Course name:	**Course code:**
Software Architecture and Development	CC 500

Location in curricular map:
Specialization Axis

Course description:
In this course the student will understand and be convinced of the importance of software in current and future society. The student will know and understand the current problems facing software development, and visualize the magnitude ad impact for the future. The principles and concepts of software engineering are studied as a ramification of computer sciences that addresses the software development problem. The student will know, understand and apply the software development process, strongly based on fundamentals and principles of software engineering, and also completely associated with the object paradigm in all its phases. UML will be used as a language throughout the course.

Course learning outcomes:
At the end of the course, the student will: Define, analyze and design an effective software system, using a particular object oriented software engineering focus: Specifically, the student will: • Know and comprehend the current context for software systems from a corporate perspective (problems and challenges). • know and comprehend the computing platforms for the development of corporate applications. • Know, comprehend and experiment a generic software development process. • Know, comprehend and apply requirement capture and specification. • Know, comprehend, and apply analysis and modeling using UML. • Know, comprehend and apply implementation, transforming UML models to source codes using object oriented languages and technologies.

Course content:	Hours
1. Current context of software 1.1 Corporate software 1.2 Platforms for corporate software	8
2 Software engineering concepts 2.1 Software architecture 2.2 Development process 2.3 Software modeling	14
3. Activities of the software development process 3.1 Requirements gathering 3.2 Requirements analysis 3.3 System design 3.4 Implementation 3.5 Deployment	14

Learning activities guided by professor	Hours 36
1. Thematic exposition by professor	16
2. Laboratory practices and/or workshops guided by professor	16
3. Presentation and/or discussion plenary guided by professor	4
4. Small group activities guided by professor	OP
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend UML modeling and its relationship with programming. Specifically, chapters 3 and 4 of the Khawar Zaman Ahed text. • The student must do individual reading to know and comprehend Software Architecture concepts and the Software Development Process. Specifically, chapters 6 and 7 of the Khawar Zaman Ahed text. • The student must do individual reading to have a deeper knowledge of UML modeling in the analysis and design phases of an web application. Specifically, chapters 8,9,10,11 and 12 of the Khawar Zaman Ahed text. 2. Writing of an article, essay or reading summary. • This activity will not be done in this course. 3. Solution of problems selected by professor. • This activity will not be done in this course. 4. Lab practices. • The student must independently do the exercises of the guided lab workshop, with small variations. The workshops sessions will be focused on developing small applications. 5. Research and development of a topic selected by the professor. • This activity will not be done in this course. 6. Integral course project. • Student must give progress reports to the professor. Consists of following the activities for software development detailed throughout the course, and elaborating a report for each development activity using UML.	20 10 30

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of topic 2 of the course content.
2. Deliverables.
    - The student will deliver progress reports for each phase of the final project.
3. Presentations.
    - All students must present their final project to the group, on the day and hour that is established by the group and professor, and demonstrate the way to implement it with distributed computing platforms.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Written or oral exam 25 points.
    - Final project 75 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

Bibliography

	Type	Title	Author	Publisher	Year
	Text	Developing Enterprise Java Applications UIT J2EE and UML	Khawae Zaman Ahmed Cary E. Umrish	Addison Wesley	2002
	Reference	The Unified Software Process	Ivar Jacobson Grady Booch James Rumbaugh	Addison Wesley	2000
	Reference	The Unified Modeling Language	Grady Booch James Rumbaugh Ivar Jacobson	Addison Wesley	2000

Course name:	**Course code:**
Distributed Networks and Systems	CC 501

Location in curricular map:
Specialization Axis

Course description:
During this course the definitions of operating systems, networks and concepts relating to the client/server model will be covered. The participant will do a presentation of some commercial network environment, commenting on advantages and disadvantages. Also, the steps for the development of an application that communicates via computer networks will be studied in detail, and for this, JAVA language programming using sockets will be used. The participant must have prior knowledge of the OSI levels, network topology, UNIX, DOS and Windows operating systems, as well as C, JAVA or other equivalent programming language.

