [2] Pontificia Universidad Javeriana Bogot. Academic Unit Regulations, 2003.
- [3] Pontificia Universidad Javeriana Cali. Directive for the implementation of teaching assistanships in undergraduate programs, 2012.
- [4] Consejo Directivo de la Pontificia Universidad Javeriana. Acuerdo no 576. misión de la pontificia universidad javeriana, 2013.
- [5] IEEE/ACM. Computing curricula, 2004.
- [6] Pontificia Universidad Javeriana. Reglamento de Estudiantes. http://www.javerianacali.edu.co/sites/ujc/files/node/field-documents/field_document_file/reglamento_estudiantes_actualizado.pdf.
- [7] Pontificia Universidad Javeriana. Reglamento de Unidades Acadmicas, April 2014. http://www.javerianacali.edu.co/sites/ujc/files/documentos-institucionales/documentos/reglamento_unidades_academicas_unificado.pdf.
- [8] School of Engineering. Directive for the selection of full-time faculty, 2010.
- [9] Computer Science Program. Minute 20120418, 2012.
- [10] Computer Science Program. Minute 20121022, 2012.
- [11] Computer Science Program. Proyecto educativo. ingeniería de sistemas y computación, 2012.
- [12] Computer Science Program. Industry board assesment (survey on PEOs), 2013.

Appendix A

Course Syllabi

The syllabi presented here are those of the courses used to satisfy the discipline-specific and the mathematics and sciences requirements, and are presented following the semester sequence of Table 5.1. The following is the index to the syllabi:

Required courses:

300MAG018, Fundamentals of Mathematics	109
300CIG001, Introduction to Computer Science	111
300CIP001, Introduction to Programming	113
300IGO001, Introduction to System Modeling	115
300MAG002, Differential Calculus	117
300MAG006, Linear Algebra	119
300CIP002, Programming Fundamentals and Data Structures	121
300CIP007, Laboratory of Programming	123
300MAG007, Integral Calculus	125
300MAG031, Discrete Mathematics for Computing	127
300CIS004, Object-Oriented and Mid-Scale Programming	129
300MAG008, Multivariate Calculus	131
300FIF003, Kinematics and Dynamics	133
300CIG005, Computer Architecture I	135
300CIG002, Logic in Computer Science	137
300MAE005, Probability and Statistics	139
300FIF002, Electricity and Magnetism	141
300CIG006, Computer Architecture II	143
300CIG007, Computability and Formal Languages	145
300CID001, Information Management and Modeling	147
300MAA015, Numerical Analysis and Computing	149
300CIS006, Networks and Communication	151
300CIS005, Software Engineering Processes	153
300CIG008, Computer Graphics	155
300CIS002, Formal Software Development	157

300CIS001, Large-Scale Software Development	159
300CIG004, Analysis and Design of Algorithms	161
300CID002, Database Implementation	163
300IGG002, Fundamentals of Research	165
300CIS003, Web-Based Services	167
300CIG011, Operating Systems	169
300CIG009, Artificial Intelligence	171
300CIG010, Social and Professional Issues	173
300IGI003, Engineering Economics	175

Concentration in Animation and Interactive Systems:

300CIG032, Animation and Simulation	177
300CIG034, Interaction and Sound	179
300CIG033, Introduction to Video Games Development	181
300CIG035, Artificial Intelligence for Games	183

Concentration in Net-centric Computing:

300CIS014, Multimedia Technologies	185
300CIG030, Information Security	187
300CIG031, Mobile Computing and Mobile Agents	189

1. **Course code number and name:** 300MAG018, Fundamentals of Mathematics.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course coordinator:** Jorge Figueroa

4. **Text book:**

- Algebra, Trigonometry and Analytic Geometry, 3rd Edition, D. Zill, J. Dewar, 2012.
- Precalculus, 3rd Ed., J. Stewart, L. Redlin, S. Watson, 2001.

5. **Specific course information:**

Description of the contents:

This is a course in fundamental mathematics and pre-calculus, with topics required by engineering students as preparation for further studies in linear algebra, and differential and integral calculus.

Prerequisites: None

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify whether an argument is valid using truth tables.
- To solve counting problems
- To demonstrate propositions using direct, indirect, and mathematical induction methods
- To simplify algebraic expressions.
- To solve equations and inequalities.
- To describe the domain and range of functions, and to plot them.
- To solve problems using polynomial, exponential, logarithmic and trigonometric functions.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Propositional logic and sets.
- Demonstration methods.
- Algebraic expressions, equations, and inequalities.
- Functions and their graphic representations.
- Polynomial, exponential, logarithmic, and trigonometric functions and their applications.

1. **Course code number and name:** 300CIG001, Introduction to Computer Science
2. **Credits and contact hours:** 3 credit hours, 4 hours per week
3. **Course Coordinator:** Pablo Miguel Grech

4. **Textbook:**

The Art of Computer Programming, D.E. Knuth, 1974.

Supplemental materials:

- The Demon-Haunted World: Science as a Candle in the Dark, C. Sagan, 1995.
- A Discipline of Programming, E.W. Dijkstra, 1973.
- Revolution OS (video), J. T. S. Moore, 2001.
- Computing Machinery and Intelligence, A.M. Turing, 1950.

5. **Specific course information:**

Description of the contents:

This is an introductory course of the Computer Science undergraduate program. The course covers topic areas that students will further study along the program. These areas are studied in a shallow but technical way so that students get familiar with the discipline and its applications.

Prerequisites: None

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify important milestones of computing and engineering history.
- To identify the purpose of the main parts of a computer and their hierarchy.
- To describe the role of operating systems.
- To recognize the different topologies of a data networks and to describe the general function of Internet.
- To describe the object of study of the software engineering discipline and its applications.
- To enumerate the different paradigms of programming and to describe their important aspects.
- To identify aspects, depth and application of artificial intelligence.
- To recognize processes and tools used in the interactive-game development industry.

- To write documents using LaTeX.
- To transform and operate (add) numbers between the following numeral systems: binary, octal, decimal and hexadecimal.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3				2	2	4	2			

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- History of computing.
- Computer architecture.
- Operating systems.
- Computer networks.
- LaTeX.
- Software engineering.
- Programming languages.
- Logics and artificial intelligence.
- Numeral systems (binary, octal, hexadecimal).
- Game theory.

1. **Course code number and name:** 300CIP001, Introduction to Programming.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** Juan Carlos Martinez.

4. **Textbook:**

Introduction to Programming, G.M. Sarria, M. Mora, 2014.

Supplemental materials:

- How to Think Like a Computer Scientist: Learning with Python, A. Downey, J. Elkner, C. Meyers, 2002.
- Learn to Program using Python: A Tutorial for Hobbyists, Self-Starters, and all Who Want to Learn the Art of Computer Programming, A. Gauld, 2001.
- Learning Python, M. Lutz, D. Ascher, 1999.
- Problem Solving with Algorithms and Data Structures using Python, B.N. Miller, D.L. Ranum, 2006.

5. **Specific course information:**

Description of the contents:

The objective of the course is to present the discipline of programming as a conceptual and technological tool that allows solving real problems in engineering.

Prerequisites: None

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To model a problem computationally.
- To apply programming notions in solutions of problems.
- To use a programming language and an environment to implement algorithms.
- To interpret, develop, evaluate, and explain algorithms that solve a problem.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2		5	1				2	3	5	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Notion of System.
- Observation.
- State.
- Abstraction.
- Condition.
- Repetition.
- Data abstraction.

1. **Course code number and name:** 300IGO001, Introduction to Systems Modeling.
2. **Credits and contact hours:** 2 credit hours, 3 hours per week.
3. **Course Coordinator:** Camilo Rueda.

4. **Textbook:**

Modeling in Event-B: System and Software Engineering, J.-R. Abrial, 2010.

Supplemental materials:

- The B-Book, Assigning Programs to Meanings, J.-R. Abrial, 2005.
- Introduction to System Modeling, C. Rueda, 2013.

5. **Specific course information:**

Description of the contents:

This course presents fundamentals to design accurately the model of a system way and for reasoning about its behavior. Students use the Event-B language to define properties of the observations of a system along with actions that establish changes in those properties in different contexts. Through the use of the model animation software in Event-B, students analyze the behavior of the system and verify that it respects the expected properties.

Prerequisites: None.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the differences between models used in computing and models used in other disciplines.
- To enumerate examples of problems that engineering and computing can solve.
- To recognize the components of a system and their interactions.
- To model a system using abstraction and refinement techniques.
- To observe the behavior of a system by means of simulating its model.
- To establish simple properties about the behavior of a system and to verify them in the built model.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance		3	3			2	1			5	2

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Static and dynamic components of a system.
- Notion of “invariant” of a system. Relationship between observations and invariant. What is the “behavior” of a system?
- Observations: notion of “type”. Sets and their operations. Simple predicates.
- Modeling with sets. Use of set operations to establish properties of observations.
- Notion of “event”. Guard and action of events. Observations before and after events.
- System requirements. How to specify them. Requirements in Rodin. Relating requirements and components of the model.
- Predicates over arithmetic expressions. Complete model of systems with arithmetic observations.
- Sets: specifying elements and collections. Membership and inclusion. Specifying the addition and elimination of elements in collections. Properties of operations over sets. Predicates over sets. Use of sets in guards of events.
- Sets and arithmetic predicates in contexts and machines. Axioms and invariants. Limitations of simple sets in system design.
- Tables: specifications by means of relations.
- Types of relations.
- Relations vs. functions.
- Use of functions in sequences and arrays specification.
- Structuring large systems.
- Notion of model refinement.
- Concreting models by means of refinement chains.
- Event actions and refinement invariant.

1. **Course code number and name:** 300MAG002, Differential Calculus.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course coordinator:** Michell Andrés Gómez

4. **Text book:**

Calculus, 9th Edition, E. Purcell, D. Varberg, S. Rigdon, 2006.

Supplemental materials:

- Single Variable Calculus, J. Stewart, 2011.
- Calculus with Analytic Geometry, 5th Edition, H. Edwards, D.E. Penney, 1997.

5. **Specific course information:**

Description of the contents:

This course presents the concepts of limit, derivative, and primitive along with their fundamental properties to understand, model and solve geometrical, physical, and engineering problems.

Prerequisites: 300MAG012, Fundamental Mathematics.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the fundamental concepts of trigonometric functions and their derivatives.
- To analyze, formulate and solve problems using concepts and techniques of differential calculus.
- To quantitatively describe change and infinitesimal variation.
- To solve problems in modeling, optimization and rate of change.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Trigonometric relations in a right triangle.

- Trigonometric functions and their graphic representations.
- Sine and cosine laws.
- Inverse trigonometric functions.
- Trigonometric equations.
- Hyperbolic and inverse hyperbolic functions.
- Limits and continuity.
- Mean value theorem.
- Derivative of a function from its formal definition.
- Estimation of a derivative from the plot of the function.
- Modeling and optimization problems.

1. **Course code number and name:** 300MAG006, Linear Algebra.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course coordinator:** Michell Gómez

4. **Text book:**

Linear Algebra with Applications, 9th Edition, S.J. Leon, 2014.

Supplemental materials:

- Linear Algebra, 5th Edition, S.I. Grossman, 1996.
- Linear Algebra with Applications, G. Nakos, D. Joyner, 2004.

5. **Specific course information:**

Description of the contents:

The course presents the fundamental concepts of matrix algebra, vector spaces, and linear applications, aimed at the understanding, modeling, and problem solving in geometry, physics, engineering, and economics.

Prerequisites: 300MAG012, Fundamentals of Mathematics.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify and solve problems involving systems of linear equations.
- To describe and perform operations with matrices.
- To identify geometrical and analytical properties of vectors.
- To describe some classical vector spaces and their properties.
- To calculate and use eigenvectors and eigenvalues.
- To identify linear transformations and their properties.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Operations and properties of matrices, and systems of linear equations.

- Inverse and transpose of a matrix.
- Matrices with complex elements.
- Matrices and systems of linear equations.
- Gauss-Jordan elimination.
- Homogeneous and inhomogeneous systems of linear equations.
- Determinants.
- Norm and conditional number of a matrix.
- Ill-conditioned systems.
- Vectors.
- Operations with vectors.
- Dot and cross product of vectors.
- Vector equations of lines and planes.
- Orthogonality and colinearity of vectors.
- Polar coordinates.
- Generating set and linear independence.
- Linear transformations.
- Eigenvectors and eigenvalues. Diagonalization.

1. **Course code number and name:** 300CIP002, Programming Fundamentals and Data Structures.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:**Gerardo Mauricio Sarria.
4. **Textbook:**

Fundamentals and Structures of Programming, Class Notes, G. M. Sarria, 2012.

Supplemental materials:

- How to Program in C, 7th Edition, H. Deitel, P. Deitel, 2012.
- Design and Management of Data Structures in C, J. Villalobos, 1996.
- Problem Solving with Algorithms and Data Structures using Python, B.N. Miller, D.L. Ranum, 2006.
- Programming Languages. Design and Implementation, 4th Edition, T.W. Pratt, M. V. Zelkowitz, 2001.

5. **Specific course information:**

Description of the contents:

This course studies the foundations of high-level programming languages and focuses on abstract data types. The course provides the base for solving problems that can be solved using a computer and for knowing high-level programming languages and classic data structures (lists, stacks, queues, trees and graphs). Given that it is the second course of in the programming line of the plan of study, the practical component will be intensive especially in the development of software and in the analysis of algorithms.

Prerequisites: 300CIP001, Introduction to Programming

Type of course: Required.

6. **Specific goals of the course**

Learning objectives:

- To describe and to tackle different kinds of problem in computer science.
- To identify the main characteristics of high-level programming languages.
- To design and to implement solutions to computational problems by means of using Abstract Data Types (ADTs).

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	5		5			2		3	3		

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Problems in computer science.
- Calculating complexity.
- Problem solving strategies.
- Implementation strategies.
- High-level programming languages.
- Compilers, virtual machines, debugging, exceptions and GUIs.
- C programming language
- References and pointers, declarations and types.
- Abstract Data Types (ADTs)
- Lists.
- Stacks and Queues.
- Binary Trees.
- N-ary Trees.
- Graphs.

1. **Course code number and name:** 300CIP007, Programming Laboratory.
2. **Credits and contact hours:** 3 credit hours, 3 hours per week.
3. **Course Coordinator:** Luisa Fernanda Rincón.

4. **Textbook:**

C/C++: How to Program, H.M. Deitel, P.J. Deitel, 1994.

Supplemental materials:

- Programming in C, H. Schildt, 1988.
- The Programming Language C, B.W. Kernighan, D.M. Ritchie, 1985.
- Complete C, J.F. Peters, H.M. Sallam, 1986.
- Data Structures using C, A.M. Tenenbaum, L. Yedidyah, M.J. Augenstein, 1990.

5. **Specific course information:**

Description of the contents:

This course presents the tools needed to mature the concepts acquired in Introduction to Programming and to develop larger-scale projects.

Prerequisites: 300CIP001, Introduction to Programming.

Type of course: Required

6. **Specific goals of the course:**

Learning objectives:

- To design and describe properly the solution of basic computing problems.
- To use different techniques to solve problems.
- To recognize the necessity and utility of properly documenting source code.
- To use useful tools for developing projects.
- To identify main practical characteristics of compiled programming languages.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance			5					1	2		

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Pseudocode for designing solutions, compiled languages, C language, Gcc compiler, inputs and outputs.
- Operators, assignment, logics, precedence.
- C language, types of bugs, strings, operators, precedence, pre increment and post decrement, priority and associativity.
- Separation between headers and implementation, directives.
- Nested if sentences, if else, while loop.
- Switch instruction, do-while, for sentence, break, continue.
- Make tool.
- Code documentation, clean code.
- Pointers.
- Arrays.
- Strings.
- Matrices.
- Dynamic memory, typedef, structs.
- Basic ordering algorithms, Bubblesort.
- Basic ordering algorithms, Selectionsort, Insertionsort.
- Quicksort algorithm.
- Mergesort algorithm.
- Graphic user interface (GUI).
- Use of the simple linked list structure.
- Use of the doubly linked list structure.
- Use of stack and queue structures.
- Use of binary trees.
- Reading/Writing text and binary files.
- Use of graphs.

1. **Course code number and name:** 300MAG007, Integral Calculus.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course coordinator:** José Eduardo Tofiño
4. **Text book:**

Calculus of a Single Variable, R. L. Finney, G. B. Thomas, 1994.

Supplemental materials:

- Single Variable Calculus, J. Stewart, 2011.
- Calculus with Analytic Geometry, 5th Edition, H. Edwards, D.E. Penney, 1997.

5. **Specific course information:**

Description of the contents:

This course presents the foundations of indefinite and definite integration in the context of applications to science and engineering. The concepts of antiderivative and Riemann definite integral are developed, with application to area, volume, and curve lengths. Sequences and series are also discussed, along with applications to function approximations and representations.

Prerequisites: 300MAG002, Differential Calculus.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the fundamental concepts of integral calculus.
- To identify the fundamental theorem of calculus.
- To apply integration techniques to solve indefinite integrals.
- To identify the concepts of sequences and infinite series.
- To identify the Taylor and Mac Laurin series.
- To calculate the convergence of series.
- To calculate approximations of functions using geometric series.
- To calculate errors in approximations.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Approximate solution of definite integrals using Riemann sums.
- Fundamental theorem of calculus.
- Indefinite integral calculations using substitution, integration by parts, trigonometric substitutions, and partial fractions.
- Sequences and their limits.
- Infinite series.
- Convergence criteria.
- Representation and approximation of functions using geometric series.
- Taylor and Mac Laurin series.
- Power series.
- Error in approximations.

1. **Course code number and name:** 300MAG031, Discrete Mathematics for Computing.
2. **Credits and contact hours:** 4 credit hours, 5 hours per week
3. **Course Coordinator:** Michell Andrés Gómez.

4. **Textbook:**

Discrete Mathematics and Its Applications, 5th Edition, K.H. Rosen, 2004.

5. **Specific course information:**

Description of the contents:

This course presents the foundations of logic, sets and functions, basic discrete structures, counting principles, recurrence relations and Boole algebra to develop the formal mathematical thinking required in different contexts of engineering.

Prerequisites: 300MAG018, Fundamental Mathematics.

Type of course: Required.

6. **Specific goals of the course**

Learning objectives:

- To describe the basic concepts of logic, sets, functions and different demonstration techniques.
- To describe the essential properties of integer numbers.
- To define and to inductively and recursively demonstrate properties.
- To use basic counting techniques.
- To describe recurrence relations.
- To identify some types of graphs.
- To identify the basic properties of Boole algebra.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3								1	3	

1:low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Logic, propositional equivalences, predicates and quantifiers.
- Demonstration methods.

