COURSE DATA

Data Subject

Code	34677
Name	Web applications developement
Cycle	Grade
ECTS Credits	6.0
Academic year	2023 - 2024

Study (s)

Degree	**Center**	**Acad. year**	**Period**
1400 - Degree in Computer Engineering	School of Engineering	3	Second term

Subject-matter

Degree	**Subject-matter**	**Character**
1400 - Degree in Computer Engineering	13 - Information systems and intelligent systems	Obligatory

Coordination

Name	**Department**
OLANDA RODRIGUEZ, RICARDO	240 - Computer Science

SUMMARY

The course "**Development of Web Applications**" is a subject of the third year of the Degree of Computer Engineering, which covers part of the compulsory subject *Information Systems and Intelligent Systems*.

This course constitutes a natural evolution of the knowledge and skills acquired in the course "*Human-Computer Interaction*" concerning the development of desktop applications. The course is thus devoted to more complex systems related to distributed environments and the client-server architecture. The basic lines of the course is structured around hypermedia systems and programming of dynamic Web environments and a brief introduction to the SOA architecture.

The aim is to provide a broad overview of the many development solutions for Web applications. Specifically, through this course we will address the programming languages used on both the client side (HTML5, CSS, Javascript) and server side (PHP, servlets, JSP) .

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

Other requirements

It is recommended to have studied all the previous subjects in the fields of computer science and Programming, Computing and Data Bases (2nd year)

OUTCOMES

1400 - Degree in Computer Engineering

- G4 - Ability to define, evaluate and select hardware and software platforms for the development and implementation of computer systems, services and applications, in accordance with both the knowledge and the specific skills acquired in the degree.
- G5 - Ability to design, develop and maintain computer systems, services and applications using software engineering methods as an instrument for quality assurance, in accordance with both the knowledge and the specific skills acquired in the degree.
- G6 - Ability to design and develop computer systems and centralised or distributed computer architectures which integrate hardware, software and networks, in accordance with both the knowledge and the specific skills acquired in the degree.
- R1 - Ability to design, develop, select and evaluate computer applications and systems while ensuring their reliability, safety and quality, according to ethical principles and current legislation and regulations.
- R4 - Ability to draw up the technical specifications of a computer system, according to standards and regulations.
- R8 - Ability to analyse, design, build and maintain applications in a robust, secure and efficient manner by choosing the most suitable paradigm and programming languages.
- R11 - Knowledge and application of the features, functionalities and structure of distributed systems, computer networks and Internet, and ability to design and implement applications based on them.
- TI2 - Ability to select, design, implement, integrate, evaluate, build, manage, exploit and maintain hardware, software and network technologies, within adequate cost and quality thresholds.
- TI6 - Ability to design systems, applications and services based on network technologies, including the Internet, the web, e-commerce, multimedia, interactive services and mobile computing.

LEARNING OUTCOMES

This course allows for the following learning outcomes or skills:

• Be able to determine the applicability of the components in the development of a particular software project.

• To choose the components development platform best suited to each type of project.

• Being able to develop in IDEs most common components in the market.

• Being able to apply the techniques of component-based development systems from WEB technologies and architectures appropriate in these systems

Furthermore specifically:

1. Modelling and distributing hypermedia content using specific languages.

2. Design and adapt presentation styles using markup languages.

3. Add dynamism to the pages (X) HTML using client side languages like Javascript.

4. Design and implement a complete Web application that integrates different programming technologies.

5. Knowing how to apply specific concepts of programming languages like Java to interact with a Web page in terms of:

- Format and processing of requests for forms (X) HTML.
- Persistence of data on the server through session variables and application.
- Elements that allow for management of persistent comfortable, like cookies, JavaBeans, custom tags.

To complement the above results, this course also allows to acquire the following skills:

• Model and solve problems being able to identify the essential elements of a situation and make approximations to reduce problems to a manageable level. This includes solutions that are not derived from the application of a standardized procedure, but providing original, creative and imaginative answers.

• Organize, plan and conduct their own learning, individually and in groups in a coordinated way.

• Working individually and in groups in a coordinated way.

• Work in groups: collaborating, leading, planning, interacting, getting consensus, negotiating, resolving conflicts and respecting the views of others.

• Argue, defend their views and be critical (and self-criticism) from rational and rigorous criteria.

- Preparation and presentation of texts in a clear, coherent, organized and understandable way.
- Oral and written comprehension.

DESCRIPTION OF CONTENTS

1. Fundamentals of Web

Web components: Using the standards URI, HTML, HTTP.
Web Container vs Aplications Container
Web applications. N-tier models.
HTTP protocol

2. Programming Languages in the client side

HTML5
CSS: Cascading Style Sheets.
Javascript.

3. Programming Languages in the server side (I)

Introduction to distributed programming. Differences regarding the desktop applications (sessions)
Models based on programming: CGIs and Servlets.