Course learning outcomes:
At the end of the course, the student will: Know and comprehend the concepts of operating systems and computer networks. Know and comprehend the primary models for distributed computing: client/server, peer-to-peer and multilayer. Know, comprehend and experiment with some network operating environments that exist in the market. Develop small applications that implement distributed computing models.

Course content:	Hours
1 Characterization of distributed systems, system models 1.1 Introduction to distributed systems. 1.2 Examples of distributed systems. 1.3 Challenges of distributed systems. 1.4 System models	6
2 Communication between processes 2.1 Introduction. 2.2 API for internet protocols. 2.3 External data representation. 2.4 Client/Server communication. 2.5 Group communication.	10
3 Distributed objects and remote calling 3.1 Introduction. 3.2 Communication between distributed objects. 3.3 Remote procedure calls. 3.4 Events and notifications.	8
4 Operating system support. 4.1 Introduction. 4.2 Levels of an operating system. 4.3 Protection. 4.4 Processes and threads. 4.5 Communication and calls. 4.6 Architecture.	12

Learning activities guided by professor	Hours 36
1. Thematic exposition by professor	24
2. Laboratory practices and/or workshops guided by professor	8
3. Presentation and/or discussion plenary guided by professor	4
4. Small group activities guided by professor	OP
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend the concepts relating to distributed systems. • The student must do individual reading to know and comprehend the concepts relating to networks and operating systems.	10
2. Writing of an article, essay or reading summary. • The student must write a technical article that presents a problem relating to distributed systems, and how to solve the problem using the concepts and technologies studied throughout the course.	8
3. Solution of problems selected by professor. • The student must define, design and program the solution to small problems where the concepts relating to operating systems and networks are applied. These problems will be defined by the professor.	15
4. Lab practices. • The student must independently do the required software installation processes. • The student must participate in the network programming workshops.	22
5. Research and development of a topic selected by the professor. • The student must research and develop a presentation of a topic relating to operating systems in network environments.	5
6. Integral course project. • Optional activity for this course, which may be exchanged for activities 2 or 4, with previous approval by professor. Consists of the implementation of a solution to a problem, which integrates the topics covered throughout the course. The Project may be established by the professor or suggested by the student, with approval of the professor, and must be developed throughout the course.	OP

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of the course topics.
2. Deliverables.
    - The student will deliver a technical report and programs, for each of the problems established by the professor.
    - The student will deliver an essay that describes what is indicated in activity 3 of the independent learning activities.
3. Presentations.
    - All students must present to the group the research on the assigned topic, on the day and hour that is established by the group and professor.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Exam 25 points.
    - Technical article 20 points.
    - Solution of problems 40 points.
    - Research and presentation of topic 15 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

Bibliography

	Type	Title	Author	Publisher	Year
1	Text	Distributed Systems, Concepts and Design, 3ra Edition	George Coulouris, Jean Dollimore Tim Kindberg	Addison-Wesley	2001
2	Reference	Distributed Systems: Principles and Paradigms	Steen, Maarten Van, Andrew S. Tanenbaum, Maarten Van Steen	Prentice Hall	2002
3	Reference	JAVA: How to Program, 5ta Edition	Deitel & Deitel	Prentice-Hall	2002
4	Reference	Java in a Nutshell, 3E.	David Flanagan	Oreilly	1999

Course name:	**Course code:**
Advanced Object Programming	CC 502

Location in curricular map:
Specialization Axis

Course description:
This course is focused on the study and application of the concepts of object oriented programming. The course contemplates a review of the basic concepts and philosophy of objects and their relationship with programming languages that follow this paradigm. Emphasis is made on the design of programs that solve complex problems, and also the correct way of implementing them, using object based design and data structures. The course contemplates the use of an object oriented language (JAVA or C#) and its aggregated tools to construct a packaged software.