- Sets. Operations between sets.
- Functions.
- Growth of functions.
- Integers and division.
- Modular arithmetic.
- Integer representation, Euclidean algorithm.
- Some applications of the number theory.
- Successions y summations.
- Mathematical induction.
- Recursive definitions and structural induction.
- Foundations of combinatorics.
- Permutation y combinations.
- Binomial coefficients.
- Recurrence relations.
- Solving recurrence relations.
- Types of graphs and examples.
- Terminology in graph theory.
- Graph representation.
- Boolean functions.
- Boolean functions representation.
- Logic gates.
- Circuit minimization.

1. **Course code number and name:** 300CIS004, Objects and Mid-Scale Programming.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week
3. **Course Coordinator:** Antal Alexander Buss.

4. **Textbook:**

Objects First with Java: A Practical Introduction Using BlueJ, D.J. Barnes, M. Kölling, 2007.

Supplemental materials:

- Program Development in Java: Abstraction, Specification, and Object-Oriented Design, B. Liskov, J. Guttag, 2001.
- A Theory of Objects, M. Abadi and, L. Cardelli, 1996.
- The Unified Modeling Language Reference Manual, J. Rumbaugh, I. Jacobson, G. Booch, 1999.

5. **Specific course information:**

Description of the contents:

This course presents concepts of design and object-oriented programming (OO). The course uses the “Objects first” approach, which is the base of the OO paradigm. It also introduces notions of software engineering such as life cycles, UML diagrams, design patterns and PSP (Personal Software Process).

Prerequisites: 300CIP002, Foundations and Structures of Programming.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the characteristics of the object-oriented paradigm.
- To model a problem using the object-oriented paradigm.
- To describe basic concepts of software engineering.
- To analyze, design and implement a solution of a medium-scale problem using the OO paradigm.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3		5	1		1					3

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Definition of classes.
- Interaction between objects (association relationships).
- Clustering of objects in data structures (sequences, lists, iterators).
- Access modifiers, variables and constants (static), program documentation, new data structures (sets, hash maps).
- Responsibility driven design, coupling, cohesion.
- Inheritance (advantages, disadvantages, subtypes).
- Inheritance (static types, dynamic types, polymorphic data structures).
- Inheritance (methods overloading, protected access modifier, polymorphism).
- Some abstraction techniques (abstract classes).
- Some abstraction techniques (multiple inheritance, interfaces).
- Application design (analysis and design, prototyped, software life cycle models).

1. **Course code number and name:** 300MAG008, Multivariable Calculus.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course coordinator:** Jorge Figueroa

4. **Text book:**

Thomas' Calculus, Multivariable, 13th Edition, G.B. Thomas, M.D. Weir, J.R. Hass, 2013.

Supplemental materials:

- Multivariable Calculus, J. Stewart, 2011.
- Calculus with Analytic Geometry, 5th Edition, H. Edwards, D.E. Penney, 1997.

5. **Specific course information:**

Description of the contents:

This course introduces various techniques in differential and integral calculus of several variables, as well as their representation and applications in different fields.

Prerequisites: 300MAG007, Integral Calculus.

Type of course: Required

6. **Specific goals of the course:**

Learning objectives:

- To learn to apply techniques in differential and integral calculus of functions of several variables and vector functions.
- To define, operate and plot using curvilinear coordinates.
- To identify and plot surfaces in space, as well as vector fields.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Differentiability, linearization, differentials.
- Directional derivatives, gradient and tangent plane.
- Lagrange multipliers.

- Double and triple integrals.
- Change of variables in multiple integrals.
- Change to polar, cylindrical and spherical coordinates.
- Vector fields, line integrals, Green's Theorem.
- Flux of a vector field through a surface.
- Parametric surfaces.
- Surface integrals.
- Divergence theorem.

1. **Course code number and name:** 300FIF003, Kinematics and Dynamics.

2. **Credits and contact hours:** 4 credit hours, 5 hours per week.

3. **Course coordinator:** Luis Alfredo Rodríguez.

4. **Text book:**

Sears and Zemansky's University Physics, Volume 2, 13th Edition, H.D. Young, R.A. Freedman, 2013.

Supplemental materials:

Physics for Scientists and Engineers, R.A. Serway, J.W. Jewett, 2013.

5. **Specific course information:**

Description of the contents:

This course presents the fundamentals of Newtonian Mechanics for engineering students to analyze phenomena and applications related to the movement of bodies and mechanical systems under the action of forces. Also, the course seeks to apply fundamental laws and principles to concrete problems using mathematical language, calculus and vectors.

Prerequisites: 300MAG002, Differential Calculus.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the mathematical concepts needed to describe the movement of a particle, a rigid body, and a system of particles.
- To apply Newton's laws of motion to simple real life and engineering problems.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Measurement, dimensions, units, significant figures, error, and uncertainty.
- Vectors.
- Kinematics.

- Newton's Laws and applications.
- Work, kinetic and potential energy, energy conservation.
- Momentum, impulse, collisions; systems of particles, center of mass.
- Rigid body mechanics: kinematics and dynamics of rotational motion.
- Statics and conditions for equilibrium of rigid bodies.

1. **Course code number and name:** 300CIG005, Computer Architecture I.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** Jairo Andrés Velasco.

4. **Textbook:**

Digital Logic Circuit Analysis & Design, V.P. Nelson, H. Troy-Nagle, B.D. Carroll, J. D. Irwin, 1996.

Supplemental materials:

- Computer Organization and Design: The Hardware/Software Interface, 3th edition, J.L. Hennessy, D.A. Patterson, 2005.
- Structured Computer Organization, 5th edition, A. Tanenbaum, 2006.
- Digital Fundamentals, T.L. Floyd, 2014.
- Computer Organization and Architecture: Designing for Performance, 7th edition, W. Stallings, 2006.

5. **Specific course information:**

The course Computer Architecture I introduces the fundamental concepts of digital electronic systems and the relationship between these systems within a basic computing system.

Prerequisites: 300MAG031, Discrete Mathematics for Computing.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To do operations and conversions between different numerical bases and to represent any number in fixed point or floating point notation.
- To identify the fundamental blocks of digital logic.
- To minimize logic functions in functional blocks by means of sum-of-products using Karnaugh maps and the theorems of the Boole algebra.
- To solve problems of combinatorial and sequential logic using fundamental blocks.
- To identify the basic components of the Von Neumann architecture.
- To identify the modes of synchronization and data transfer between input/output devices and the processor.
- To calculate, project and organize required memory blocks to represent arrays of numeric and non-numeric data.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3		2	1					1	2	2

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Historic perspectives of computing.
- Numerical systems.
- Fundamentals of the Boole algebra, logic gates and fundamental blocks (decoders, registers, multiplexers, counters, memories).
- Basic computing system.
- Machine language.
- Input/Output devices.

1. **Course code number and name:** 300CIG002, Logic in Computer Science.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** José Eduardo Tofiño.

4. **Textbook:**

Logic For Computer Science: Foundations of Automatic Theorem Proving, J. Gallier, 2003.

Supplemental materials:

- Logic in Computer Science: Modeling and Reasoning about Systems, 2nd Edition, M. Huth, M. Ryan, 2004
- Mathematical Logic for Computer Science, 2nd Edition, M. Ben-Ari, 2001.
- Modeling in Event-B: System and Software Engineering, J.-R. Abrial, 2010.
- Interactive Theorem Proving and Program Development - Coq Art: The Calculus of Inductive Constructions, Y. Bertot, P. Castéran. 2004.
- Programming with Higher-Order Logic, D. Miller, G. Nadathur, 2012.

5. **Specific course information:**

Description of the contents:

Logic provides the machinery needed to check whether a given argument follows from a given set of premises. It allows to formally proving whether a statement is *true* or not. Since the 19th century, logic has been established as the language of mathematics. It thus gives the rigor needed to perform mathematical proofs. Logic is also of paramount importance for computer science and it has been called *the calculus of computer science*, i.e., it plays the same role that calculus plays for physical sciences and traditional engineering disciplines. For instance, the language of logic allows expressing, in a precise way, the specification of the system we want to build. Furthermore, logic provides *mechanisms* to (semi-automatically) prove if the proposed design and implementation satisfy the specification. In this course we study the syntax, semantics and procedure definitions for propositional logic and first-order logic. With the help of tools such as COQ and Rodin, students will apply their knowledge in the specification and verification of computer-based systems.

Prerequisites: 300MAG031, Discrete Mathematics for Computing.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To recognize the role of logic in mathematics and computer science.

- To translate statements from natural language into the language of Propositional Logic (PL).
- To establish the truth of a sentence in PL.
- To prove the validity of a sentence in PL.
- To specify properties by using the language of First Order Logic (FOL).
- To prove FOL formulas by using the sequent calculus.
- To relate computational steps and derivations in a proof.
- To specify and verify systems and program properties.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3									3	3

1: baja relevancia; 2: media relevancia; 3: alta relevancia.

7. Topics of the course:

- The language of PL
- Inductive proofs: Properties of the PL language.
- Satisfiability, unsatisfiability and tautologies.
- Equivalences and Boolean algebras.
- Proof Theory: Sequents and proof systems à la Gentzen
- Interactive proofs.
- Resolution in PL and Complexity
- Soundness and Completeness.
- Signatures, terms and substitutions.
- Semantics: Structures and models.
- Satisfaction and validity.
- Normal forms.
- Proof Systems for FOL.
- Interactive Proofs.
- Soundness and Completeness.
- Undecidability and incompleteness.
- Specification and properties.
- Specific theories and axioms: equality, sets, relations, functions, etc.
- Resolution in FOL.
- Horn Clauses and PROLOG.

1. **Course code number and name:** 300MAE005, Probability and Statistics.

2. **Credits and contact hours:** 3 credit hours, 5 hours per week.

3. **Course coordinator:** Sandra Milena Ramírez

4. **Text book:**

Statistics for Engineers and Scientists, W. Navidi, 2006.

Supplemental materials:

- Probability and Statistics, G. Canavos, 1988.
- Mathematical Statistics with Applications, W. Mendenhall, R. Scheaffer, D. Wackerly, 1986.

5. **Specific course information:**

Description of the contents:

This course presents the fundamentals of probability and statistics, and the descriptive and analytical methods to study the variability of data generated by different processes and systems. It presents the foundations of the statistical thinking to accompany decision-making in the context of uncertain events. These statistical methods enable engineers to perform experiments to improve quality, optimization, and development of products and services.

Prerequisites: 300MAG007, Integral Calculus.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To plan, summarize and present statistical information.
- To solve problems by the application of statistical rules and concepts, or the use of computational tools.
- To solve problems by considering probability distributions in one or two, discrete or continuous, random variables.
- To identify and use estimation methods of statistical parameters for one or two variables.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3	2							1	1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Sampling.
- Statistical summary and graphs.
- Counting methods.
- Conditional probability and independence.
- Random variables.
- Linear function of random variables.
- Joint distributions of random variables.
- Bernoulli's distribution.
- Binomial, Poisson, Normal, Log-normal, and Exponential distributions.
- Gamma and Weibull distributions.
- Probability graphs.
- Use of simulation to construct confidence intervals.
- Hypothesis testing.
- Correlations.
- Least squares fitting and its uncertainty.
- Assumption checking and data transformations.

1. **Course code number and name:** 300FIF002, Electricity and Magnetism.

2. **Credits and contact hours:** 4 credit hours, 5 hours per week.

3. **Course coordinator:** Alberto Pretel

4. **Text book:**

Sears and Zemansky's University Physics, Volume 2, 13th Edition, H.D. Young, R.A. Freedman, 2013.

Supplemental materials:

Physics for Scientists and Engineers, R.A. Serway, J.W. Jewett, 2013.

5. **Specific course information:**

Description of the contents:

The course presents concepts and applications of electrostatics, electric current and circuits, magnetostatics and magnetic induction phenomena, focusing on learning in an engineering context using problem solving, experimental, and inquiry techniques.

Prerequisites: 300FIF003, Kinematics and Dynamics.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the concepts of electric charge and electric field.
- To identify the concept of Gauss's law.
- To calculate currents and voltages in electric circuits.
- To identify the concept of magnetic field, Ampere's law and Faraday induction.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2	2								1	

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Electric charge
- Electrostatics, Gauss's Law.
- Electric current.
- Basic electric circuits.

- Magnetostatics.
- Ampère's Law.
- Faraday induction.

1. **Course code number and name:** 300CIG006, Computer Architecture II.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** Maribel Sacanamboy.
4. **Textbook:**

Computer Organization and Design, 4th Edition: the Hardware/Software Interface, J.L. Hennessy, D.A Patterson, 2009.

Supplemental materials:

- Structured Computer Organization, 5th Edition, A. Tanenbaum, 2006.
- Interconnection Networks: an Engineering Approach, J. Duato, 2003.
- Computers Architecture, J.P. Shen, M.H. Lipasti, 2006.
- Computer Organization and Architecture: Designing for Performance, 7th Edition, W. Stallings, 2006.

5. **Specific course information:**

Description of the contents:

Computer Architecture II introduces and discusses the organization and structure of a computer system. Instead of focusing on brands and the latest models of computers in the market, the course is focused on developing the fundamental of computer architecture from basic concepts. The course will show the evolution of concepts to take advantage of the architectural and technological characteristics that allow faster architectures.

Prerequisites: 300CIG006, Computer Architecture I.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the fundamental concepts of a digital computer architecture.
- To design and optimize a digital computer architecture.
- To implement a digital computer architecture.
- To evaluate the performance of a digital computer architecture.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	4		4	1		1			2	2	2

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Implementation of a datapath of the monocycle architecture.
- Implementation of the monocycle control unit (hardwired).
- Implementation of a datapath of the multicycle architecture.
- Implementation of the multicycle control unit (hardwired).
- Implementation of the multicycle control unit (microprogramed).
- Introduction to ILP.
- Pipeline management.
- Taxonomy.
- Shared memory systems.
- Metrics.
- Memory hierarchy.
- Memory organization and operations.
- Cache memory.
- Virtual memory.
- Failure recovery management.
- Cache coherence.
- Input-output devices, interconnection networks.
- Interrupts structure: vectored and prioritized.
- Introduction to networks.
- Interconnection networks.

1. **Course code number and name:** 300CIG007, Computability and Formal Languages
2. **Credits and contact hours:** 3 credit hours, 4 hours per week
3. **Course Coordinator:** Gloria Inés Álvarez
4. **Textbook:**

Introduction to the Theory of Computation, Second Edition, M. Sipser, 2006.

Supplemental materials:

- An Introduction to Formal Languages and Automata, 4th Edition, P. Linz, 2006.
- Introduction to Automata Theory, Languages and Computation, 3rd Edition, J.E. Hopcroft, R. Motwani, J.D. Ullman, 2007.
- Computers Ltd.: What they Really Can't Do, D. Harel, 2000.
- Compilers: Principles, Techniques and Tools, 3rd Edition, A.V. Aho, R. Sethi, J.D. Ullman, 2007.

5. **Specific course information:**

Description of the contents:

The aim of this course is to present fundamental topics of theoretical computing that are an essential part of the training of systems (computing) engineers. Regarding the study of formal languages, knowing and learning how to use models such as regular expressions or finite-state automata give students the possibility of extending the models that allow an accurate representation of diverse aspects of the real world. On the other hand, knowing the inherent limitations of the concept of computability reveals a (sometimes) unknown reality: not every problem is suitable of being solved by an algorithm; in fact, an endless number of problems cannot be solved by a computer. Finally, this course incorporates a very important element of the best practices of programming: the calculation of temporal and spatial complexity of algorithms. In conclusion, the course combines theoretical and fundamental topics that will enrich the students' vision about computer science and enable them to apply this science in the qualified practice of engineering.

Prerequisites: 300MAG031, Discrete Mathematics for Computing

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To define and to interpret computational models of diverse expressive power.

- To outline and argument about the truthfulness of the main concepts of computing theory.
- To analyze the complexity of a Turing machine and of an algorithm.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	4					1				4	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Regular languages.
- Deterministic and nondeterministic finite-state automata.
- Minimization of deterministic automata.
- Regular expressions.
- Application to lexical analysis.
- Context-free languages.
- Context-free grammar.
- Pushdown automata.
- Chomsky normal form.
- Application to syntactic analysis.
- Recursive and recursively enumerable languages.
- Deterministic and nondeterministic Turing machines with one or more tapes.
- Computing functions through Turing machines.
- Decidability.
- Reducibility.
- P, NP and NP-complete classes.
- Order notations of magnitude and temporal complexity.
- Space complexity.
- Complexity of algorithms over trees.
- Complexity of algorithms over graphs.

1. **Course code number and name:** 300CID001, Information Management and Modeling.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week
3. **Course Coordinator:**
4. **Textbook:**

Introduction to Database Systems, J.D. Ullman, J. Widom, 1999.

Supplemental materials:

- Fundamentals of Databases, A. Silberschatz, H.F. Korth, S. Sudarshan, 2004.
- Introduction to Database Systems, C. J. Date, 2001.
- Fundamentals and Database Models, A. de Miguel, M. Piattini, 1997.

5. **Specific course information:**

Description of the contents:

This course introduces concepts and techniques of database systems. The course studies the components of information system and the characteristics of a database management system (DBMS). The course focuses on relational databases and it uses the entity-relationship model and the relational model to represent typical problems where it is necessary to storage and manage data. The course also studies database design based on functional dependencies and normal forms. Finally, the student will implement databases and handle their contents via the SQL language.

Prerequisites: 300MAG031, Discrete Mathematics for Computing; 300CIS004, Objects and Middle-Scale Programming.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the fundamental concepts of database systems.
- To model a database from a problem or particular reality via the Entity-Relationship Model (ERM) and the Object Definition Language (ODL).
- To model a problem or a particular reality by means of the Relational Model (RM) and to express queries about the model using the Relational Algebra.
- To implement a database and to handle its contents via SQL (Structured Query Language).
- To design a database based on the functional dependencies of its components.
- To implement procedures to handle data in a database.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2		5		1	1	2		5		

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Database Management System.
- Data modeling concepts.
- ODL.
- Entity-Relationship Model.
- Exercises of data modeling.
- Introduction to relational model and transformations from ERM (and ODL) to RM.
- Referential integrity restrictions.
- Fundamental operations of Relational Algebra.
- Derived operations of Relational Algebra.
- Introduction to SQL.
- DML y DDL.
- Functional dependencies concepts.
- Functional dependencies rules.
- Database schemes design (DBNF, 3NF).
- Multivalued functional dependencies and 4NF.
- Functions, procedures and triggers.
- Indexes and views.

1. **Course code number and name:** 300MAA015, Numerical Analysis and Computing.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course coordinator:** Daniel Suescun

4. **Text book:**

Numerical Methods with Matlab, 3rd Edition, J.H. Mathews, 2000.