4. Programming Languages in the server side (II)

Models based on templates: PHP and JSP.
Model View Controller (MVC). Frameworks.
Introduction to the SOA architecture.

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Laboratory practices	20,00	100
Classroom practices	10,00	100
Development of group work	12,00	0
Development of individual work	8,00	0
Preparation of evaluation activities	15,00	0
Preparing lectures	26,00	0
Preparation of practical classes and problem	29,00	0
TOTAL	150,00	

TEACHING METHODOLOGY

During the on-site basis theoretical activities, the main topics of the course will be shown by providing a global and inclusive vision, analyzing in detail the key and more complex issues, encouraging at all times the students participation. These activities are complemented by practical activities in order to apply the basic concepts and to expand the knowledge and experience that is acquired during the performance of the proposed work. The on-site activities comprise the following:

• Problem-based lectures and questions in the classroom

• Sessions devoted to moderated discussions, and the resolution of problems and exercises that the students have previously worked

• Laboratory-based practical exercises

In addition to on-site activities, students must perform personal tasks (outside the classroom), including: monographs, guided literature research, questions and problems as well as the preparation of classes and exams (study). These tasks will be primarily conducted on an individual basis, thus enabling to enhance self-employment.

The University of Valencia e-learning platform (Aula Virtual) will be used to support the communication with students. Through this platform the studens will have access to course materials used in class as well as the problems and exercises to solve.

EVALUATION

Students can choose between two different assessments:

- Continuous assessment system (First Call)
- Overall system (Second Call)

Continuous assessment system (First Call)

The evaluation of the course is conducted by the aggregation of the following elements:

- Continuous assessment (N_Continua), based on the students participation and their degree ofinvolvement in the teaching-learning process, taking into account the regular attendance to onsite activities and the resolution of questions and problems, and the development of the works proposed to be delivered.
- Assessment of practical activities (N_Practicas,) from the achievement of objectives in the laboratoryand problems sessions, and the preparation of reports. Individually Oral presentations will be also part of this evaluation elements in order to evaluate the students capacity for creating documents and transfering knowledge.

Final Score = 35% × N_Continua + 65**%**× N_Practicas

Continuous assessment is distributed among the following items:

- Attendance *5%*
- Participation *5%*
- Activities throughout the course 25%

It will be necessary, at least, to obtain a 5 in N_Continua and N_Practicas to have the possibility to pass the course.

Attendance to laboratory sessions is mandatory. Students who do not attend at least a 80% of the laboratory sessions will not pass the practices evaluation in first call.

Overall System (Second Call)

This method is applicable to any student who did not pass the subject in first call.

The evaluation of the course is conducted by the aggregation of the following elements:

- Continuous assessment (N_Continua), based on the students participation and their degree ofinvolvement in the teaching-learning process, taking into account the regular attendance to onsite activities and the resolution of questions and problems, and the development of the works proposed to be delivered.
- Assessment of practical activities (N_Practicas,) from the achievement of objectives in the laboratoryand problems sessions, and the preparation of reports. Individually Oral presentations will be also part of this evaluation elements in order to evaluate the students capacity for creating documents and transfering knowledge.
- Individual objective test (N_Examen), consisting of an exams, or knowledge test, which consist of both theoretical/practical questions and problems.

Final Score = 10% × N_Continua + 40**%** × N_Practicas + **50%**× N_Examen

It will be necessary, at least, to obtain a 5 in N_Practicas and N_Examen to have the possibility to pass the course.

In both cases, the evaluation of this subject will be done in compliance with the University Regulations in this regard, approved by the Governing Council on 30th May 2017 (ACGUV 108/2017)

REFERENCES

Basic

- David Gourley & Brian Totty. HTTP. The Definitive Guide. ISBN-10: 1-56592-509-2, ISBN-13: 978-1-56592-509-0. Editorial: O´Reilly. 2002
- Collings, Matk J. Pro HTML5 with CSS, Javascript, and Multimedia. ISBN: 1-4842-2462-0, 978-1-4842-2462-5. 2018
- Budi Kurniawan, Servlet & JSP: A Tutorial. ISBN: 1-7719-7027-8, 978-1-7719-7027-3, 2015.
- Carr, David, Beginning PHP. ISBN: 1-78953-590-5, 978-1-78953-590-7, 2018

Additional

- HTML, CSS, Javascript recursos, https://www.w3schools.com/

- Javascript 1.2. http://www.programacion.net/html/tutorial/js/
- Servlets (Básico). http://www.programacion.net/java/tutorial/servlets_basico/
- Servlets y JSP. http://www.programacion.net/java/tutorial/servlets_jsp/
- Introducción a los Servicios Web en Java. http://www.programacion.net/java/tutorial/servic_web/
- HTML5 and JavaScript Projects, Meyer, Jeanine. ISBN: 1-4842-3863-X, 978-1-4842-3863-9, 2018
- Pro HTML5 Games, Shankar, Aditya Ravi, ISBN: 1-4842-2909-6, 978-1-4842-2909-5, 2017.