

<b>BLOCK NAME</b>	SERVER-SIDE WEB PROGRAMMING
<b>BLOCK CODE</b>	CS-L3B3
<b>COURSE</b>	2
<b>LEVEL</b>	3
<b>CREDITS</b>	5
<b>CLASS HOURS</b>	50
<b>HOMEWORK</b>	75
<b>TOTAL HOURS</b>	125

**DESCRIPTION**

This block introduces the basis of server-side web programming. We will face the challenge of completing our electronic business by implementing it's server-side part, making use of a server-side scripting language, accessing a database and taking care of the server-side web security. All this will be done following the Model-View-Controller design pattern.

**PRE-REQUISITES**

Basic programming and client-server web programming skills are needed. Knowledge about networking is desirable.  
**CS-L1B1, CS-L3B1, CS-L3B2**

**OBJECTIVES**

The goal is for students to be able to implement web applications at server-side, including access to databases.

**SKILLS TO BE DEVELOPED**

- 1 - Server-side scripting.**
  - 1.1 - Understand the differences between GET and POST requests.
  - 1.2 - Create server-side scripts that write HTML output as response from a browse request.
  - 1.3 - Create server-side scripts that reply AJAX requests.
  - 1.4 - Make use of received parameters.
  - 1.5 - Manage user sessions at web server.
- 2 - Database access.**
  - 2.1 - Connect to databases from server-side scripts.
  - 2.2 - Query databases from server-side scripts.
- 3 - Server-side web security fundamentals.**
  - 3.1 - Understand code injection attacks and know how to avoid them.
  - 3.2 - User password storage and management.
- 4 - Model-View-Controller (MVC).**
  - 4.1 - Understand the principles of MVC design pattern.
  - 4.2 - Use a MVC framework to create websites.

**SYLLABUS**

- 1 - Server-side scripting.
- 2 - Database access.
- 3 - Server-side web security fundamentals.
- 4 - Model-View-Controller (MVC).

**METHODOLOGY**

Resolution of practical activities supervised by the mentor. Compulsory attendance.

## DEDICATION AND EVALUATION

The student must pass the mandatory activities (challenges/projects) that are covered in the block.

Each challenge/project produces its own score and has been designed to cover certain block percentages.

Such score is 80% objective (the program that solves the challenge/project works without errors and producing the expected results) and 20% subjective (solution elegance, how clean the code is, documentation).

Block scores are finally calculated by prorating individual activities with respect to their block coverage percentages.