Supplemental materials:

- Numerical Analysis, J. Figueroa, 2001.
- Numerical Methods for Engineers, 5th Edition, S. Chapra, R. Canale, 2007.

5. **Specific course information:**

Description of the contents:

This course presents the foundations of traditional numerical methods and their engineering applications. Those will enable the student to find approximate numerical solutions to mathematical problems using computing tools. The theoretical existence and unicity conditions of such solutions are discussed, as well as the complexity and error in the corresponding algorithms. Finally, discrete transformed are introduced and discussed.

Prerequisites: 300MAG008, Multivariable Calculus; 300CIP001, Introduction to Programming.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the numerical methods for solving linear equations systems, nonlinear equations, interpolation problems, data fitting and integration.
- To apply algorithms for discrete and fast Fourier and other integral transforms.
- To identify the limitations of computers as tools to solve mathematical problems resulting from the modeling of real-life systems.
- To evaluate and choose, among several available methods to solve a given numerical problem, the most efficient one.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	5							3		2	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Basics of MATLAB programming.
- Numeric error
- Solutions of nonlinear equations.
- Solutions of linear equations systems.
- Interpolating polynomial for a set of data.
- Numerical integration rules for the calculation of definite integrals.
- Approximate numerical methods for initial value problems.

1. **Course code number and name:** 300CIS006, Networks and Communication.
2. **Credits and contact hours:** 3 credit hours, 3 hours per week
3. **Course Coordinator:** Carlos Andrés Olarte

4. **Textbook:**

Data and Computer Communications, W. Stallings, 2003.

Supplemental materials:

- Computer Networks, Andrew S. Tanenbaum, D.J. Wetherall, 2002.
- TCP/IP: Architecture, Protocols, and Implementation with IPv6 and IP Security, S. Feit, 1998.
- Cisco Networking Academy Program: First-Year Companion Guide, V. Amato, W. Lewis, 2002.
- Making, Breaking Codes: Introduction to Cryptology, P. Garrett, 2001.
- Secure Communicating Systems: Design, Analysis, and Implementation, M.R.A. Huth, 2001.

5. **Specific course information:**

Description of the contents:

This course presents the fundamentals of the technology of communication networks, focusing on data communication. It also studies the OSI model, concepts of routing and switching applied on network design.

Prerequisites: 300CIG006, Computer Architecture II.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify fundamental concepts of communication networks technologies.
- To describe the process of addressing in a data network.
- To identify the characteristics of different network architectures.
- To analyze a data network by means of the layer model.
- To describe different network protocols and the functions of network devices.
- To identify security risks in a data network.
- To identify the notion of netcentric computing and its applications.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3		1			2			2	2	2

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- OSI model vs TCP-IP protocol.
- Application layer.
- Transport layer.
- Network layer.
- Data link layer.
- Physical layer.
- Complete encapsulation process.
- Network simulation.
- Introduction to network design.
- Generalities about different routing protocols.
- Static routing and laboratory.
- Dynamic routing by RIP V1 and V2, laboratory.
- Application in network design.
- Generalities about VLAN and laboratory.
- Application in network design.

1. **Course code number and name:** 300CIS005, Software Engineering Processes.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week
3. **Course Coordinator:** Juan Carlos Martínez.

4. **Textbook:**

Software Engineering: A Practitioner's Approach, 7th Edition, R.S. Pressman, 2010.

Supplemental materials:

- Software Engineering, 9th Edition, I. Sommerville, 2011.
- Object-Oriented Software Engineering, A. Weitzenfeld, 2004.
- The Unified Modeling Language, Reference Manual, 2nd Edition, J. Rumbaugh, I. Jacobson, G. Booch, 2004.
- The Rational Unified Process and Introduction, 2nd Edition, P. Kruchten, 2003.
- Design Patterns: Elements of Reusable Object-Oriented Software, E. Gamma, R. Helm, R. Johnson, J. Vlissides, 1994.

5. **Specific course information:**

Description of the contents:

This course studies the processes involved in software development, from the software engineering perspective. The different stages of a software development process and their goal are reviewed. This course also introduces several software development methodologies and their application according to the context. It focuses on the application of the Rational Unified Process (RUP) and the establishment of required documentation for software projects. Finally, the concepts and techniques studied in the course are applied by means of a software development project.

Prerequisites: 300CID001, Data Management and Modeling.

Type of course: Required

6. **Specific goals of the course:**

Learning objectives:

- To identify the fundamental concepts of software engineering.
- To recognize software development life cycles, software development process models and software development methodologies.
- To recollect, analyze and model product requirements of a medium-scale software.
- To design a medium-scale software product.
- To develop, validate and test a medium-scale software product.

- To use methodic processes for analyzing, designing, developing, validating and supporting software of a real project.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	4	5	5	1		2			2	1	2

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Life cycles and models of process.
- Software development methodologies.
- Software development methodologies; Comparisons. Rational Unified Process (RUP)
- Software specification and requirements.
- Definition of requirements for a six months project (application of the RUP methodology). Classroom workshop.
- Requirements engineering and modeling.
- UML diagrams – Use cases
- Specification of requirement document.
- Presentation of a software development mini-project based on agile methods.
- Software design.
- UML diagrams – Classes and others.
- Design patterns.
- UML diagrams – Activity diagrams
- Best practices for designing.
- Software development standards.
- Quality of software.
- Verification and validation.
- Testing
- Metrics.
- Software evolution.
- Software support.

1. **Course code number and name:** 300CIG008, Computer Graphics.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course Coordinator:** Andrés Adolfo Navarro.

4. **Textbook:**

Computer Graphics: Principles and Practice in C, 2nd Edition, J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, 1995.

Supplemental materials:

- 3D Computer Graphics, 3rd Edition, A. Watt, 2000.
- Computer Graphics with OpenGL, 3rd Edition, D. Hearn, 2004.
- OpenGL Programming Guide: the Official Guide to Learning OpenGL, version 2, J. Neider, T. Davis, M. Woo, 2006.
- Computational Geometry: Algorithms and Applications, M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, 2008.
- Efficient Mesh Generation Using Subdivision Surfaces. A. A. Navarro, G. Wyvill, B. McCane, 2008.

5. **Specific course information:**

Description of the contents:

This course presents the foundations of computer graphics. From these concepts students will be capable to develop applications in two and three dimensions in a computational device. Students will identify structures, models, techniques and tools of computer graphics to develop these applications and will be able to implement those applications using modeling tools and graphic-oriented programming libraries.

Prerequisites: 300MAG06, Linear Algebra; 300CIS004, Objects and Middle-Scale Programming.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To know the fundamental concepts of computer graphics.
- To implement applications based on computer graphics.
- To explain the development of a computer graphics application.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	5		4			4		1	3	3	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Models of color, graphics pipeline, maps and bits depth. Basic objects of computer graphics, primitives, meshes and data structures.
- Geometric transformations, quaternions.
- Workshop of data structures and geometric transformations.
- Computational geometry, triangulation, subdivision surface, surface extraction, intersections and distances.
- Camera or observer model, projection modes.
- Illumination models, materials, light sources, transparency. Bitmaps texture, procedural textures, shadows.
- Introduction to ray tracing.
- Constructive solid geometry (CGS), metaballs, extrusions, surface of revolution.
- Foundations of digital images. Intensity and filters.
- Internet computer graphics technologies, introduction and examples. Cloud computer graphics. Mobile computer graphics.
- HTML5, WebGL.
- Human-computer interaction, user-centered design. Methodologies, techniques and trends of interaction, technologies and trends. GUI design principles.
- QT interfaces.

1. **Course code number and name:** 300CIS002, Formal Software Development
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** Camilo Rueda.

4. **Textbook:**

The B-Book - Assigning Programs to Meanings, 1st Edition, J.-R. Abrial, 2005.

Supplemental materials:

- Software Engineering with B, J.B Wordsworth, 1996.
- Logic in Computer Science: Modeling and Reasoning about Systems, 2nd Edition, M. Huth, M. Ryan, 2004
- Modeling in Event-B: System and Software Engineering, J.-R. Abrial, 2010.

5. **Specific course information:**

Description of the contents:

The objective of the course is to illustrate the convenience of using a nontraditional methodology of systems modeling and software development, i.e. formal modeling by means of case studies, which promotes the need of assigning the bulk of the modeling effort to the specification and design stages before the implementation and debugging. To this purpose, the B-method is used, supported by software that automatically generates proof obligations and automatically-generated Java code.

Prerequisites: 300CIP001, Introduction to Programming; 300CIG002, Logic in Computer Science.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the advantages of developing software with formal methods.
- To argument why formal methods are a mechanism of quality assurance in software.
- To formally describe complete systems by means of formal refinement.
- To design and verify systems using the Event-B method.
- To integrate the proof obligations in the software development process and to understand its importance.
- To express the model of a system in a support tool to the formal development.
- To analyze strategies to achieve an interactive proof.

- To apply interactive proofs to the development and verification of systems.
- To formally model a program.
- To build a program based on its formal model.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	1	2			1		1		1	5	5

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Use of mathematics in the modeling of systems.
- Where to integrate formal methods of design.
- Notion of model: static and dynamic parts.
- Machines, contexts. A simple system.
- What to prove in a system, invariants.
- Sequent calculus. Proofs in Rodin.
- Hierarchy of models.
- Relationship between abstraction and refinement.
- Refinement proof obligations.
- Refinement proofs in Rodin.
- What is a design pattern.
- Weak and strong synchronization.
- The model of a program.
- Refinement of programs, variant and invariant.
- To optimize from the invariant.

1. **Course code number and name:** 300CIS001, Large-Scale Software Development.
2. **Credits and contact hours:** 3 credit hours, 3 hours per week.
3. **Course Coordinator:**Luisa Fernanda Rincón.

4. **Textbook:**

The Making of Information Systems: Software Engineering and Management in a Globalized World, K.E. Kurbel, 2008.

Supplemental materials:

- The Unified Modeling Language Reference Manual, 2nd Edition, J. Rumbaugh, I. Jacobson, G. Booch, 2004.
- Service-Oriented Architecture: Concepts, Technology, and Design, T. Erl. 2005.
- Microsoft Solutions Framework Essentials, M. Turner, 2009.
- Agile & Iterative Development:A Manager's Guide, C. Larman, 2003.
- Business Modeling with UML: Business Patterns at Work, H.-E. Eriksson, M. Penker, 2001.

5. **Specific course information:**

Large-scale programming generally refers to software development of large dimensions involving not only development processes but also management, evolution and maintenance. In that context, the course presents topics concerning the methodologies of software development (SCRUM, RUP), software architectures and the best practices to manage software projects (estimation of size and schedule, allocation of resources, control of versions and changes, risk management, load and regression tests), and different frameworks for corporate and commercial software development.

Prerequisites: 300CIS005, Software Engineering Processes.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To recognize a big high-quality system in which development a defined process is followed, and to identify risks involved in this kind of software development.
- To learn the key characteristics of commercial and/or large-scale software development.
- To identify methods of software development according to the magnitude and complexity of the projects.
- To recognize best practices of software development management used in industry.

- To describe and apply necessary tools for specifying, designing, verifying and validating large-scale software systems.
- To learn the most used techniques and tools in industry for implanting large-scale software systems.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	1	5	5	3		1	2		3	1	5

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- The notion of size in software, case studies of software by size.
- Examples and project cases of complex software.
- Programming environments, tools for analysis and design.
- Tools for tests and configuration (Bugzilla, Maven, Ant, Subversion, Git).
- Large-scale software project management (agile methodologies (MSF, SCRUM) vs. formal methods).
- Common practices (project monitoring, deliveries, risk analysis).
- Tools for software project management.
- Software architectures.
- Component-based software development.
- Large-scale development techniques.
- Software development frameworks.
- Scalability techniques of large-scale applications.
- Enterprise application integration (EAI) techniques.

1. **Course code number and name:** 300CIG004, Analysis and Design of Algorithms.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** Juan Manuel Reyes.

4. **Textbook:**

Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, 2001.

Supplemental materials:

- Algorithmics: Theory & Practice, G. Brassard, P. Bratley, 1988.
- Algorithms, S. Dasgupta, C.H. Papadimitriou, U. Vazirani, 2006.
- The Algorithm Design Manual, S.S. Skiena, 1998.
- Computers and Intractability: a Guide to the Theory of NP-Completeness, M.R. Garey, D.S. Johnson, 2003.
- Foundations of Statistical Natural Language Processing. C.D. Manning and H. Schutze, 2000.

5. **Specific course information:**

Description of the contents:

This course presents the well-known techniques of algorithm design such as divide and conquer, dynamic programming, greedy search and backtracking. The algorithms developed with these techniques are applied to two different kind of fundamental analysis in algorithmic theory: correctness and efficiency. Additionally, this course covers the study of intractable problems and alternatives for their solution, for example, randomized, parallel or probabilistic algorithms. Finally, NP-Completeness theory is resumed; the course focuses on making demonstration by reduction.

Prerequisites: 300CIG007, Computability and Formal Languages.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To use design techniques to propose effective and efficient solutions to real-world problems.
- To apply classic algorithms (ordering, graph search, shortest path in graphs, among others) to models of real situations.
- To develop solutions to intractable problems.
- To rigorously analyze the performance of algorithms and operations in data structures.

- To argument the correction of classic and studied algorithms, and algorithms developed by students themselves.
- To choose the appropriate algorithm and data structure to be used in the solution of a given problem following criteria of computational effectiveness and efficiency.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3	2	3			2				5	1

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Divide and conquer: solution to recurrence equations.
- Master theorem.
- Dynamic programming.
- Greedy algorithms.
- Union-Find structure.
- Search trees.
- Red-Black trees. B-trees.
- Binomial heap.
- Backtracking.
- Simplex method of linear programming.
- Approximation and randomized algorithms.
- Demonstrations of NP-Completeness.

1. **Course code number and name:** 300CID002, Database Implementation.
2. **Credits and contact hours:** 2 credit hours, 2 hours per week
3. **Course Coordinator:** Alexander Yela

4. **Textbook:**

Database System Concepts, A. Silberschatz, H.F. Korth, S. Sudarshan, 2010.

Supplemental materials:

- Introduction to Database Systems, J.D. Ullman, J. Widom, 1999.
- An Introduction to Database Systems, C. J. Date, 2001.
- Oracle 10g – Management and Analysis of Databases, C. Pérez, 2008.

5. **Specific course information:**

Description of the contents:

This course continues the study of database technologies allowing the student to put into practice modeling and data design concepts. It also goes deep in techniques used to implement relational database management systems and XML data, all in enterprise application integration (EAI) context.

Prerequisites: 300CID001, Data Management and Modeling.

Type of course: Required

6. **Specific goals of the course:**

Learning objectives:

- To implement a database design in relational engines and XML.
- To decide which kind of database management system is the best choice for being used in the solution of a given problem.
- To identify possible alternatives for integrating applications in an enterprise environment.
- To identify roles of a database administrator (DBA).
- To design and implement concurrent transactions.
- To design and implement multidimensional databases.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2		3	2				3	2	5	4

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- E-R \rightarrow MRD. SQL-DDL.
- SQL: DML y SQL: Procedural option.
- PL/SQL.
- RDBMS: Architectures.
- XML databases.
- XPath - xQuery ().
- Data storage – Queries.
- Transactions – Concurrent transactions.
- Transaction planning.
- Concurrency control – Recovering.
- Situations in the current environment of Information Systems – EAI.
- HDBMS
- Data warehouses.
- Information recovering.
- Temporal data and special data.

1. **Course code number and name:** 300IGG002, Fundamentals of Research.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course coordinator:** Hernán Darío Benítez

4. **Text book:**

The Craft of Research, W. Booth, 3rd Edition, G. Colomb, J. Williams, 2008.

Supplemental materials:

- The Value of Basic Scientific Research, ICSU Working Group, 2004.
- On Innovation, G.C. Stelluto, 2007.
- English for Writing Research Papers, A. Wallwork, 2011.

5. **Specific course information:**

Description of the contents:

The objective of the course is to train students to use research as a tool to improve their professional performance. The course presents the fundamentals for students to write the proposal of the undergraduate project and is structured about the following topics: analysis of problems and identification of needs, review and analysis of literature, scientific communication and research ethics. Faculty members from the Department of Electronics and Computer Science make presentations about their research interests.

Prerequisites: Thermal and Wave Physics.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To distinguish between scientific research, applied research and innovation.
- To identify the components of the scientific method.
- To identify the role of engineering in society and its relationship with industrial innovation.
- To identify the trends shaping research interest at the national scale.
- To identify the essential components of a research proposal. Problem identification, identification of research opportunities, literature search, communication.
- To identify the ethical standards and conflicts rising upon doing research.
- To apply the components and steps of the research methodology.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance		5		3	2	3	2	3			

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Scientific research. Applied research. Innovation. Economic development. The scientific method.
- Identification of needs and analysis of problems. The technique of the nine windows to analyze research problems.
- Research interests in the Department of Electronics and Computer Science.
- Literature search and organization.
- The research process. Research problem. Objectives. Theoretical framework. Hypotheses. Variables. Revision of patents.
- Scientific communication. Reading and writing of scientific papers. Oral presentations. The preparation of the proposal of the undergraduate project.
- Research ethics. Handling of data. Authorship. Errors and negligence. Misbehavior. Professional standards. Intellectual property. Sharing of research results. Conflicts of interest. Research and society.

1. **Course code number and name:** 300CIS003, Web-Based Services.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course Coordinator:**Luisa Fernanda Rincón.

4. **Textbook:**

Web Engineering: The Discipline of Systematic Development of Web Applications, G. Kappel, B. Prýýll, S. Reich, W. Retschitzegger, 2006.

Supplemental materials:

- Java 2EE and XML Development, K.A. Gabrick, D.B. Weiss, D. Weiss, 2002.
- Data on the Web: From Relations to Semistructured Data and XML, S. Abiteboul, P. Buneman, D. Suciu, 1999.
- Web Application Architecture: Principles, Protocols and Practices, L. Shklar, R. Rosen, 2009.
- Pro JSF and Ajax: Building Rich Internet Components, J. Jacobi, J.R. Fallows, 2006.

5. **Specific course information:**

Internet has become an important platform for implementing software applications in diverse business domains. Such applications must respond to requirements about usability, performance, security, reliability and scalability. This makes necessary for the developer to know the nature of their composition, mechanisms and used and available technologies. The objective of this course is to present the development and implementation of those applications.