Course learning outcomes:
At the end of the course, the student will: Know and comprehend object oriented concepts and identify these in an object oriented language. Apply object oriented concepts to the design of programs of medium and high complexity. Apply object oriented concepts to the design and programming of data structures. Know, understand and use software design patterns for the implementation of programs. Know, comprehend and apply the concept of class packaging for the creation of reusable libraries.

Course content:	Hours
1. 1. Object Oriented Programming Concepts	6
1.1 Classes and objects	
1.2 Data encapsulation and information hiding	
1.3 Messages and operations.	
1.4 Polymorphism	
1.5 Inheritance	
1.6 Relationships between classes and objects	
2. Object oriented program design	10
2.1 Definition of functionality	
2.2 Class structure design	
2.3 Design of object dynamics	
3. Predefined classes and error handling	8
3.1 Predefined library classes	
3.2 Encapsulation classes for data types	
3.3 Exception handling classes	
3.3 Collection handling classes	
4. Design of reusable classes and libraries	12
4.1 Design patterns	
4.2 Design of basic data structures	
4.3 Design of complex data structures	
4.4 Use of logic interfaces	
4.5 Class packaging	

Learning activities guided by professor	Hours 36
1. Thematic exposition by professor	10
2. Laboratory practices and/or workshops guided by professor	20
3. Presentation and/or discussion plenary guided by professor	6
4. Small group activities guided by professor	OP
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend the concepts relating to object orientation. • The student must do individual reading relating to the application of object orientation to program design.	10
2. Writing of an article, essay or reading summary. • The student must write a technical article for each of the problems he or she solves, either those selected by the professor or those of the final project. The article must present the problem, the suggested solution as well as the way to implement the technologies studied throughout the course to solve the problem, as well as conclusions.	8
3. Solution of problems selected by professor. • The student must define, design and program the solution to small problems where the concepts relating to object oriented programming. These problems will be defined by the professor.	15
4. Lab practices. • The student must independently do the required software installation processes. • The student must participate in the use of object oriented language workshops.	22
5. Research and development of a topic selected by the professor. • The student must research and develop a presentation of a topic relating to some JAVA class library, which will be assigned by the professor with input by the student.	5
6. Integral course project. • Optional activity for this course, which may be exchanged for activities 2 or 4, with previous approval by professor. Consists of the implementation of a solution to a problem, which integrates the topics covered throughout the course. The Project may be established by the professor or suggested by the student, with approval of the professor, and must be developed throughout the course.	OP

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of the course topics.
2. Deliverables.
    - The student will deliver a technical report and programs, for each of the problems established by the professor.
3. Presentations.
    - All students must present to the group the research on the assigned topic, on the day and hour that is established by the group and professor.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Exam 25 points.
    - Solution of problems 60 points.
    - Research and presentation of topic 15 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

	Type	Title	Author	Publisher	Year
1	Text	JAVA: How to Program, 5ta Edition	Deitel & Deitel	Prentice-Hall	2002
2	Reference	Java in a Nutshell, 3E.	David Flanagan	Oreilly	1999

Course name:	**Course code:**
Object and Data Systems	CC 503

Location in curricular map:
Specialization Axis

Course description:
This course has the purpose of achieving integration between data base management systems with software developed with object oriented programming languages. For this, the student will analyze and work on data modeling using various data models, such as ER, EER and the relational model. Also, working with RDBMS and how to integrate these with an object oriented language using data and object interconnection technologies. During the course, lab practices will be done. Also, the student must do reading, essays, problem solving and lab practices in an independent manner. At the end of the course the student will develop an application that involves data design and integration with object oriented languages in a network environment.

Course learning outcomes:
At the end of the course, the student will: Know and comprehend the primary concepts used in data bases and object oriented concepts. Know, comprehend and use concepts and characteristics of the ER, EER and relational models for information modeling. Know, comprehend and apply the normalization process as a data base design tool. Know and comprehend the primary characteristics of SQL as a standard and use it on a commercial DBMS. Know, comprehend and use object oriented languages for data access in a network environment. Know, comprehend and use technologies for data access or interconnection for the integration of objects with data. Know the primary tendencies relating to object oriented data bases, as well as those indicated by the ODMG.