Prerequisites: Communication Networks (300CIS006)

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the structure and technological components of a Web application.
- To describe the techniques and standards involved in its development and administration.
- To recognize the principles in the design of web applications and sites.
- To manipulate audit and design tools for creating and managing sites.
- To identify aspects of security, ethics and legal issues in the web.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance		2	1	1		1		2	1		5

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Introduction to the Web, Web 2.0 and Semantic Web.
- Client-Server model (Java RMI).
- Technologies for the web (Javascript, PHP, jQuery, Dart, Ajax).
- MVC architectural pattern.
- Knowing some frameworks for web development (PHP – Java – ASP.Net – Ruby).
- Security in Web applications.
- Usability in Web applications.
- Non-SQL databases.
- Cloud computing.
- SOAP/Rest Web services.

1. **Course code number and name:** 300CIG011, Operating Systems.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course Coordinator:** Andrés Becerra.

4. **Textbook:**

Operating Systems, A.S. Tanenbaum, 1999.

Supplemental materials:

- Modern Operating Systems, A.S. Tanenbaum, 1993.
- Operating Systems, G. Nutt, 2004.
- Operating Systems Concepts, A. Silberschatz, G. Gagne, P.B. Galvin, 1999.
- Operating Systems, William Stallings, 1992.

5. **Specific course information:**

Description of the contents:

This course presents the fundamentals and efficient use of modern operating systems by means of presentation, study and analysis of their main components.

Prerequisites: 300CIG006, Computer Architecture II.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To enumerate the milestones in evolution of computer systems.
- To write concurrent programs using different constructors.
- To identify the difference between competition (critical section) and cooperation.
- To use semaphores to solve cooperating processes problems.
- To solve coordination problems using high-level constructors.
- To identify the notion of kernel of an operating system.
- To apply scheduling principles.
- To identify the notion of deadlock.
- To identify physical memory.
- To apply virtual memory principles.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	4							2	2	5	3

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- The Process Notion
- Defining and Instantiating Processes.
- Shared Memory Methods.
- Other Classic Synchronization Problems.
- Kernel Definitions and Objects.
- Implementing Processes and Threads.
- A System Model.
- Dynamic Deadlock Avoidance.
- Preparing a Program for Execution.
- Allocation Strategies for Variable Partitions.
- Principles of Virtual Memory.
- Memory Allocation in Paged Systems.
- Single-Copy Sharing.
- Sharing in Paging Systems.
- Hierarchical Model of a File System.
- File Directories.
- Device Organization Methods.
- A Hierarchical Model of the Input/Output System.
- Device Drivers.
- The Access Control Environment.
- High-Level Access Control.

1. **Course code number and name:** 300CIG009, Artificial Intelligence.

2. **Credits and contact hours:** 3 credit hours, 3 hours per week.

3. **Course Coordinator:**Gerardo Mauricio Sarria.

4. **Textbook:**

Artificial Intelligence: A Modern Approach, P. Norvig, S. Russell, 2009.

Supplemental materials:

- Turing Test: 50 Years Later, A.P. Saygin, I. Cicekli, V. Akman, 2000.
- Agent Technology: Foundations, Applications and Markets, N.R. Jennings, M.J. Wooldridge, 1998.
- An Introduction to Genetic Algorithms, M. Mitchell, 1998.
- Programming with Constraints: an Introduction, K. Marriott, P. Stuckey, 1998.
- Artificial Intelligence and Knowledge Engineering, G.P. Martinsanz, M. Santos, 2006.

5. **Specific course information:**

Description of the contents:

John McCarthy of Stanford University has defined artificial intelligence as “the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.” This course presents diverse techniques of artificial intelligence.

Prerequisites: 300CIG004, Design and Analysis of Algorithms.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To describe philosophical considerations about intelligence in machines.
- To apply techniques for solving constraint satisfaction problems (CSP).
- To identify and use formalisms and technologies to represent knowledge.
- To detail formalisms for reasoning in uncertainty.
- To apply knowledge discovery and data mining techniques.
- To apply basic mechanisms for automatic learning.
- To explain notions and basic algorithms of neural networks and genetic programming.

- To describe robotics related concepts.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	5					1	2	1		5	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- CSP.
- Search and genetic algorithms.
- Deduction and reasoning systems.
- Probabilistic/Statistical methods.
- Knowledge discovery and data mining.
- Machine learning / Neural networks.
- Robotics.
- Multi-agent systems.

1. **Course code number and name:** 300CIG010, Social and Professional Issues
2. **Credits and contact hours:** 2 credit hours, 4 hours per week.
3. **Course Coordinator:** Diego Fernando Polo.

4. **Textbook:**

A Gift of Fire: Social, Legal, and Ethical Issues in Computing, S. Baase, 1997.

Supplemental materials:

- Technology and Society, J.L. Harrington, 2009.

5. **Specific course information:**

Description of the contents:

The Social and Professional Issues course presents the cultural, social, legal and ethical issues related to the professional practice of the systems (computing) engineer. The course motivates students to adopt a critical and ethical posture with respect to situations related to the use of technology from a general vision of the social and professional context in which computing is developed.

Prerequisites: 144 approved credit hours.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the social, cultural and professional aspects related to computing ethics.
- To explain and justify the opinions related to social and cultural aspects that have been, and still are, altered by computing.
- To recognize the trends of application of technology in the professional practice.
- To argument about the philosophical and ethical questions emerged from the use of computers, with a professional vision beyond the technical problems.
- To explain the causes and consequences of possible mistakes made in a software development and with the use of technology.
- To recognize the importance of security in software products and technology.
- To identify the basic legal aspects related to software and hardware.
- To describe the issues related to professional contracting: kinds of contract, requests for bids, execution and settlement, auditing, evaluation of proponents, best practices for contract management.

- To effectively communicate, oral and written, opinions about discussed topics.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance				1	5	5	5	2			

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Social implications of networks. Internet growth, control and access. Cultural aspects.
- Technology trends in organizations (focusing on lecturer's area of expertise)
- Ethics, ethical relativism, professional ethics.
- Codes of professional ethics.
- Analysis of ethical cases.
- Risks created by software, implications of software complexity.
- Physical security, access control, operational security, security policies for systems and networks.
- Intellectual property: copyrights, patents, trade secret, piracy, transnational aspects.
- Privacy and civil rights: ethical and legal foundations, implications of database systems.

1. **Course code number and name:** 300IGI003, Engineering Economics.

2. **Credits and contact hours:** 3 credit hours, 4 hours per week.

3. **Course coordinators:** Jorge Enrique Álvarez; John Wilmer Escobar

4. **Text books:**

- Fundamentals of Engineering Economics, 2nd Edition; C.S. Park, 2009.
- Engineering Economics, 6th Edition; L. Blank, A. Tarquin, 2006.

5. **Specific course information:**

Description of the contents:

This course is focused on the fundamental principles of the evaluation of engineering projects and on the analysis of decisions related to capital investment, cost estimation, finance mathematics and decision under risk and uncertainty.

Prerequisites: 100 approved credit hours.

Type of course: Required.

6. **Specific goals of the course:**

Learning objectives:

- To identify the fundamental concepts, terminology and notation of engineering economics.
- To apply the criteria of economic decision to compare alternatives of capital investment.
- To identify the methods of economic evaluation of investment alternatives in the public sector.
- To calculate the cash flow of an investment project.
- To evaluate investment projects under risk and uncertainty.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	3						3				

1: low relevance; 2: medium relevance; 3: high relevance.

7. **Topics of the course:**

- Engineering economics for decision making.
- Interest rates and return rates, simple interest, and compound interest.

- Minimum attractive return rate.
- Linear series, geometrical series.
- Nominal and effective interest rates.
- Independent projects, mutually excluding projects, types of investment alternatives.
- Analysis of net present value, capitalized costs.
- Analysis of equivalent annual value, capital costs, operating costs.
- Analysis of return rate, return on investment, return on invested capital.
- Economic evaluation of public sector projects.
- Inflationary effects.
- Costs categories, breakeven point, cash flows before taxes, cash flows after taxes.
- Methods to describe the risk of a project, sensitivity analysis for mutually excluding alternatives, probabilistic approach.

1. **Course code number and name:** 300CIG032, Animation and Simulation
2. **Credits and contact hours:** 3 credit hours, 3 hours per week
3. **Course Coordinator:** Andres Navarro Newball

4. **Textbook:**

Computer Animation, 2nd Edition: Algorithms and Techniques (The Morgan Kaufmann Series in Computer Graphics), R. Parent, 2007.

Supplemental materials:

- 3D Games: Real-Time Rendering and Software Technology, 1st Edition, A. Watt, F. Policarpo, 2001.
- Advanced Game Development with Programmable Graphics Hardware, A. Watt, F. Policarpo, 2005.
- Computer Graphics: Principles and Practice. J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, 1996.
- Game Physics (Interactive 3d Technology Series), D.H. Eberly, 2003.
- Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3D Technology), C. Ericson, 2005.

5. **Specific course information:**

Description of the contents:

Cinema, entertainment, and videogames industry use effects and scenes increasingly more complex and realistic. Some of those are supported by concepts of animation increasingly complex and realistic. Others are supported by concepts of animation and simulation. Also, in many cases entertainment and modern education systems are supported by those technologies. This course focuses on giving movement to objects.

Prerequisites: 300CIG008, Computer Graphics.

Type of course: Selected Elective

6. **Specific goals of the course:**

Learning objectives:

- To learn the difference between animation and simulation.
- To give motion to objects inside a virtual graphic world.
- To apply interpolation and modeling based on physics in order to produce motion.
- To develop special effects to give more realism to objects.
- To use computer graphics libraries to produce motion.

- To use an animation tool.
- To model rigid and soft objects and their interactions.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2			3					3	4	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Cubic spline, bi-cubic surface, curves
- Interpolation, animation control
- Kinematics, inverse kinematics
- Digital characters: case studies, skeletons, rigging, skinning, anatomy
- Parametric deformation: fold, twist, smash, free deformation
- Facial animation: background, expressions, expression coding, human systems, animal expressions
- Simulation: modeling based on physics, simulation cycle, Euler and Lagrange integration, virtual reality, augmented reality
- Rigid and soft objects: spring-mass systems, finite element modeling, discrete element modeling
- Collision detection: bounding boxes, voxels, trees of spheres, tabulated spheres subsets
- Fractal geometry and its applications, plants generation (L-Systems), lands generation
- Special effects: systems of particles, hair, fire

1. **Course code number and name:**300CIG034, Interaction and Sound.

2. **Credits and contact hours:** 2 credit hours, 2 hours per week.

3. **Course Coordinator:** Gerardo Mauricio Sarria.

4. **Textbook:**

Making Media: Foundations of Sound and Image Production, J. Roberts-Breslin, 2007.

Supplemental materials:

- Computer Music, C. Dodge, T.A. Jerse, 1997.
- The Computer Music Tutorial, C. Roads, 1996.
- The Complete Guide to Game Audio, A. Marks, 2008.
- Digital Signal Processing in Communication Systems, M.E. Frerking, 2003.
- How to Read a Film: The World of Movies, Media, and Multimedia, J. Mnaco, 2000.

5. **Specific course information:**

Description of the contents:

This course presents the fundamentals of sound, music and its production. In addition, the course explores the relationships between sound and images, sound and animated objects and music and people. It also studies processes of interaction, improvisation and multimedia integration.

Prerequisites: 300CIS004, Objects and Mid-Scale Programming.

Type of course: Selected Elective.

6. **Specific goals of the course:**

Learning objectives:

- To describe basic physics concepts of sound.
- To use with different computational tools to filter, analyze, compose, synthesize and, in general, sound treatment.
- To recognize relations between images and animated objects with music and sounds.
- To make an improvisation, interaction and musical creation process.
- To develop a multimedia production process.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	1							4	4		

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Sound: physical properties, musical and scientific terminology, tools, MIDI.
- Perception and aesthetics (process of listening, what sounds good).
- Speakers and microphones.
- Sampling, synthesis, DSP.
- Sounds and images.
- Movement (animated objects, Flash).
- Improvisation, interaction and creation.
- Production: multimedia interaction.

1. **Course code number and name:** 300CIG033, Introduction to Video Games Development
2. **Credits and contact hours:** 2 credit hours, 2 hours per week
3. **Course Coordinator:** Diego Fernando Loaiza
4. **Textbook:**

The Art of Game Design: A Book of Lense, J. Schell, 2008.

Supplemental materials:

- Game Engine Architecture, J. Gregory, J. Lander, M. Whiting, 2009.
- Rules of Play: Game Design Fundamentals, K. Salen, E. Zimmerman, 2004.
- Half-real: Video Games between Real Rules and Fictional Worlds, J. Juul, 2005.
- Computer Graphics with OpenGL– 3rd edition, D. Hearn, 2004.
- The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments, R.S. Kalawsky, 1993.

5. **Specific course information:**

Description of the contents:

Nowadays games are one of the greatest trends in computing. In entertainment industry they are used as elements of social interaction and educational tools in different contexts. They are popular in many platforms and they use advanced engines and software development technologies. Games are an excellent example of computing, multimedia interaction, creativity and interdisciplinary. This course presents the fundamental concepts to design and implement video games.

Prerequisites: 300CIG008, Computer Graphics.

Type of course: Selected Elective.

6. **Specific goals of the course:**

Learning objectives:

- To apply the video game development methodology.
- To produce a simple video game.
- To work with an inter-disciplinary team.
- To play different roles in a video game development team.
- To adapt a story to a video game development.
- To use video game development engines.
- To evaluate video games.

- To apply advance concepts of computer graphics to video game development.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance				3		3			5		

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Video game engines: definition, architecture, use.
- Prototype: prototype, tests, documentation, assets, mechanics, story, interaction, characters (leading role, opponent), sprites.
- Balance: balance, levels, strategies.
- Design perspective: media edition tools, art of the game.
- Technology: consoles, computers, smart phones, 2D and 3D graphics, the role of artificial intelligence, network games.
- Models of development and distribution.
- Effects: motion parallax, explosions, real and cartoon physics.
- Interaction: devices, interaction metaphor, multi-modality.
- Wrap up: game post mortem.

1. **Course code number and name:** 300CIG035, Artificial Intelligence for Video Games.
2. **Credits and contact hours:** 2 credit hours, 3 hours per week.
3. **Course Coordinator:** Diego Fernando Loaiza.
4. **Textbook:**

Programming Game AI by Example, M. Buckland, 2005.

Supplemental materials:

- Artificial Intelligence Strategies Applied to Chess, J.G. Camacho, A.A. Navarro, 1994.
- Programming a Computer for Playing Chess, C. Shannon, 1950.
- Chess and Computers, D. Levy, 1976.
- AI Game Programming Wisdom 4, S. Rabin 2008.
- AI Game Programming Wisdom 2, S. Rabin 2004.

5. **Specific course information:**

Description of the contents:

When we talk about Artificial Intelligence (AI) we are talking about all the tools and methods that can be used to make computer-controlled entities in a videogame appear to behave intelligently. AI often gets far less attention than it deserves, whereas graphics gets far more attention. Indeed, the course also explores some visual techniques that can be used to make the videogame appear natural or in a way “intelligent” for the user. Finally, we discuss the problem of keeping the player interested in the game and present multiplayer massive online games as an alternative to AI and visual effects, which provides real human intelligence to other game characters.

Prerequisites: 300CIG008, Computer Graphics.

Type of course: Selected Elective

6. **Specific goals of the course:**

Learning objectives:

- To apply the videogame development methodology.
- To produce a simple videogame.
- To work in an interdisciplinary team.
- To play the different roles in a videogame development team.
- To adapt a story to a videogame development.

- To use videogame development engines.
- To evaluate techniques that allows perceiving the videogames as intelligent.
- To apply advance concepts of computer graphics and intelligent techniques to videogame development.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	4					1			5	5	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Techniques and elements that allow intelligent illusion in a videogame.
- The evolution of AI in videogames. Chess as a case study.
- A* and key search algorithms in automatic decision-making.
- Quakebot's, Visibility Matrix.
- Group behavior, packs.
- Composed, reactive, intelligent scenarios.
- Introduction to data networks.
- Virtual spaces vs. real spaces.

1. **Course code number and name:** 300CIS014, Multimedia Technologies.
2. **Credits and contact hours:** 3 credit hours, 3 hours per week.
3. **Course Coordinator:** Pablo Bejarano.

4. **Textbook:**

Multimedia: Making it Work, T. Vaughan, 2004.

Supplemental materials:

- The Computer Music Tutorial, C. Roads, 2000.
- Real-Time Digital Signal Processing, 2nd Edition, S.M. Kuo, B.H. Lee, W. Tian, 2006.
- Managing Multimedia and Unstructured Data in the Oracle Database, M. Kratochvil, 2013.
- Introduction to Data Compression, Fourth Edition, K. Sayood, M. Kaufmann, 2012.
- The Definitive Guide to HTML5 Video, S. Pfeiffer, 2010.

5. **Specific course information:**

Description of the contents:

The objective of this course is to present a variety of distributed multimedia applications, i.e. libraries of network videos, telephony over the Internet and video-conference, which result from the ability of modern computers to handle large amounts of continuous time-dependent data.

Prerequisites: 300CIS004, Objects and Mid-Scale Programming; 300CIS006, Communication Networks.

Type of course: Selected Elective.

6. **Specific goals of the course:**

Learning objectives:

- To define minimum specifications of computational platforms both for development and for using multimedia products.
- To use at least one basic tool for digitalizing and editing sound, images, graphics and video.
- To use a hypermedia application development methodology.
- To develop a multimedia application with design standards and recommendations presented in this course.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2					2		3	5		1

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Multimedia protocols and standards.
- Planning the capacity and performance considerations.
- Input and output devices: scanners, digital cameras, touch-screens, voice activation.
- MIDI keyboards and synthesizers.
- Storage standards.
- Servers and multimedia file systems.
- Tools for multimedia application development.

1. **Course code number and name:** 300CIG030, Information Security.
2. **Credits and contact hours:** 3 credit hours, 3 hours per week.
3. **Course Coordinator:** Siler Amador.

4. **Textbook:**

Web Security, Privacy & Commerce, S. Garfinkel and, G. Spafford, 2001.

Supplemental materials:

- Applied Cryptography, B. Schneier, 1996.
- Security Engineering - A Guide to Building Dependable Distributed Systems, R. Anderson, 2001.
- Secrets and Lies – Digital Security in a Networked World, B. Schneier, 2000.
- Computer Related Risks, P. Neumann, 1994.
- Usability Engineering, J. Nielsen, 1993.
- Network Security: Private Communication in a Public World, C. Kaufman, R. Perlman, M. Speciner, 2002.

5. **Specific course information:**

Description of the contents:

The World Wide Web is the largest heterogeneous computer distributed system ever created. E-mail is critical in most of public and private institutions. Electronic commerce and Internet-based services are milestones of the international economy. Security of all of these systems depends on network infrastructure, operating systems and distributed applications. They are decentralized and chaotic by design and there are huge computer risks every day. This course presents the most serious risks in information security and techniques that allow minimizing them systematically.

Prerequisites: 300MAG031, Discrete Mathematics for Computing; 300CIS006, Communication Networks.

Type of course: Selected Elective.

6. **Specific goals of the course:**

Learning objectives:

- To identify the risks a distributed and multiuser computer system is exposed.
- To identify protocols and technologies that can be used to secure a computer infrastructure.
- To design a continuum attack monitoring strategy.

- To make proper information security risks inventory.
- To design an infrastructure with various security layers.
- To use cryptographic techniques to protect data and messages.
- To identify the security limitations of current technologies.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance	2				2		2	3	1	5	

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- Network attacks and malware: denial of service, flooding, traffic collision, identity theft, DNS hijacking, Trojans, social engineering, domestic attacks (buffer overflows, overruns, rootkits).
- Techniques to secure multiuser and distributed systems.
- Cryptography: private and public keys, digital signatures.
- Authentication and identification schemes (single sign on, Kerberos, LDAP).
- Formal models of information security (capabilities, calculi).
- Secure operating systems.
- Security in emails and web sites.
- Secure electronic commerce.
- Security measures: firewalls, detection of intruders, monitoring file changes, rootkits detection, port-scanning.
- Secure programming.

1. **Course code number and name:** 300CIG031, Mobile Computing and Mobile Agents.
2. **Credits and contact hours:** 3 credit hours, 4 hours per week.
3. **Course Coordinator:** Jairo Andrés Velasco.
4. **Textbook:**

Mobile Communications, F. Mazda, 1997.

Supplemental materials:

- Communicating and Mobile Systems: the PI-calculus, R. Milner, 1999.
- Wireless Communications, T. Rappaport, 2002.

5. **Specific course information:**

This course presents the fundamentals of mobile communication, mobile computing and mobile agents, which may be put into practice by means of the design and development of mobile apps for mobile devices.

Prerequisites: 300CIS006, Communication Networks; 300CIG009 Artificial Intelligence.

Type of course: Selected Elective

6. **Specific goals of the course:**

Learning objectives:

- To identify and apply the main concepts of mobile computing.
- To identify and take advantage of concepts about applications in mobile devices implemented in a cellular network.
- To use at least two kinds of tools for developing mobile applications.
- To recognize the historical advance of mobile devices and the current state of the application development for these devices.
- To identify platforms of mobile devices and their characteristics.
- To develop apps for mobile devices.

Relationship with student outcomes:

	Student Outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Relevance								5	5	5	5

1: low relevance; 2: medium relevance; 3: high relevance.

7. Topics of the course:

- History of mobile devices.
- Computing and mobility.
- Platforms and characteristics.
- Concepts in cellular networks.
- Development environments for mobile devices.
- Usability in mobile applications.
- Platform-specific development (Android, iOS, WP-7, others).
- Mobile agents and applications.
- Cross-platform mobile development.
- Development of the integration project.

Appendix B

Faculty Vitae

The CVs presented here are those of the full-time faculty members that were in charge of the courses evaluated in the program assessment.

Dr. Gloria Inés Álvarez	192
Antal Alexander Buss	194
Michell Andrés Gómez	196
Juan Carlos Martínez	197
Dr. Andrés Navarro	199
Dr. Carlos Alberto Olarte	202
María Constanza Pabón	204
Prof. Camilo Rueda	206
Dr. Gerardo Mauricio Sarria	209
José Eduardo Tofiño	211

1. **Name** Gloria Inés Alvarez

2. **Education**

- Bachelor of Science, Systems and Computing Engineering, Corporación Autónoma Universitaria de Manizales, 1990.
- Master of Science, Computer Science, Universidad de Los Andes, Colombia, 1994.
- Doctor of Philosophy, Pattern Recognition and Artificial Intelligence, Universidad Politécnica de Valencia, 2007.

3. **Academic experience**

- Universidad Autónoma de Manizales, Assistant Professor, 1990 – 1998, full time.
- Pontificia Universidad Javeriana Cali, Lecturer, 1997 – 1999, part time.
- Pontificia Universidad Javeriana Cali, Assistant Professor, 1999 – 2007, full time.
- Pontificia Universidad Javeriana Cali, Associate Professor, 2008 – current, full time.
- Pontificia Universidad Javeriana Cali, Head of the Electronics and Computer Science Department, 2012 – current, full time.

4. **Non-academic experience**

- Central Hidroeléctrica de Caldas, Analyst and Programmer, development of system modules of enterprise billing, 1988 – 1990, full time.
- SX Consultores Ltda., Analyst and Programmer, development of commercial applications in C for the UNIX platform, Bogotá, Colombia, 1991 – 1993, full time.

5. **Certifications or professional registrations**

Professional registration No. 17255118865CLD issued by the National Professional Council of Engineering.

6. **Current membership in professional organizations**

7. **Honors and awards**

Cum Laude Doctor of Philosophy, Universidad Politécnica de Valencia, 2007.

8. **Service activities**

9. **Briefly list the most important publications and presentations from the past five years**

Refereed papers:

- G.I. Alvarez, J.H. Victoria, E. Bravo, P. García. HyRPNI Algorithm and an application to bioinformatics, *Revista de Ingeniería* 33 (2011) 44-52.
- P. García, D. López, J. Ruiz, G. I. Alvarez, From regular expressions to smaller NFAs, *Theoretical Computer Science* 412 (2011) 5802-5807.
- R. Naranjo and G.I. Alvarez. A classifier model for detecting pronunciation errors regarding the Nasa Yuwe language's 32 vowels, *Ingeniería e Investigación* 32 (2012).
- G.I. Alvarez, E. Bravo, D. Linares, J. Vargas, and J. Velasco, Machine learning techniques applied to the cleavage site prediction problem, *Advances in Artificial Intelligence and Its Applications* 8265 (2013) 497-507.
- J. Vargas, J.A. Velasco, G.I. Alvarez, D.L. Linares, E. Bravo, False positive reduction in automatic segmentation system, *Advances in Computational Biology* 232 (2014) 103-108.

Refereed conferences:

- D. Loaiza, C. Oviedo, A. Castillo, A. Portilla, G.I. Alvarez, D. Linares, A.A. Navarro, A video game prototype for speech rehabilitation, 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), Bournemouth, 2013.
- G.I. Álvarez, J.H. Victoria, E. Bravo, P. García. A non-deterministic grammar inference algorithm applied to the cleavage site prediction problem in bioinformatics, 10th International Colloquium on Grammatical Inference (ICGI 2010) LNAI 6339 (2010) 267-270.

10. Briefly list the most recent professional development activities

Seminar: Academic Writing in English, Pontificia Universidad Javeriana Cali, 2011.

1. **Name** Antal Alexander Buss

2. **Education**

- Bachelor of Science, Computer Science, Pontificia Universidad Javeriana Cali, 1999.
- Master of Science, Computer Science, Universidad del Valle, 2008.
- Doctor of Philosophy, Computer Science, Texas A&M University, expected graduation 2014.

3. **Academic experience**

- Pontificia Universidad Javeriana Cali, Research Assistant, AVISPA Research Group, 1997 – 1999, part time.
- Pontificia Universidad Javeriana Cali, Lecturer, 1996 – 2000, part time.
- Pontificia Universidad Javeriana Cali, Instructor, 2000 - 2008, full time.
- Texas A&M University, Research Assistant, 2007 – 2013, full time.
- Pontificia Universidad Javeriana Cali, Assistant Professor, 2008 – current, full time.
- Texas A&M University, Teaching Assistant, 2009, part time.
- Pontificia Universidad Javeriana Cali, Director of the Computer Science Program, 2013 - current, full time.

4. **Non-academic experience**

- IP Total Software, Cali, Software Designer, 1999 – 2000, full time.
- Pontificia Universidad Javeriana Cali, Computer Science Laboratory Administration, 2001 – 2007, part time.

5. **Certifications or professional registrations**

Professional registration No. 76208118500VLL issued by the National Professional Council of Engineering.

6. **Current membership in professional organizations**

- Member of the Association for Computing Machinery (ACM), since 2005.
- Member and Regional Academic Leader, Colombian Community of Advance Computing (3CoA), since 2014.

7. **Honors and awards**

- Honorable mention in B.S. Thesis, Pontificia Universidad Javeriana – Cali, Colombia, 1999.
- Outstanding Professor, Pontificia Universidad Javeriana – Cali, Colombia, 2007.
- Industrial Affiliates Program scholarship, Department of Computer Science, Texas A&M University, 2007 – 2008.
- Colciencias-LASPAU (Fulbright) Scholarship, Colombia/USA, 2007 – 2010.
- Broader Engagement grant recipient, Supercomputing Conference 2008 – 2009.
- AMCS-KAUST Graduate Fellow, Texas A&M University, 01/2012 – 08/2012.

8. Service activities

External reviewer for LCPC, PACT, PPOPP, ICS, IPDPS.

9. Briefly list the most important publications and presentations from the past five years

- G. Tanase, X. Xu, A. Buss, Harshvardhan, I. Papadopoulos, O. Pearce, T. Smith, N. Thomas, M. Bianco, N. M. Amato, L. Rauchwerger, The STAPL pList, 22nd International Conference on Languages and Compilers for Parallel Computing (LCPC09), Newark, 2009.
- A. Buss, A. Fidel, Harshvardhan, T. Smith, G. Tanase, N. Thomas, X. Xu, M. Bianco, N. M. Amato, L. Rauchwerger, The STAPL pView, 23rd International Conference on Languages and Compilers for Parallel Computing (LCPC10), Houston, 2010.
- A. Buss, Harshvardhan, I. Papadopoulos, O. Tkachyshyn, T. Smith, G. Tanase, N. Thomas, X. Xu, M. Bianco, N. M. Amato, L. Rauchwerger, STAPL: Standard template adaptive parallel library, 3rd Annual Haifa Experimental Systems Conference, Haifa, Israel, 2010.
- G. Tanase, A. Buss, A. Fidel, Harshvardhan, I. Papadopoulos, O. Pearce, T. Smith, N. Thomas, X. Xu, N. Mourad, J. Vu, M. Bianco, N. M. Amato, L. Rauchwerger, The STAPL parallel container framework, 16th ACM Symposium on Principles and Practice of Parallel Programming (PPOPP11), San Antonio, 2011.

10. Briefly list the most recent professional development activities

1. **Name** MichellAndrés Gómez
2. **Education**
 - Bachelor of Science, Mathematics, Universidad Del Valle, 2002.
 - Master of Science, Mathematics, Universidad Del Valle, 2005.
3. **Academic experience**
 - Pontificia Universidad Javeriana Cali, Lecturer, 2009, part time.
 - Universidad Icesi, Colombia, Lecturer, 2005 – 2009, part time.
 - Pontificia Universidad Javeriana – Cali, Colombia, Assistant Professor, 2010 - current, full time.
4. **Non-academic experience**
5. **Certifications or professional registrations**
6. **Current membership in professional organizations**
7. **Honors and awards**
8. **Service activities**
9. **Briefly list the most important publications and presentations from the past five years**
 - M. Gómez, G. Restrepo, Tensor product of radian measures, *Matemáticas Enseñanza Universitaria* 17 (2009) 23 – 34.
 - M. Gómez, L. Solarte. On the topology of the tangent bundle to the sphere, *Matemáticas Enseñanza Universitaria* 19 (2011) 49 – 53.
10. **Briefly list the most recent professional development activities**
 - Certificate in Learning-Service, Pontificia Universidad Javeriana Cali, 2011.
 - Certificate in Biodiversity, Pontificia Universidad Javeriana Cali, 2011.
 - Seminar, First show of best practices in pedagogy, Pontificia Universidad Javeriana Cali, 2012
 - Certificate in how to generate manuals and university textbooks, Pontificia Universidad Javeriana Cali, 2013.

1. **Name** Juan Carlos Martinez

2. **Education – degree, discipline, institution, year**

- Bachelor of Science, Systems Engineering, Universidad Piloto de Colombia, 1990.
- Specialist, Business Administration, Universidad Icesi, 2000.
- Master of Engineering, Pontificia Universidad Javeriana Cali, 2012.

3. **Academic experience**

- Pontificia Universidad Javeriana Cali, Instructor, 2007 – 2009, part time.
- Pontificia Universidad Javeriana Cali, Assistant Professor, 2010 - current, full time.

4. **Non-academic experience**

- EMSIRVA E.S.P., Professional in the informatics department in charge of designing and developing information systems, 1990 – 2001, full time.
- EMSIRVA E.S.P., Deputy commercial manager in charge of planning, projecting and controlling strategies for optimal customer service, 2001 – 2004, full time
- L.A.M. Asesorías, Consultant and project manager in charge of consulting public services, informatics, billing and portfolio, 2004 – 2007, full time.

5. **Certifications or professional registrations**

Professional registration No. 25256160278 issued by the National Professional Council of Engineering.

6. **Current membership in professional organizations**

7. **Honors and awards**

- Outstanding Professor, Pontificia Universidad Javeriana Cali, 2009.
- SEI-Certified PSP Developer, Carnegie Mellon University - SEI Certification, 2013.

8. **Service activities**

9. **Briefly list the most important publications and presentations from the past five years**

- J.C. Martinez, G. Sarria, Didactic and interdisciplinary experiences in a Software Engineering course, 43rd Frontiers in Education Conference, Oklahoma City, 2013.

- K. S. Lopez, J.C., Martínez, Writing for learning and communicating in the course Software Engineering Processes, 5th International and 4th National Meeting for Reading and Writing in Higher Education, Bucaramanga, Colombia, 2014.

10. Briefly list the most recent professional development activities

- Certificate in writing scientific papers, Pontificia Universidad Javeriana Cali, 2011.
- Seminar, Teaching in the Pontificia Universidad Javeriana Cali, 2012.
- Certificate in reading and writing for learning, Pontificia Universidad Javeriana Cali, 2012.
- Personal Software Process Fundamentals, Software Engineering Institute, 2013
- Personal Software Process Advanced, Software Engineering Institute, 2013
- Mentoring formation online about best practices of software managing, Kybele Consulting, 2013.

1. **Name** Andrés Adolfo Navarro

2. **Education – degree, discipline, institution, year**

- Bachelor of Science, Computer Science, Pontificia Universidad Javeriana Cali, 1994.
- Specialist, Networks and Communications, Universidad Icesi, 2001.
- Master Of Science, Computer Graphics and Virtual Environments, University Of Hull, 1998.
- Doctor Of Philosophy, Computer Science, University Of Otago, New Zealand, 2010.

3. **Academic experience**

- Pontificia Universidad Javeriana – Cali, Colombia, Instructor, 1994 – 1996, part time.
- Pontificia Universidad Javeriana – Cali, Colombia, Assistant Professor, 1998 – 2010, full time.
- Universidad Santiago de Cali, Colombia, Instructor, 2000 – 2003, part time.
- Universidad Icesi, Colombia, Instructor, 2004, part time.
- Pontificia Universidad Javeriana – Cali, Colombia, Associate Profesor, since 2011, full time.

4. **Non-academic experience**

- Coltec Ingeniería Ltda, 1999 – 2000, part time.
- Grupo de Trabajo Avalon, 2000 – 2001, part time.
- Centro de Telemedicina de Colombia Ltda., Cco-founder, 2002 – 2007, part time.

5. **Certifications or professional registrations**

Professional registration No. 7620854387VLL issued by the National Professional Council of Engineering.

6. **Current membership in professional organizations**

- Member of the Association for Computing Machinery (ACM).
- Member of the Microsoft Faculty Connection.

7. **Honors and awards**

- Honorable mention, Pontificia Universidad Javeriana – Cali, Colombia, 1994

- Best In Show Prize, Telemedicine and eHealth Forum, The Royal Society Of Medicine, 2006
- Silver Medal for 15 years of service, Pontificia Universidad Javeriana Cali, 2010
- Outstanding Professor, Pontificia Universidad Javeriana Cali, 2012

8. Service activities

9. Briefly list the most important publications and presentations from the past five years

Refereed papers:

- S. Ochoa, J.A. Aguilar, A. Navarro, A. Jaramillo, L. Henao, Design o fan education scenario for museums using TRIZ and ACT, Pensamiento Psicológico 11 (2013) 71-88.
- I.M. Sánchez, A.A. Navarro, Cultural Communication and TICs: The accesible representation of the Chimú culture, Historia y Comunicación Social, 8 (2013) 541-554.

Refereed conferences:

- D. Loaiza, C. Oviedo, A. Castillo, A. Portilla, G. Álvarez, D. Linares, and A. Navarro, A videogame prototype for speech rehabilitation, 5th International Conference on Games and Virtual Worlds for Serious Applications, Bournemouth, 2013.
- C.C. Ventes, A.A. Navarro, D.A. Velasco, and E.C. Prakash, VirtuaOM: tangible human-computer interface for collaborative applications, 7th Annual International Conference on Computer Games, Multimedia and Allied Technologies (CGAT 214), Singapore, 2014.

10. Briefly list the most recent professional development activities

- Certificate in Network Security, Pontificia Universidad Javeriana – Cali, Colombia, 2007.
- Certificate in Software Development in .NET, Pontificia Universidad Javeriana – Cali, Colombia, 2007
- Preparing for academic careers, University of Otago, New Zealand, 2008.
- How to mobilize international grants, Escuela De Ingeniería De Antioquia, 2010.
- Web 2.0 Technologies for Teaching, Universidad Autónoma De Bucaramanga, 2011.
- Post-doctoral research, Technological representation of complex cultural contents, Universidad Complutense, Madrid, since 2012.

- Microsoft Faculty Summit, Redmond, 2014
- Microsoft Latin American Faculty Summit, Viña del Mar, Chile, 2014

1. **Name** Carlos Alberto Olarte

2. **Education**

- Bachelor of Science, Computer Science and Engineering, Pontificia Universidad Javeriana Cali, 2002.
- Doctor of Philosophy, Computer Science, École Polytechnique of Paris, 2009.

3. **Academic experience**

- Pontificia Universidad Javeriana Cali, Assistant Professor, 2002 - 2009, full time.
- Pontificia Universidad Javeriana Cali, Associate Professor, 2009 – current, full time.

4. **Non-academic experience**

IP Total Software S.A., Development Chief, 2000 – 2002, full time.

5. **Certifications or professional registrations**

Professional registration No. 76208118506VLL issued by the National Professional Council of Engineering

6. **Current membership in professional organizations**

Member of the Association of Computing Machines - ACM.

7. **Honors and awards**

- Magna-Cum-Laude Award, Pontificia Universidad Javeriana, 2002.
- ACOFI (Colombian Association of Engineering Schools) award to the best projects in engineering for the TELEMEC project, 2002.