Course content:	Hours
1. Introduction to data and object systems	4
1.1 Primary data base concepts	
1.2 Object oriented concepts	
2. Data base design	14
2.1 The Entity-Relationship model (E-R)	
2.2 The relational data model	
2.3 Normalization	
2.4 DBMS	
2.5 SQL	
3. Integration of data and objects	
3.1 Data access object oriented technologies	12
3.2 Object design for data access	
4. Object oriented data bases	
4.1 Introduction to object oriented data base systems	6
4.2 ODMG object standards model	
4.3 General vision of relational object oriented data bases (ORDBMS)	

Learning activities guided by professor	Hours **36**
1. Thematic exposition by professor	12
2. Laboratory practices and/or workshops guided by professor	12
3. Presentation and/or discussion plenary guided by professor	6
4. Small group activities guided by professor (problem solving)	6
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend the concepts covered throughout the course (Elmsri/Navathe text). • The student must do individual reading of manuals and references relating to software covered throughout the course.	10
2. Writing of an article, essay or reading summary. • The student must write a technical article or essay where he or she presents and identifies the characteristics of a good DBMS, which serve as evaluation criteria for the selection of a DBMS. This essay must be based on suggested reading by the professor and/or two references from articles suggested by the student.	3
3. Solution of problems selected by professor. • The student must solve data modeling problems using the tools studied throughout the course.	10
4. Lab practices. • The student must independently do the required software installation and configuration, as well as exercises and lab practices suggested by the professor.	12
5. Research and development of a topic selected by the professor. • Optional activity that may be exchanged for reading hours, or exchanged for activities 1 or 2, with previous approval by the professor.	OP
6. Integral course project. • Consists of the design and implementation of a solution to a real case or one established by the professor, which integrates the topics covered throughout the course. This may be exchanged for activity 4 with previous approval from the professor.	15

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of the course topics.
2. Deliverables.
    - The student will deliver a technical article or essay that presents and identifies the characteristics of a good DBMS, where he or she suggests the criteria for evaluation and selection of a DBMS.
    - The student will deliver a technical report and programs, for each of the problems established by the professor.
    - The student will deliver a technical report for each of the lab practices and exercises, which present the problems during their elaboration, and what was done to solve them.
    - The student will deliver a technical report and programs, used to solve the final project.
3. Presentations.
    - All students must present and essay and final project, on the day and hour that is established by the group and professor.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Exams 30 points.
    - Technical article 5 points.
    - Solution of problems 25 points.
    - Research and presentation of topic 10 points.
    - Final project 30 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

Bibliography

	Type	Title	Author	Publisher	Year
	Text	Fundamentals of Database Systems. Fourth Edition	Elsmasri/Navathe	Addison Wesley	2004
	Reference	MySQL and Java Developer's Guide	Matthews/ Cole / Gradecki	Wiley	2003
	Reference	MySQL Tutorial	Loke Welling/Laura Thomson	Sams Publishing	2004
	Reference	Servlets and JavaServer Pages	Marty Hall	Prentice Hall	2000

Course name:	**Course code:**
Distributed System Design	CC 504

Location in curricular map:
Specialization Axis

Course description:
This course is focused so the student knows, comprehends and applies the design of distributed software systems using object based design patterns and a distributed computing platform. The course is divided into four sections: the first presents the distributed computing platform (J2EE or .NET), the second discusses topics on the design of distributed software systems, the third addresses design pattern concepts, as well as considerations for their use in distributed environments, the fourth part of the course seeks for the student to experiment the use of a design pattern catalog.

Course learning outcomes:
At the end of the course, the student will: Know and comprehend the distributed computing technology, specifically .NET or J2EE. Know and comprehend the concepts of design patterns and their use in distributed system software design. Apply a select group of design patterns to the development of a distributed system. Experiment programming based on object patterns. Know the tendencies relating to software development for distributed systems based on patterns.