8. **Service activities**

9. **Publications and presentations from the past five years**

Refereed papers

- H.A. López, C. Olarte, J.A. Pérez: Towards a unified framework for declarative structured communications, *Elect. Proc. Theor. Computer Sci.* 17 (2009) 1-15.
- D. Chiarugi, M. Falaschi, D. Hermith, M. Guzman, C. Olarte, Simulating signalling pathways with BioWayS, *Electr. Notes Theor. Comput. Sci.* 293 (2013) 17-34.
- M. Falaschi, C. Olarte, C. Palamidessi: Abstract interpretation of temporal concurrent constraint programs. *Theor. Pract. Logic Program.* Available on CJO2014.

- C. Olarte, V. Nigam, E. Pimentel, Dynamic spaces in concurrent constraint programming, *Electr. Notes Theor. Comput. Sci.* 305 (2014) 103-121.
- E. Pimentel, C. Olarte, V. Nigam, A proof theoretic study of soft concurrent constraint programming, *Theor. Pract. Logic Program.* 14 (2014) 649-663.

Refereed conferences

- C. Olarte, F.D. Valencia, Universal concurrent constraint programming: symbolic semantics and applications to security, *SAC* (2008) 145-150.
- C. Olarte, F.D. Valencia, The expressivity of universal timed CCP: undecidability of monadic FLTL and closure operators for security, *PPDP* (2008) 8-19.
- M. Falaschi, C. Olarte, C. Palamidessi, A framework for abstract interpretation of timed concurrent constraint programs, *PPDP* (2009) 207-218.
- D. Chiarugi, M. Falaschi, C. Olarte, C. Palamidessi, Compositional modelling of signalling pathways in timed concurrent constraint programming, *BCB* (2010) 414-417.
- D. Hermith, C. Olarte, C. Rueda, F.D. Valencia, Modeling cellular signaling systems: an abstraction-refinement approach, *PACBB* (2011) 321-328.
- C. Olarte, E. Pimentel, C. Rueda, N. Cataño, A linear concurrent constraint approach for the automatic verification of access permissions. *PPDP* (2012) 207-216.
- V. Nigam, C. Olarte, E. Pimentel, A general proof system for modalities in concurrent constraint programming, *CONCUR 2013, Lecture Notes in Computer Science* 8052 (2013) 410-424.
- D. Chiarugi, M. Falaschi, D. Hermith, R. Marangoni, C. Olarte, Stochastic modelling of non Markovian dynamics in biochemical reactions, *IWBBIO* (2013) 537-544.
- C. Olarte, C. Rueda, F.D. Valencia, Models and emerging trends of concurrent constraint programming, *Constraints* 18 (2013) 535-578.

10. Most recent professional development activities

Dr. Olarte has been invited researcher at LIX Ecole Polytechnique for the past 4 years. He is also Program Committee member of the ACM's Symposium on Applied Computing (SAC) and the International Workshop on Functional and (Constraint) Logic Programming (WFLP'14).

1. **Name** María Constanza Pabón.

2. **Education**

- Bachelor of Science, Computer Science, Pontificia Universidad Javeriana Cali, 1993.
- Master, Business Administration, Universidad Del Valle, 2000.

3. **Academic experience**

Pontificia Universidad Javeriana Cali, Assistant Professor, 2001 - current, full time.

4. **Non-academic experience**

- Transportes Expreso Palmira. Analyst programmer in charge of developing information systems, installing and maintaining software and hardware, 1992 – 1995, full time.
- Financiera FES, Project coordinator in charge of developing information systems and assisting users, 1995 – 2001, full time.

5. **Certifications or professional registrations**

Professional registration No. 7620851095VLL issued by the National Professional Council of Engineering.

6. **Current membership in professional organizations**

- Member of the Colombian Computing Society (SCo2).
- Member of the Association for Computing Machinery (ACM).

7. **Honors and awards**

- Andrés Bello Award, Ministry of Education, 1987.
- Bachiller–Ecopetrol, Ecopetrol, Colombia, 1987.
- Honor Mention, Pontificia Universidad Javeriana Cali, 1993.

8. **Service activities**

9. **Briefly list the most important publications and presentations from the past five years**

- M.C. Pabón, G. Montoya, M. Millán, Mediation and graph data models for medical data integration, 39th Latin American Computing Conference, Naiguatá, Venezuela, 2013.

- M.C. Pabón, M. Millán, A data integrating architecture as a base for a medical data retrieving conceptual language, 37th Latin American Computing Conference, Quito, 2011.
- M.C. Pabón, C. Roncancio, M. Millán, Graph data transformations and querying, International C* Conference on Computer Science & Software Engineering (C3S2E '14), Montreal, 2014.
- M.C. Pabón, C. Roncancio, M. Millán, Graph management to improve querying of health and social data, International Conference on Health Informatics (HealthInf), Angers, Loire Valley, France, 2014.

10. **Briefly list the most recent professional development activities**

Certificate in Academic Writing In English, Pontificia Universidad Javeriana Cali, 2012.

1. **Name** Camilo Rueda

2. **Education**

- Bachelor of Science, Systems and Computing Engineering, Universidad de Los Andes, 1970
- Master of Science, Computer Science, MIT, 1975
- Doctor Electrical Engineering, Computer Science, MIT, 1977

3. **Academic experience**

- Universidad de Los Andes, Instructor, 1978-1980, full time.
- Universidad Autónoma de Manizales, Dean of Engineering, 1983-1990, full time.
- Institut de Recherche/Coordination Acoustique-Musique Paris, researcher, 1991-1994, full time.
- Pontificia Universidad Javeriana Cali, Associate Professor, 1994-1997, full time.
- Institut de Recherche/Coordination Acoustique-Musique Paris, researcher, 1998-1998, full time.
- Pontificia Universidad Javeriana Cali, Director of the Computer Science Program, 1999-2004, full time.
- Institut de Recherche/Coordination Acoustique-Musique Paris, researcher, 2004-2005, full time.
- Pontificia Universidad Javeriana Cali, Head of the Department of Science and Engineering of Computing, 2005-2011, full time.
- Pontificia Universidad Javeriana Cali, Professor, 2011-2013, full time.
- Pontificia Universidad Javeriana Cali, Emeritus Professor, 2014 - current, half time.

4. **Non-academic experience**

- Wang Laboratories, Lowell-Massachusetts, software developer. Design of the file system and text editor for a multi-pc network, 1977-1978. Full time.
- Planeación Distrital, Bogotá, software developer. Implementing of a model of agriculture productivity in Colombia. 1980, part time.

5. **Certifications or professional registrations**

6. **Current membership in professional organizations**

Member of the French Music Informatics Association - AFIM.

7. Honors and awards

8. Service activities

9. Publications and presentations from the past five years

Refereed papers and chapters in books:

- C. Olarte, C. Rueda, F.D. Valencia, Concurrent Constraints Calculi: A Declarative Paradigm for Modeling Music Systems, in G. Assayag (Ed.), *New Computational Paradigms For Computer Music*, 2009.
- N. Cataño, F. Barraza, D. García, P. Ortega, C. Rueda: A case study in JML-assisted software development, *Electr. Notes Theor. Comput. Sci.* 240 (2009) 5-21.
- C. Rueda, C. Olarte, F.D. Valencia, G. Sarria, M. Toro, Concurrent Constraints Models of Music Interaction, in Truchet, G. Assayag (Eds.), *Constraint Programming in Music*, 2011.
- C. Olarte, C. Rueda, F.D. Valencia, Models and emerging trends of concurrent constraint programming, *Constraints* 18 (2013) 535-578.
- M. Toro, M. Desainte-Catherine, C. Rueda, Formal semantics for interactive music scores: a framework to design, specify properties and execute interactive scenarios, *J. Mathematics and Music*, 8 (2014) 93-112.
- N. Cataño, C. Rueda, V. Zúñiga, T. Wahls, Translating B and Event-B Machines to Java and JML, in J. Boulanger (Ed.), *Formal Methods Applied to Industrial Complex Systems: Implementation of the B Method*, 2014.

Refereed conferences:

- J. Aranda, G. Assayag, C. Olarte, J.A. Pérez, C. Rueda, M. Toro, F.D. Valencia, An overview of FORCES: an INRIA project on declarative formalisms for emergent systems, *ICLP* (2009) 509-513.
- N. Cataño, C. Rueda, Teaching formal methods for the unconquered territory, *TFM* (2009) 2-19.
- N. Cataño, C. Rueda, Matelas: A predicate calculus common formal definition for social networking, *ASM* (2010) 259-272.
- C. Olarte, E. Pimentel, C. Rueda, N. Cataño, A linear concurrent constraint approach for the automatic verification of access permissions. *PPDP* (2012) 207-216.
- N. Cataño, T. Wahls, C. Rueda, V. Rivera, D. Yu: Translating B machines to JML specifications, *SAC* (2012) 1271-1277.
- N. Cataño, S. Hanvey, C. Rueda, Poporo: A Formal methods tool for fast-checking of social network privacy policies, *TOOLS* 50 (2012) 9-16.

- D. Hermith, C. Olarte, C. Rueda, F.D. Valencia, Modeling cellular signaling systems: an abstraction-refinement approach, PACBB (2011) 321-328.

10. **Most recent professional development activities**

- Invited researcher, Université de Bordeaux I, France, 2011 and 2012.
- Invited researcher, Universidade de Madeira, Portugal, 2010 and 2011.
- Invited researcher, IRCAM-Paris, 2012.
- Invited professor, Université Pierre-Marie Curie, Paris, 2013.
- Invited researcher, Ecole Polytechnique-Paris, 2012 and 2013.

1. **Name** Gerardo Mauricio Sarria

2. **Education**

- Bachelor of Science, Computer Science and Engineering, Pontificia Universidad Javeriana Cali, 2001.
- Doctor in Engineering, Computer Science, Universidad del Valle, 2008.

3. **Academic experience**

- Universidad del Valle, Teaching assistant, 2001 - 2002, part time.
- Institute de Recherche et Coordination Acoustique/Musique – IRCAM, Paris, Researcher and Developer, 2003 – 2004, full time.
- Pontificia Universidad Javeriana Cali, Research assistant, 2004, full time.
- Pontificia Universidad Javeriana Cali, Instructor, 2005 - 2008, full time.
- Pontificia Universidad Javeriana Cali, Assistant Professor, 2009 - 2013, full time.
- Pontificia Universidad Javeriana Cali, Associate Professor, 2014 - current, full time.

4. **Non-academic experience**

IP Total Software, Cali, DBA and Network Administrator. 1999 – 2000, full time.

5. **Certifications or professional registrations**

Professional registration No. 76208121241VLL issued by the National Professional Council of Engineering.

6. **Current membership in professional organizations**

- Member of the Association for Computing Machinery (ACM).
- Member of the Colombian Computing Society (SCo2).

7. **Honors and awards**

Distinguished Professor, Pontificia Universidad Javeriana, 2008.

8. **Service activities**

9. **Publications and presentations from the past five years**

Refereed papers and chapters in books:

- C. Olarte, C. Rueda, G. Sarria, M. Toro, F. Valencia, Concurrent constraint models of music interaction, *Constraint Programming in Music*, pp. 133-153, 2011.

- G. Sarria, Application of real-time concurrent constraint calculus, Intelligent Automation and Systems Engineering, Lecture Notes in Electrical Engineering 103 (2011) 379-391.
- G. Sarria, Real-time concurrent constraint calculus - the complete operational semantics, Engineering Letters 19 (2011) 38-45.
- G. Sarria, A Survey of computational calculi used in musical applications, Ingeniería y Competitividad 15 (2013) 103-114.

Refereed conferences:

- G. Sarria, Introduction to programming for engineers following the parachute paradigm, 39th IEEE/ASEE Frontiers in Education Conference (FiE2009), San Antonio, 2009.
- G. Sarria, S. Perchy, Dissonances: brief description and its computational representation in the Rtcc calculus, 6th Sound and Music Computing Conference (SMC2009), Porto, Portugal, 2009.
- G. Sarria, Improving the real-time concurrent constraint calculus with a delay declaration, International Conference on Computer Science and Applications (ICCSA2010), held as part of the World Congress on Engineering and Computer Science (WCECS2010), San Francisco, 2010.
- G. Sarria, An interpreter for the Rtcc calculus, in 7th Colombian Computing Congress (7CCC), 38th Latinamerican Conference on Informatics (CLEI2012), Medellín, 2012.
- G. Sarria, S. Perchy, Building models of musical compositions using process algebras, 39th Latinamerican Conference on Informatics (CLEI2013), Naiguata, Venezuela, 2013.
- G. Sarria, J.C. Martinez, Didactic and interdisciplinary experiences in a Software Engineering course, 43rd IEEE/ASEE Frontiers in Education Conference (FiE2013), Oklahoma City, 2013.

10. Most recent professional development activities

Seminar-Workshop, Scientific Writing in English, Pontificia Universidad Javeriana Cali, 2011.

1. **Name** Jose Eduardo Tofiño.

2. **Education**

- Bachelor of Science, Mathematics, Universidad Del Valle, 1986.
- Specialist, Information Systems, Universidad Del Valle, 1996.
- Master, Education, Pontificia Universidad Javeriana Cali, 2001.
- Master of Science, Mathematics, Universidad Del Valle, 2006.

3. **Academic experience**

- Universidad del Valle, Colombia, Instructor, 1986 – 1995, part time.
- Universidad Icesi, Instructor, 1989 – 1990, part time.
- Pontificia Universidad Javeriana Cali, Instructor, 1989 - 2001, full time.
- Universidad Autónoma de Occidente, Instructor, 1992 – 1994, part time.
- Universidad Nacional de Colombia Sede Palmira, Assistant Professor, 2001 – 2002, full time.
- Pontificia Universidad Javeriana Cali, Assistant Professor, 2002 – 2012, full time.
- Pontificia Universidad Javeriana Cali, Associate Professor, 2012 - current, full time.
- Pontificia Universidad Javeriana Cali, Director of the Applied Mathematics Program, 2009 – 2013, full time.

4. **Non-academic experience**

5. **Certifications or professional registrations**

6. **Current membership in professional organizations**

7. **Honors and awards**

Mention of Honor to Masters Thesis, Pontificia Universidad Javeriana Cali, 2001.

8. **Service activities**

9. **Briefly list the most important publications and presentations from the past five years**

- J.E. Tofiño, The well ordering of sets: a historical and epistemological view, 17th Colombian Congress of Mathematics, Cali, 2009.
- J.E. Tofiño, M. Gómez, D. Suescún, Hamming method for calculating the nuclear reactivity, International Conference on Applied Mathematics and Informatics (ICAMI10), San Andrés, Colombia, 2010.

- J.E. Tofiño, Historical-epistemological implications of well ordering of sets in the historic development of mathematics (in Spanish), 4th National School of History and Epistemology of Mathematics (ENHEM4), Cali, 2013.

10. Briefly list the most recent professional development activities

- Seminar, Trends and Issues in Higher Education, Paris, France, 2009.
- Research internship, Universidad de Sevilla, Spain, 2010.
- Short course, School of Applied Mathematics and Innovation: Celestial Mechanics and Computing Orbits (SAMI 2011), Pontificia Universidad Javeriana Cali and Universidad Sergio Arboleda – Bogotá, Santa Marta, 2011.
- Short course, Topological Methods in Ordinary Differential Equations, First School of Applied Mathematics (EMA I), Pontificia Universidad Javeriana Cali, 2012.
- Seminar, First show of best practices in pedagogy, Pontificia Universidad Javeriana Cali, 2012.

Appendix C

Equipment

The following is the inventory of equipment and software used in the laboratories of the Computer Science program.

C.1 Computer Science Laboratory

Equipment

- 3 iMac computers: processor Intel Core I5; 4 Gb in RAM; 500-Gb hard drive.
- 7 Dell XPS 8300 computers: processor Intel Core I5; 3 Gb in RAM; 500-Gb hard drive.
- 11 Dell Optiplex 960 computers: processor Intel Core 2 Quad; 3 Gb in RAM; 500-Gb hard drive.
- 10 Hewlett-Packard EliteDesk computers: processor Intel Core I7; 8 Gb in RAM; 500-Gb hard drive.
- 2 Logitech C930E webcams.
- 1 Handy Cam video camera.
- 12 iPad 2 tablets.
- 4 Samsung Galaxy Note tablets.
- 1 Samsung Galaxy Tab 2 tablet.
- 1 tripod.
- 2 Wii videogame controllers.
- 1 Videomic directional video condenser microphone.
- 1 portable speaker.

- 1 Kinect interaction device.
- 1 Pen tablet series Genius interaction device.

Software

- Under Windows 7 and Windows 8.1: OpenMusic; MongoDB; SQL Developer; PgAdmin; XAMPP; WampServer; C, C++; Python; Java; Ruby; SWI-Prolog; Coq; Microsoft Virtual C# 2010 Express; Microsoft Visual Studio Professional 2013; Microsoft XNA Game Studio 4.0; NetBeans IDE; Eclipse IDE; Cisco Packet Tracer; Lego Mindstorms nxt; Oracle VM VirtualBox; LibreOffice; Paint.net; Unity; Makehuman; Inkscape; Git; Dia; Audacity; GIMP; Blender; Qt SDK; Qt Creator; and Processing.
- Under Linux Ubuntu 14.04: C, C++, Python, Java; Ruby; SWI-Prolog; Coq; Lisp; Rodin; Isabelle; NetBeans IDE; Eclipse IDE; Oracle VM VirtualBox; LibreOffice; Inkscape; Git; Dia; Audacity; GIMP; Blender; Emacs; PgAdmin; Qt Creator; Latex; and Data Display Debugger.
- Under OS X Mavericks: Python; Rodin; Isabelle; Xcode; Oracle VM VirtualBox; LibreOffice; Dia; Audacity; GIMP; Blender; NetBeans IDE; Qt Creator; Processing; Latex; PgAdmin; Unity; XAMPP; OpenMusic; Pure Data; and Android SDK.
- Servers use Microsoft SQL Server 2008 R2; PostGre SQL; and Oracle Database 11g Express Edition.

C.2 Networks Laboratory

Equipment

- 6 iMac computers: processor Intel Core 2 Duo; 4 Gb in RAM; 500-Gb hard drive.
- 1 Dell Allienwre computer: processor Intel Core I7; 12 Gb in RAM; 1-Tb hard drive.

Software The following licenses run under OS X Mavericks: XCode, OpenMusic, Pure Data; Reactivision; Audacity; Blender; and Android SDK.

C.3 Interaction and Sound Laboratory

Equipment

- 13 Dell Optiplex 520 computers: processor Intel Pentium 4 at 800 MHz.
- 8 Dell Optiplex 755: processor Intel Core 2 Duo E4500 at 2.2 GHz.
- 4 Hewlett-Packard Compaq MXL30103 computers.

- 2 Cisco 1721 routers.
- 3 Cisco 1841 routers.
- 4 Cisco 2501 routers.
- 1 Cisco 2514 router.
- 3 Cisco 2611XM routers.
- 6 2811W/AC PWR routers.
- 5 Linksys WRT300N wireless routers.
- 1 Cisco Aironet 300 Series AIR-WGB352R wireless router.
- 1 Cisco 1900 Series 24 ports switch.
- 12 Cisco Catalyst 2950 24 ports switch.
- 6 Cisco Catalyst 2960 24 ports switch.
- 3 WS-C424M fast hub.
- 3 ASA 5505 Firewall NGN security service.
- 2 Firewall PIX515E security appliances.
- 2 Fortinet-Fortigate 1000A next Generation Firewall.

Software Access is free to the following software: Packet Tracert 6.2; Cisco TFTP Server; Putty; Eagle CadSoft; PsPice; CISCO VPN Client; xLight FTP Server; CCP Ver 2; NMAP; WireShark; Capsa Soft Tools.

Appendix D

Institutional Summary

D.1 The Institution

- a) Name and address of the institution:

Pontificia Universidad Javeriana Cali, Calle 18 No. 118-250, Av. Caasgordas, Cali, Colombia

- b) Name and title of the chief executive officer of the institution

Luis Felipe Gmez, S.J., Rector.

- c) Name and title of the persons submitting the Self-Study Report.

Antal Alexander Buss, Director of the Computer Science Program.

Prof. Jorge Francisco Estela, Professor of Thermodynamics, Coordinator of Accreditation of the School of Engineering.

- d) Name the organizations by which the institution is now accredited, and the dates of the initial and most recent accreditation evaluations.

On the 6th of March 2012 the National Council of Accreditation awarded the University with the institutional accreditation with validity of eight years.

D.2 Type of Control

This is a private non-profit higher-education institution founded and run by the Society of Jesus.

D.3 Educational Unit

The School of Engineering is organized as Figure D.1 shows.

The highest collegiate authority of the School is the Council of the School, which is integrated by the Dean of the School, the Heads of Department, Program Directors, two faculty members elected by their peers, two students elected by their fellow students, and the Secretary of the School. The Dean of the School is the highest academic and administrative authority of the School, to whom all the academic directors and administrative staff are accountable.

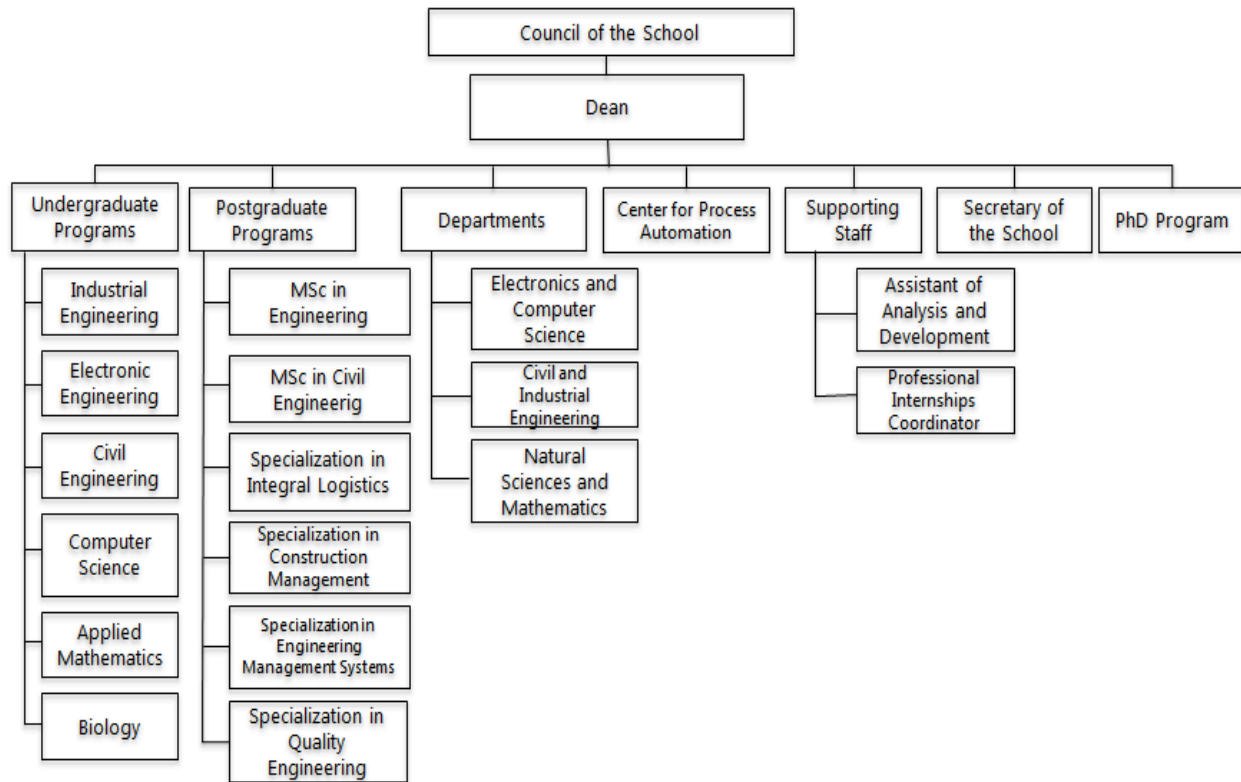


Figure D.1: Organization chart of the School of Engineering

As it was indicated in Section 6.A of Criterion 6, Departments are academic units composed of faculty members who provide teaching to the academic programs and carry out research and service. For the Computer Science Program, the faculty members supporting the core-curriculum courses are ascribed to the Electronics and Computer Science Department. Each Department is headed by a Head of Department appointed by the Academic Vice-rector upon the results of a consultation carried out by the Dean amongst faculty members. Heads of Department are appointed for a term of three years, but can be extended for another term.

Academic programs (undergraduate, postgraduate, and the PhD program) are comprised of enrolled students and administer the curriculum. Each undergraduate program is headed by a Program Director; postgraduate programs are headed by one Program Director, and the PhD program (which is starting in the second semester of 2015) is headed by another Program Director. Program Directors are appointed by the Academic Vice-rector from candidates recommended by the Dean. As with Heads of Department, Program Directors are appointed for a term of three years, but can be extended for another term.

In Colombia, Specializations are postgraduate programs, usually of duration of three semesters, but their curricula do not include a thesis, as is the case with the traditional MSc programs.

As of January of 2015, The School of Engineering comprised 1572 undergraduate students, 229 postgraduate students, 84 full-time faculty members, 105 part-time faculty members, and an administrative staff of 20 persons.

The chain of responsibility down from the chief executive officer to the Director of the Computer Science Program is:

- Luis Felipe Gmez, S.J.: Rector of the University.
- Ana Milena Yoshioka: Academic Vice-rector.
- Prof. Jaime Alberto Aguilar: Dean of the School of Engineering.
- Antal Alexander Buss: Director of the Computer Science Program.

As said above, the Program Director is appointed by the Academic Vice-rector but is accountable to the Dean. On the other hand, the Dean is appointed by the Rector but is accountable to the Academic Vice-rector.

D.4 Academic Support Units

The following are the names and positions of the individuals responsible for the academic units that teach courses required by the Computer Science Program:

Departments of the School of Engineering:

- Dr. Diego Daro Prez: Head of the Civil and Industrial Engineering Department.
- Dr. Gloria Ins lvarez: Head of the Electronics and Computer Science.
- Daniel Enrique Gonzlez: Head of the Natural Sciences and Mathematics Department.

Departments of the School of Humanities and Social Sciences:

- Manuel Sevilla: Head of the Humanities Department.
- Ricardo Rodriguez: Head of the Communication and Language Department.
- Luis Edison Bertn: Head of the Legal and Political Sciences Department.

Departments of the School of Economic and Administrative Sciences:

- Alina Gmez: Head of the Accounting and Finance Department.
- Julio Csar Paz: Head of the Organizational Management Department.

D.5 Non-academic Support Units

The following are the names and positions of the individuals responsible for the units that provide non-academic support to the Computer Science Program:

Rectory:

- Luis Felipe Gmez, S.J.: Rector of the University.
- Pablo Rubn Vernaza: Secretary General.
- Enrique Alberto Quintana: Director Legal Office.
- Mauricio Corts: Director Office of Planning.
- Clara Ins Mojica: Director of Alumni Relationships.
- Mara Luca Lloreda: Director of University Industry Relationships.
- Claudia Mara Castao: Director of International Relationships.
- Juan Carlos Prado: Director of Communications.
- Claudia Luca Mora: Director of Social Responsibility.
- Jaime Alberto Reinoso: Director of Computer Services.
- Iris Cabra: Director of Multimedia.

Academic Vice-rectory:

- Ana Milena Yoshioka: Academic Vice-rector.
- Fabín Ramírez: Director of Academic Development.
- Ximena Botero: Director of Faculty Development.
- Beatriz Helena Giraldo: Director of Student Development.
- Dr. Diego Luis Linares: Director of Research, Development and Innovation.
- Liliana Martínez: Registrar and Admissions.
- Andrés Felipe Echavarrá: Director of the Library.
- Oscar Mauricio Vsquez: Coordinator of Entrepreneurship.
- Adriana Reina: Director of Marketing.
- Julio César Grisales: Director of the Center of Consulting and Continued Education.
- Tatiana Valencia: Director Technologies of Information and Communication.
- Manuel Ramiro Muñoz: Director of Center of Intercultural Studies.

University Medium Vice-rectory:

- Luis Fernando Granados, S.J.: University Medium Vice-rector.
- Ricardo Caicedo: Director of the Center for Cultural Expression.
- Martha Cecilia Gómez: Director Sports Center.
- Liliana Tamayo: Director Helath Center.
- Carlos Alberto Romero, S.J.: Director Center Saint Francis Xavier.

Administrative Vice-rectory:

- Juan Vianey Gómez: Administrative Vice-rector.
- James Orlando Perea: Chief of Accounting and Budget.
- Mara Mercedes Escudero: Treasurer.
- Alba Doris Morales: Chief of Human Resources.
- Adriana Paredes: Chief of Student Loans.
- Mario Torres: Chief of Purchasing and Special Services.
- Harold Nates: Chief of Operational Services.

D.6 Credit Unit

Here we reproduce the definition of the credit hour given in Section 5.A.1 of Criterion 5: one credit hour equals 48 hours of study in a semester of 16 weeks of classes, with a distribution of two hours of independent study per hour of class work. Therefore, nominally, one credit hour corresponds to a course of one hour per week of class work.

D.7 Tables

Complete the following tables for the program undergoing evaluation.

D.7.1 Program Enrollment and Degree Data

Name of the Program: Computer Science

	Academic Year	Enrollment Year					Undergrad Total	Grad Total	Degree Awarded			
		1st	2nd	3rd	4th	5th			Associates	Bachelors	Masters	Doctorates
Current Year	FT	43	18	21	16	18	116			10		
	PT											
1	FT	29	21	19	22	19	110			3		
	PT											
2	FT	35	24	23	22	19	123			8		
	PT											
3	FT	29	21	28	21	22	121			0		
	PT											
4	FT	22	36	24	18	26	126			14		
	PT											

Table D.1: Program Enrollment and Degree Data

Give official fall term enrollment figures (head count) for the current and preceding four academic years and undergraduate and graduate degrees conferred during each of those years. The “current” year means the academic year preceding the on-site visit.

FT–full time

PT–part time

D.7.2 Personnel

Name of the Program: Computer Science

The faculty members counted in Table D.2 are ascribed to the Electronics and Computer Science Department for the support of the engineering-topic core-curriculum courses of the

Computer Science Program.

Year¹: Second semester of 2014

	HEAD COUNT		FTE ²
	FT	PT	
Administrative ²	1		1
Faculty (tenured)	1		1
Faculty (tenure-track) ³	5		4
Other Faculty (non tenure-track)	4	11	8
Student Teaching Assistants ⁴		10	5
Technicians/Specialists	1		1
Office/Clerical Employees	1		1
Others ⁵			

Table D.2: Personnel

Report data for the program being evaluated.

1. Data on this table should be for the fall term immediately preceding the visit. Updated tables for the fall term when the ABET team is visiting are to be prepared and presented to the team when they arrive.
2. The only person holding a joint administrative/faculty position is the Program Director, who is also counted as a tenure-track faculty member in Table 6.1.
3. The full-time workload is 40 hours/week with a nominal workload of teaching of 12 hours/week for faculty members who do not participate in research, which is used to calculate the full-time equivalents of part-time faculty members.
4. The full-time workload of student teaching assistants is 10 hours/week per course section. The figures given in the Table D.2 are for discipline-related core-curriculum courses from the Electronics and Computer Science Department.
5. Specify any other category considered appropriate, or leave blank.

Appendix E

The score system

These scores are given in numbers 92 to 94 of the Faculty Regulations:

Assessment of performance: Excellent: 30 points; Good: 20 points; Medium or bad: no points.

Assessment of academic production:

- Scientific, technical or humanistic works:
 - Book: up to 150 points.
 - University text: up to 100 points.
 - School text: up to 70 points.
 - University handbook: up to 50 points.
 - School handbook: up to 30 points.
 - Paper in an international journal: up to 60 points.
 - Essay, paper in a national journal or book chapter: up to 40 points.
 - Talk at a scientific conference: up to 20 points.
 - Translation of someone else's work: up to 50 percent of the work's score.
- Artistic works: up to 150 points.
- Inventions, designs, and technological or cultural developments: up to 150 points.

The assessment of academic production is made according to the following criteria (number 71 of the Faculty Regulations): quality of the content; originality; contributions to the field; consistency with the methodology of the discipline; style and use of language; use, benefit and functionality of the work.

Assessment of degrees and additional sources of score: these scores are earned once.

- Second PhD degree: up to 160 points.
- Second masters degree: up to 80 points.

- Second undergraduate degree: up to 60 points.
- First and second specialization degree: up to 20 points each.
- Postdoctoral stays: up to 50 points each.
- Diploma courses: up to 10 points each and maximum 50 points in a rank.
- Distinctions and awards: up to 50 points each.
- Proficiency in a non-native language: 40 points.
- Translations of the faculty members works made by another person: up to 40 points.

Numerical model for the assessment of academic production The normalized score for academic production is calculated with the formula:

$$N_{AP} = 100 \left[1 - \exp \left(-\frac{2P_R + P_P}{2P_O} \right) \right].$$

In this formula, N_{AP} is the normalized score from academic production; P_R is the score from the sum of recent academic production (last five years); P_P is the score from the sum of previous academic production (before the last five years); and P_O is the minimum score for admission to the rank corresponding to position sought. If that position is for an Assistant Professor, $P_O = 120$; and $P_O = 350$ if the position is for a Associate Professor.

Numerical model for the assessment of experience The normalized score for years of experience is:

$$N_E = 100[1 - \exp(-T/T_0)].$$

In this formula N_E is the normalized score for experience; T is the full-time equivalent of the years of experience; and T_0 is the minimum time for promotion to the rank in question. If the position is for an Assistant Professor, $T_0 = 3$ years; and $T_0 = 8$ years for an Associate Professor.

Appendix F

Course evaluation

F.1 Course registry

Figure F.1 shows the template used for the course registry of the courses selected for the program assessment.

1. Catalogue code and name of the course
2. Credit hours and contact hours
3. Name of course coordinator
4. Course specific information <ul style="list-style-type: none">• Description• Prerequisites• Type of course
5. Learning objectives
6. Contents and time distribution
7. Curricular integration <ul style="list-style-type: none">• Curricular formula• Performance indicators• Integration of learning objectives, contents and methodology• Balance of evaluation
8. Course rules: <ul style="list-style-type: none">• Grading• Use of materials in exams• Attendance
9. Resources: <ul style="list-style-type: none">• Textbook and other sources• Facilities

Figure F.1: Template of the course registry

F.2 Performance indicators

Table F.1: Performance indicators associated with courses

Courses	Student outcomes										
	A	B	C	D	E	F	G	H	I	J	K
Fund. Mathematics	1,2,4	2,3								3	
Linear Algebra	1,2,4	2,3								3	
Differential Calculus	1,2,4	2,3								3	
Integral Calculus	1,2,4	2,3								3	
Multivariate Calculus	1,2,4	2,3								3	
Kin. and Dynamics	A	2,3,4								3	
Elec. and Magnetism	A	2,3,4								3	
Numerical Analysis	A							H		2,3	
Prob. and Statistics	A	2,3,4							2	2,3	
Discrete Mathematics	1,2,4								2	1,2,3	
Oral and Written Exp.				2,3,4		F					
Humanities I				2,3,4	2,3,4		3,4				
Humanities II				2,3,4	2,3,4		3,4				
Theology I				2,3,4	2,3,4		3,4				
Theology II				2,3,4	2,3,4		3,4				
English I						F					
English II						F					
English III						F					
Pol. Const. & Dem.				2,3,4	E		3,4				
Ethics				2,3,4	E		3,4				
Organizational Manag.				2,3,4	2,3,4		3,4				
Anal. & Des. of Alg.	A	B	C			F				J	2,3
Computer Architecture I	2,3		1,2,3	2					2,3	1	2
Computer Architecture II	2,3		1,2,3	2					2,3	1	2
Social and Prof. Issues				2,4	1,3,4	1,2,4	1,3,4	3,4			
Comput. & Formal Lang.	1,2,4					2,4				1,2,4	
Computer Graphics	1,2,3		1,2			F		1	I	1,2	
Large Scale SW Dev.	2	B	C	D		F	2,3		I	J	K
Formal Software Dev.	1	1,2,3			1		1		2	2,3	2,3
Web-based Services		2,3	1,2,3	1,2,4		F		2,3,4	I		K
Prog. Fund. & Data Str.	1,2		1,2,3			H		1,2	2		
Inf. Manag. & Modeling	1,2,3		1,2,3	2,4	2,3	1,4	G		1,2,4		
Database Implement.	2,3		1,2,3	D				2,3,4	I	2,3,4	1,2,4
Artificial Intelligence	A					F	G	H		J	
Int. to Computer Science	1,3				1,2	F	G	H			
Int. to Programming	2,3		1,2,3	1,2,4				2,3,4	1	2,3,4	
Int. to Modeling		B	2,3			F	G			J	3,4
Lab. of Programming			1,4					2,4	1		
Logic in Comp. Science	2,4									1,2,3	3
Obj. & Middle Scale	2		C	1,2,4		F					K
Software Eng. Processes	1,2,3	1,2,3	C	2,3,4		2,3			1,2	1,2,4	2,3,4
Networks and Comm.	1,2,3		2,3			F			3	2,4	4
Operating Systems	1,2							2,3,4	1,2,3	2,3,4	2,4
Fundam. of Research		B		D	1,2	F	G	H			
Undergraduate Project	A	B	C	D	2	F	G	H	I	J	K
Professional Internship	2,4	B	C	D	E	F	G	H	I	J	K
Engineering Economics	2,3						3				
Animation and Sim.	1,2,3			D					I	2,3,4	
Interaction and Sound	1,2,3							2,3,4	1,2,3		
Int. Video Games Dev.				D		F			2,3,4		
Art. Intel. for Games	1,2,3					1,2			1,2,4	2,3,4	
Mobile Computing								2,3,4	I	2,4	2,4
Information Security	1,2				2,3		G	2,3,4	1,2	2,3	
Multimedia Tech.	1,2					F		2,3,4	I		2,4

In Table F.1, cell numbers indicate the performance indicators that were assigned to each course and cell letters indicate that all the performance indicators of that student outcome were assigned to the course. The performance indicators of the Program were defined in Section 3.C of Criterion 3.