Course content:	Hours
1 Distributed computing platforms	6
1.1 General software platform concepts	
1.2 J2EE	
1.3 .NET	
2 Software design for distributed systems	10
2.1 Layers	
2.2 Tiers	
2.3 Interfaces	
2.4 Services	
2.5 Protocols	
5 Pattern concepts.	8
5.1 Design pattern concepts.	
5.2 Considerations for the presentation layer.	
5.3 Considerations for the logic layer.	
5.4 Re-factorization.	
6 Design patterns.	12
6.1 Presentation layer patterns.	
6.2 Logic layer patterns.	
6.3 Integration patterns.	

Learning activities guided by professor	Hours 36
1. Thematic exposition by professor	10
2. Laboratory practices and/or workshops guided by professor	20
3. Presentation and/or discussion plenary guided by professor	6
4. Small group activities guided by professor	OP
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend the concepts relating to design patterns. • The student must do individual reading relating to the design of distributed software systems.	10
2. Writing of an article, essay or reading summary. • The student must write a technical article for each of the problems he or she solves, either those selected by the professor or those of the final project. The article must present the problem, the suggested solution as well as the way to implement the technologies studied throughout the course to solve the problem, as well as conclusions.	8
3. Solution of problems selected by professor. • The student must define, design and program the solution to small problems where the concepts relating to design patterns are experimented and applied. These problems will be defined by the professor.	15
4. Lab practices. • The student must independently do the required software installation processes. • The student must participate in the use of object oriented language workshops.	22
5. Research and development of a topic selected by the professor. • The student must research and develop a presentation of a topic relating to some design pattern.	5
6. Integral course project. • Optional activity for this course, which may be exchanged for activities 2 or 4, with previous approval by professor. Consists of the implementation of a solution to a problem, which integrates the topics covered throughout the course. The Project may be established by the professor or suggested by the student, with approval of the professor, and must be developed throughout the course.	OP

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of the course topics.
2. Deliverables.
    - The student will deliver a technical report and programs, for each of the problems established by the professor.
3. Presentations.
    - All students must present to the group the researched topic, on the day and hour that is established by the group and professor.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Exam 25 points.
    - Solution of problems 60 points.
    - Research and presentation of topic 15 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

Bibliography

	Type	Title	Author	Publisher	Year
	Text	J2EE Design Patterns			2003
	Reference	Data structures in Java	Mark Allen Weiss	Addison Wesley	2000
	Reference	Introduction to Java Programming	David Arnow, Gerald Weiss	Addison Wesley	2001
	Reference	Java in a Nutshell, 3E.	David Flanagan	Oreilly	1999

Course name:	**Course code:**
Integration of Heterogeneous Systems	CC 505

Location in curricular map:
Specialization Axis

Course description:
This course is oriented towards the application of knowledge in the area of distributed systems as well as the use of an emerging technology known as "Web Services". During the class session, elements and components of this technology will be studied as well as their relationship with the interconnection of distributed systems. During the course, lab practices will be done. Also, during the course the student will do reading and lab practices independently to solve interconnection problems defined by the professor. At the end of the course, the student must identify an application area for this technology and suggest a design for implementation.

Course learning outcomes:
At the end of the course, the student will: Know and comprehend the context and problems for system interconnection in network environments and heterogeneous distributed systems. Know and comprehend the primary protocols and software technologies for the interconnection of distributed systems. Experiment the use of these protocols and technologies for the interconnection of distributed homogeneous and heterogeneous systems. Solve problems relating to the integration of distributed systems using these protocols and software technologies. Identify the primary areas of opportunity for the application of system interconnection. Know the primary tendencies of distributed system interconnection form the open network system and services design perspective.