F.3 Numerical model for the evaluation of courses

The objective of this model is to obtain the overall result of a student in a course from the evaluation of student outcomes while satisfying the relative weights of the student outcomes and of the instruments of evaluation of the course.

The variables of the model are:

- j is the index for student outcomes corresponding to the course.
- k is the index for the instruments of evaluation in the course.
- P_{jk} is the performance of a student in the outcome j on the instrument k .
- w_{jk} is the relative weight of the student outcome j on the instrument k , which is given by the balance of evaluation of the course.

Therefore, the grade of a student on each of the student outcomes is given by:

$$g_j = \sum_k P_{jk} W_{jk}. \quad (\text{F.1})$$

In Eq. F.2 is the normalized weight of outcome j on the instrument k . It is necessary to use this variable as a particular outcome might not be distributed throughout the set of evaluation instruments. This normalized weight is given by the relation $W_{jk} = w_{jk} / \sum_k w_{jk} = w_{jk} / Q_j$. Thus, Eq. F.2 is reduced to:

$$g_j = \sum_k P_{jk} (w_{jk} / Q_j). \quad (\text{F.2})$$

It must be noticed that Q_j is the overall weight of the student outcome in the course, i.e. the weight given by the mapping of SOs versus courses (Table 4.1). The overall grade of the student in the course, i.e. through the evaluation of all the student outcomes, is:

$$G = \sum_j w_j \sum_k P_{jk} \left(\frac{w_{jk}}{Q_j} \right) \quad (\text{F.3})$$

In Eq. F.3, w_j is the relative weight of the outcome j in the overall evaluation given by:

$$w_j = \frac{\sum_k w_{jk}}{\sum_j \sum_k w_{jk}} = \frac{Q_j}{Q}$$

where Q is the overall sum of weights in the course evaluation. Thus, Equation F.3 is reduced to:

$$G = \sum_j \sum_k P_{jk} \left(\frac{w_{jk}}{Q} \right) \quad (\text{F.4})$$

The result of the procedure thus far is a table of students with results in each student outcome and the overall result in the course. However, the normal practice in the University is to report the grades of the evaluation instruments of the course, which is done through the following procedure. The grade of each student on the instrument k is:

$$g_k = \sum_j P_{jk} \left(\frac{w_{jk}}{Q_k} \right) \quad (\text{F.5})$$

where $Q_k = \sum_j w_{jk}$ is the weight of instrument k . Therefore, the overall grade of the student is:

$$G = \sum_k w_k \sum_j P_{jk} \left(\frac{w_{jk}}{Q_k} \right) \quad (\text{F.6})$$

Analogously to the above procedure,

$$w_k = \frac{\sum_j w_{jk}}{\sum_k \sum_j w_{jk}} = \frac{Q_j}{Q}$$

Thence,

$$G = \sum_k \sum_j P_{jk} \left(\frac{w_{jk}}{Q} \right) \quad (\text{F.7})$$

Clearly, Equation F.7 is the same as Equation F.4, which is the necessary result because the overall result of the student is unique no matter whether it is reported from student outcomes or from evaluation instruments. However, the procedures that led to each of the two expressions have different uses. Thus, the procedure leading to Equation F.4 is used to assess the contribution of courses to the development of student outcomes, whereas the procedure leading to Equation F.7 is used in the ordinary report of grades to the University registry.

F.4 Example of an evaluation balance:

This is the evaluation balance of the Formal Software Development course. To facilitate the reading of the balance, the distribution is given in integer “points”.

	Percentage	Student outcomes							Σ
		A	B	E	G	I	J	K	
Relative weights		1	2	1	1	1	5	5	16
Percentage		6.25	12.5	6.25	6.25	6.25	31.25	31.25	100
Exam 1	30	6	13	6	6		38	31	100
Exam 2	30	6	13	6	6		38	31	100
Homework	15	6	13	6	6	13	25	31	100
Course project	25	6	13	6	6	13	25	31	100
Total	100	24	52	24	24	26	126	124	400
Percentage		6.00	13.00	6.00	6.00	6.50	31.50	31.00	100

Table F.2: Evaluation balance of the Formal Software Development course

The following is an example of the distribution of evaluation in an exam of the Formal Software Development course.

	A	B	E	G	J	K	Total
Problem 1	6						6
Problem 2		13					13
Problem 3			6				6
Problem 4				6			6
Problem 5					38		38
Problem 6						31	31
Total	6	13	6	6	38	31	100

Table F.3: Distribution of evaluation in an exam of the Formal Software Development course.

As this table shows, the exam was structured in terms of six problems, each one of which evaluated one student outcome with the relative weight, in percentage points, given by the distribution of Table F.2. Therefore, the relative weights of student outcomes are satisfied according to the evaluation balance of the course.

F.5 Evaluation instruments used in the courses selected for the program assessment

This is the summary of evaluation instruments used by the courses selected for the program assessment. The cell labels are acronyms of the instruments used to that particular outcome, for instance, EPH means evaluation by exams (E), projects (P) and homework (H).

Table F.4: Evaluation instruments used in the courses selected for evaluation (H: Homework and workshops; E: Exams; N: News reports; P: Course project; Q: Quizzes; R: Reading report; S: Socialization; T: Presentations (talks))

Courses	Student outcomes										
	A	B	C	D	E	F	G	H	I	J	K
P. Fund. & Data Struct.	HEP		HEP			HEPT		HET	HPT		
Discrete Mathematics	E								EQ	E	
Logic in CS	E									HE	HE
Computer Architecture II											
Comp. and Formal Lang.	HEP					EP				HEP	
Info. Mng. and Mod.	HE		EP	P	HE	HP	HEP		HEP		E
Software Eng. Processes	HEQT	HEPQ	HEPQ	P		HEPQ			P	HPQ	EP
Computer Graphics	E		HP			PT		H	HT	HT	
Formal SW. Develop.	HEP	HEP			HEP		HEP			HEP	HEP
Social and Prof. Issues				PT	NPST	NPRST	NPRST	PRST			

F.6 Course report

Figure F.2 shows the template used for the course report at the end of courses.

1. Name of the course:
2. Name of the course coordinator:
3. Academic period:
4. Contribution of the course to the student outcomes: <ul style="list-style-type: none">• Attainment of the student outcomes (grades in SOs):• Description of the method used to collect the above information (evaluation balance):
5. Development and attainment of the learning objectives of the course:
6. Development of the course processes with regards to: <ul style="list-style-type: none">• Course contents and body of knowledge.• Course contents.• Students time of study.• Text, information resources, software.• Facilities.• Course rules and policies.
7. Changes implemented in the academic period with regards to: <ul style="list-style-type: none">• Curricular formula and evaluation balance.• Course contents.• Course methodology.• Facilities and resources.
8. Recommendations for future development with regards to: <ul style="list-style-type: none">• Curricular formula.• Evaluation balance and instruments.• Areas of special interest.• Contents and scheduling.• Resources (facilities, text, bibliography, software, hardware).
9. Additional comments:

Figure F.2: Template of the course report

F.7 Results of student outcomes in courses

These are the average results of the evaluation of student outcomes in courses over the period of observation 2012-2 to 2014-2. The scale is 0.00 to 5.00 with the pass at 3.00.

Table F.5: Average results on student outcomes of the evaluated courses

Courses	Program outcomes									
	A					B				
	12-2	13-1	13-2	14-1	14-2	12-2	13-1	13-2	14-1	14-2
Prog. Fund. and Data Struct.	2.92	2.30	2.39	2.80	2.83					
Discrete Mathematics	2.84	3.48	3.11	3.82	3.59					
Logic in Computer Science		3.20	3.02	3.25	3.36					
Computer Architecture II		2.57	3.01	2.92	3.09					
Comp. and Formal Lang.		2.86	3.27	3.10						
Info. Management and Mod.	1.90	2.04	3.05	3.04	2.69					
Software Eng. Processes	3.26	3.50	2.66	2.62	3.68	3.70	4.20	3.40	3.76	3.77
Computer Graphics		3.20	3.78	4.63	4.50					
Formal SW. Development	3.20	2.90	2.73	2.73	2.41	3.50	3.10	2.93	3.44	2.89
Social and Professional Issues										
Overall	2.79	2.88	3.03	3.41	3.25	3.64	3.89	3.27	3.67	3.51

Courses	Program outcomes									
	C					D				
	12-2	13-1	13-2	14-1	14-2	12-2	13-1	13-2	14-1	14-2
Prog. Fund. and Data Struct.	2.22	2.60	1.83	3.00	2.20					
Discrete Mathematics										
Logic in Computer Science										
Computer Architecture II		2.80	2.84	3.03	2.65		2.82	3.29	4.31	3.58
Comp. and Formal Lang.										
Info. Management and Mod.	3.30	3.01	3.68	3.78	3.10	3.50	4.06			
Software Eng. Processes	3.62	3.70	3.22	3.64	3.27	4.35	4.60	3.39	3.16	4.33
Computer Graphics		4.20	3.59	5.00	5.00					
Formal SW. Development										
Social and Professional Issues						3.93	4.10	0.72	4.50	3.98
Overall	3.05	3.24	3.01	3.66	2.90	3.93	3.90	2.47	3.99	4.03

Courses	Program outcomes									
	E					F				
	12-2	13-1	13-2	14-1	14-2	12-2	13-1	13-2	14-1	14-2
Prog. Fund. and Data Struct.						3.21	3.30	3.48	3.90	3.30
Discrete Mathematics										
Logic in Computer Science										
Computer Architecture II							2.94	4.15	3.90	4.38
Comp. and Formal Lang.							2.66	3.12	2.10	
Info. Management and Mod.	2.50	3.73	3.74	3.47	3.71	3.50	3.35	3.51	3.68	3.22
Software Eng. Processes						3.65	3.90	3.49	2.97	3.62
Computer Graphics							5.00	4.88	3.55	4.75
Formal SW. Development	3.80	4.10	4.31	3.89	3.54					
Social and Professional Issues	4.02	3.60	4.00	4.60	3.73	4.14	3.70	3.80	4.50	3.86
Overall	3.61	3.70	4.01	4.34	3.68	3.79	3.87	3.98	3.83	3.79

Courses	Program outcomes									
	G					H				
	12-2	13-1	13-2	14-1	14-2	12-2	13-1	13-2	14-1	14-2
Prog. Fund. and Data Struct.						3.35	2.90	3.17	3.50	2.58
Discrete Mathematics										
Logic in Computer Science										
Computer Architecture II										
Comp. and Formal Lang.										
Info. Management and Mod.	2.20	3.74	3.83	3.65	3.54					

Software Eng. Processes										
Computer Graphics							2.30	2.88	4.60	5.00
Formal SW. Development	3.60	3.10	3.37	3.70	4.14					
Social and Professional Issues	3.87	3.40	3.90	4.30	3.78	3.81	3.30	4.28	4.60	3.60
Overall	3.42	3.45	3.82	4.06	3.81	3.53	2.93	3.49	4.05	2.96
Courses	Program outcomes									
	I					J				
	12-2	13-1	13-2	14-1	14-2	12-2	13-1	13-2	14-1	14-2
Prog. Fund. and Data Struct.	2.39	2.20	1.13	2.40	1.53					
Discrete Mathematics	3.87	4.33	4.24	4.02	4.49	3.01	2.69	2.87	2.56	3.38
Logic in Computer Science							3.03	3.52	3.53	3.43
Computer Architecture II		2.73	3.54	3.89	4.24		2.00	3.09	3.57	3.54
Comp. and Formal Lang.							2.98	3.08	3.20	
Info. Management and Mod.	1.80	3.08	2.86	3.42	2.34					
Software Eng. Processes	3.27	4.00	2.73	2.16	2.99	3.06	3.30	2.53	3.04	3.94
Computer Graphics		4.60	3.89	5.00	5.00		4.50	4.88	4.70	5.00
Formal SW. Development	3.10	3.10	3.01	2.86	3.59	3.10	2.50	2.55	2.75	3.35
Social and Professional Issues										
Overall	2.61	3.41	2.89	3.43	2.83	3.07	2.96	3.22	3.29	3.56
Courses	Program outcomes									
	K					Overall				
	12-2	13-1	13-2	14-1	14-2	12-2	13-1	13-2	14-1	14-2
Prog. Fund. and Data Struct.						2.74	2.58	2.27	3.03	2.45
Discrete Mathematics						3.06	3.26	3.17	3.31	3.63
Logic in Computer Science		3.07	3.61	3.73	3.61		3.15	3.38	3.50	3.47
Computer Architecture II		2.60	3.66	3.23	3.74		2.61	3.36	3.43	3.37
Comp. and Formal Lang.							2.89	3.17	3.14	
Info. Management and Mod.						2.59	3.20	3.36	3.53	2.91
Software Eng. Processes	3.32	3.60	2.00	1.00	2.33	3.52	3.84	3.00	3.18	3.46
Computer Graphics							4.12	4.10	4.55	4.83
Formal SW. Development	2.90	2.70	2.72	2.98	3.19	3.17	2.84	2.85	3.04	3.26
Social and Professional Issues						3.98	3.57	3.77	4.48	3.78
Overall	3.09	2.90	2.98	2.88	3.16	3.21	3.26	3.25	3.57	3.29

F.8 Results in the national examinations

The results in the Saber Pro exam of 2012 and 2013 are summarized in Table F.6 regarding generic and program-specific competencies. Comparisons were available to the national reference groups only for generic competencies.

Generic competencies	No. of students evaluated		Average		Standard deviation		National average	
	2012	2013	2012	2013	2012	2013	2012	2013
Citizenship	19	22	10.78	10.37	1.09	0.76	10.12	10.04
Written communication	19	22	10.63	9.87	1.07	0.97	10.19	9.97
English	19	22	12.06	11.90	1.63	1.70	10.63	10.56
Critical reading	19	22	10.65	10.57	1.06	0.63	10.17	10.23
Quantitative reasoning	19	22	11.41	11.28	0.93	1.00	10.57	10.63

Program-specific competencies	No. of students evaluated			Average			Standard deviation		
	12	13-1	13-2	12	13-1	13-2	12	13-1	13-2
Formulation of engineering projects	13	10	12	10.00	10.47	10.56	1.16	0.47	1.17
Software design	13		12	10.19		11.16	0.91		1.07
Scientific thought	13	10	12	10.34	10.27	10.70	0.87	0.65	1.11

Table F.6: Annual averages of results in the Saber Pro exams of 2012 and 2013

Appendix G

Bibliographic data bases

Data base	Subject areas / Selected journal titles
Academic Search Complete (Ebsco)	Engineering & Computing, International Journal on Electrical Engineering & Informatics, Information & Systems Engineering; Systems Engineering, International Journal Of Smart Engineering System Design, Cognition, Technology & Work, IET Software, Software Magazine.
ACM	ACM Computing Surveys (CSUR), Journal of the ACM (JACM), Journal of Data and Information Quality (JDIQ), Journal of Experimental Algorithmics (JEA), ACM Journal on Emerging Technologies in Computing Systems (JETC), ACM Transactions on Autonomous and Adaptive Systems (TAAS), ACM Transactions on Algorithms (TALG).
Computer Source (Ebsco)	Computers & Industrial Engineering, Computers & Electrical Engineering, International Journal of Web Engineering & Technology, IEEE Software, ACM Transactions On Mathematical Software, Computer Life, Software Magazine.
IEEE Xplore Digital Library	IEEE Access, Annals of the History of Computing, IEEE, Circuits and Devices Magazine, IEEE; Communications, IEEE Transactions on, Computer, Computational Social Systems, IEEE Transactions on, Computational Social Systems.

ISI Web	Advanced engineering informatics, ACM Transactions on Autonomous and Adaptive Systems, Artificial Intelligence Review; Cognitive Computation, Autonomous Robot, IEEE Transactions on Fuzzy Systems.
MasterFile Premier (Ebsco)	Computer Conference Analysis Newsletter, Computer Weekly; Student Computer Use, Journal Of Computer Security, MacUser; MicroComputer Journal, Computerworld, Australian MacWorld, MC: Marketing Computers, Computing Japan.
ProQuest Career and Technical Education	Active Server Developer's Journal, Byte, C/C++ Users Journal, Computer Graphics World, Computer Security Journal, Data Communications, Data Mining, Distributed Computing, Information Technology, Learning, and Performance Journal, Macworld, MacUser.
ProQuest Computing	Circuits Assembly, Computer - Aided Engineering, ACM Transactions on Graphics, Byte; Computational Management Science, Computational Statistics, Dr. Dobb's Journal, IEEE Software, Journal of Software Engineering and Applications.
ProQuest Research Library	Academy of Information and Management Sciences Journal, ACM Transactions on Computer Systems, Comparative Economic Studies, IEEE Intelligent Systems, International Journal on Software Tools for Technology Transfer, Microsoft Systems Journal.
ProQuest Science Journal	ACM Computing Surveys, ACM Transactions on Modeling and Computer Simulation, AI Magazine, Applied Artificial Intelligence, Circuits, Systems, and Signal Processing, Computer - Aided Engineering, Computer Graphics World.
Science Direct	Advanced Engineering Informatics, Applied Soft Computing, Computational Geometry, Computer Aided Geometric Design, Computer Standards & Interfaces, Computers & Chemistry, Computers in Industry, Digital Investigation.

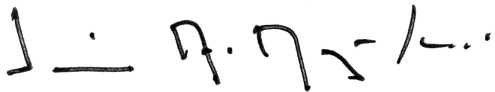
Table G.1: Data bases available for the Computer Science program

Signature Attesting to Compliance

By signing below, I attest to the following:

That the Computer Science Program has conducted an honest assessment of compliance and has provided a complete and accurate disclosure of timely information regarding compliance with ABETs *Criteria for Accrediting Engineering Programs* to include the General Criteria and any applicable Program Criteria, and the ABET *Accreditation Policy and Procedure Manual*.

Prof. Jaime Alberto Aguilar
Deans Name (As indicated on the RFE)



Signature

27th of May, 2015

Date