Course content:	Hours
1. Introduction to network application connection	4
2.1 Introduction to Web Services	
2.2 Architecture of Web Services	
2.3 Security in Web Services	
2. Application connection and integration protocols	14
2.1 XML-RPC	
2.2 SOAP	
2.3 Interoperability aspects	
3. Connected application and services design	14
3.1 WSDL	
3.2 UDDI	
4. Application connection tendencies	
4.1 Service based system architecture	4
4.2 Nomenclature standardization	
4.3 Primary established consortiums	

Learning activities guided by professor	Hours 36
1. Thematic exposition by professor	16
2. Laboratory practices and/or workshops guided by professor	16
3. Presentation and/or discussion plenary guided by professor	4
4. Small group activities guided by professor	OP
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend "Web Services" technologies and the connection protocols. Specifically chapters 2,3,4,5,6 and 7 of the E. Cerami text. • The student must read an application of "Web Services" in current problems. Specifically chapters 1,2 and 3 of the Douglas K. Barry text.	20
2. Writing of an article, essay or reading summary. • The student must write a technical article that presents a problem relating to distributed systems interconnection, as well as the way to implement the technologies studied throughout the course to solve the problem.	10
3. Solution of problems selected by professor. • The student must solve 4 small problems relating to system interconnection, where the use of "Web Services" technology is demonstrated for specific situations.	20
4. Lab practices. • The student must independently do the exercises of the guided workshops, with small variations. The sessions will be based on chapters 2,3,4,5,6 and 7 of the E. Cerami text.	10
5. Research and development of a topic selected by the professor. • Optional activity that may be exchanged for reading hours, or exchanged for activities 1 or 2, with previous approval by the professor.	OP
6. Integral course project. • Optional activity for this course, which consist of the implementation of the solution stated in the technical article, and may also be exchanged with activity 3, with previous approval of the professor.	OP

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of the course topics.
2. Deliverables.
    - The student will deliver a technical article that presents a system interconnection problem that may be solved with the technologies covered throughout the course, and explain how it will be used.
    - The student will deliver a technical report and programs, for each of the problems established by the professor.
    - Alternatively, the student may do a technical report and deliver the programs done to solve the application problem presented in the technical article.
3. Presentations.
    - All students must present their technical article, on the day and hour that is established by the group and professor.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Technical article 25 points.
    - Solution to 4 problems 60 points.
    - Research and presentation of topic 15 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

Bibliography

	Type	Title	Author	Publisher	Year
	Text	Web Services Essentials.	Ethan Cerami	Oreilly	2002
	Reference	Web Services and Service-Oriented Architectures.	Douglas K. Barry	Mogan Kaufmann	2003
	Reference	Programming Web Services with SOAP	James Anell, Doug Tidwell y Pavel Kulchenko.	Oreilly	2002

Course name:	**Course code:**
Mobile Computing	CC 506

Location in curricular map:
Specialization Axis

Course description:

The course is designed as a seminar revolving around the topic of Mobile Computing. The student will first analyze the technology that supports this type of computing, and afterwards analyze and experiment with a mobile computing programming environment. Lastly, students will identify and analyze the applications of this computing model in real problems.

Course learning outcomes:

At the end of the course, the student will:

- Acquire an ample knowledge of hardware and software technologies that support mobile computing.
- Experiment in a programming environment for this paradigm.
- Know current applications of mobile computing and the advantages these offer.
- Identify and analyze real life situations in which mobile computing may be required.

Course content	Hours
1. Introduction to mobile computing. 1.1 Fundamentals of mobile computing 1.2 Challenges 1.3 Mobile computing in the 21st century	**10**
2. **Wireless networks.** 2.1 Wireless connectivity 2.2 Wireless network protocols 2.3 Mobile IP 2.4 Ad-Hoc networks	**10**
3. Models for mobile computing. 3.1 Platforms for mobile computing 3.2 Client/server in a mobile environment 3.3 Tool and languages.	**10**
4. Mobile computing applications and services 4.1 Mobile access to the WWW 4.2 Mobile telephony applications 4.3 Mobile data bases 4.4 Localization	**6**

Learning activities guided by professor:	Hours 36
1. Thematic exposition by professor	24
2. Laboratory practices and/or workshops guided by professor	8
3. Presentation and/or discussion plenary guided by professor	4
4. Small group activities guided by professor	OP
5. Individual activities guided by professor	OP

Independent learning activities:	Hours 60
1. Reading of materials selected by professor. • The student must do individual reading to know and comprehend the concepts relating to design patterns. • The student must do individual reading relating to networks and operating systems.	10
2. Writing of an article, essay or reading summary. • The student must write a technical article that presents a problem relating to distributed systems, as well as the way to implement the technologies studied throughout the course to solve the problem.	8
3. Solution of problems selected by professor. • The student must define, design and program the solution to small problems where the concepts relating to operating systems and networks are experimented and applied. These problems will be defined by the professor.	15
4. Lab practices. • The student must independently do the required software installation processes. • The student must participate in the network programming workshops.	22
5. Research and development of a topic selected by the professor. • The student must research and develop a presentation of a topic relating to operating systems in network environments.	
6. Integral course project. • Optional activity for this course, which may be exchanged for activities 3 or 4, with previous approval by professor. Consists of the implementation of a solution to a problem, which integrates the topics covered throughout the course. The Project may be established by the professor or suggested by the student, with approval of the professor, and must be developed throughout the course.	5

Evaluation instruments and procedures:

The evaluation procedures and instruments are the following:

1. Written or oral exam.
    - The student must prove to the professor via a written exam, the knowledge of the course topics.
2. Deliverables.
    - The student will deliver a technical report and programs, for each of the problems established by the professor.
    - The student will deliver the essay described in activity 3 of the independent learning activities.
3. Presentations.
    - All students must present to the group the researched topic, on the day and hour that is established by the group and professor.
4. Participation in discussion sessions.
    - This will not be subject to evaluation.

Evaluation criteria:

1. The evaluation instruments and procedures will be centered in the learning activities, either guided or not guided by the professor.
2. The professor will evaluate and assign a grade for each of the evaluation instruments. The assigned grade must be in a scale from 0 to 100.
    - Exam 25 points.
    - Technical article 20 points.
    - Solution of problems 40 points.
    - Research and presentation of topic 15 points.
3. The professor will report to the Graduate College the average grade of all the evaluation instruments, for each student.
4. The minimum passing grade for the course is 80.
5. A student may not obtain a failing grade due to accumulated non-attendance.

Bibliography

	Type	Title	Author	Publisher	Year
1	Text	Designing Software for the Mobile Context: A Practitioner's Guide	Roman Longoria	Springs, UK	2002
2	Reference.	.NET Wireless Programming	Mark Ridgeway	Sybex	2002

Course name:	**Course code:**
Application Project	CS 501

Location in curricular map:
Terminal Axis

Course description:

Throughout the course, the student will develop pan application project that demonstrates the capacity for analysis, team work, interpretation and application of knowledge and tools acquired throughout the masters program

Course learning outcomes:
The student will be capable of applying the knowledge and abilities acquired throughout the courses of the masters program, contributing to the development of practical solutions that benefit the community.

Course Content	Hours
1. Definition of application pre-project.	**16**
2. Ethics in professional services.	**4**
3. Project presentation.	**4**
4. Follow up by professor.	**4**
5. Presentation of pre results.	**4**
6. **Presentation of final results.**	**4**

Learning activities:

- **Guided activities:** **36**
  - Presentation of subject by professor.
  - Presentation by guest researchers.
  - Discussions of subjects and cases.
  - Final project presentation.
- **Independent activities:** **60**
  - Applied research case reading.
  - Information gathering.
  - Research reports.
  - Problem analysis.
  - Solution design.

Evaluation criteria and procedures:

The evaluation instruments are the following:

Homework and research work
Final project research
Participation

The points distribution for each instrument will be established in accordance with the group in the first class session.

Bibliography

	Type	Title	Author	Publisher	Year
1	None				