INSTITUTO POLITÉCNICO DE BRAGANÇA

Course Unit	Web Development Laboratory			Field of study	Computer Science		
Bachelor in	Informatics Engineering			School	School of Technology and Management		
Academic Year	2025/2026	Year of study	3	Level	1-3	ECTS credits	6.0
Type	Semestral	Semester	1	Code	9119-850-3104-00-25		

Workload (hours)	162	Contact hours	T -	TP 60	PL -	TC -	S -	E -	OT -	O -

T - Lectures; TP - Lectures and problem-solving; PL - Problem-solving, project or laboratory; TC - Fieldwork; S - Seminar; E - Placement; OT - Tutorial; O - Other

Name(s) of lecturer(s)	José Luís Padrão Exposto, Nuno Romeu Cardoso Sequeira

Learning outcomes and competences

At the end of the course unit the learner is expected to be able to:
1. Set up and configure web development environments.
2. Configure and manage development and production environments.
3. Apply collaborative development practices.
4. Develop complete web applications using the Django Framework.
5. Integrate technical and methodological knowledge.

Prerequisites

Before the course unit the learner is expected to be able to:
1. Knowledge of object-oriented programming.
2. Understanding of relational database fundamentals.

Course contents

Version control with Git and project management with GitLab. Containerization and orchestration technologies. Python fundamentals applied to web development. Web fundamentals. Django Framework. Authentication and authorization. Advanced topics: Django REST Framework, and production deployment with Docker.

Course contents (extended version)

1. Version control and collaboration
   - Git: repository initialization, commits, branching, merging, and conflict resolution.
   - GitLab: use of issues, milestones and merge requests.
   - Best practices for versioning and teamwork collaboration.
2. Containers and orchestration
   - Concepts of containers and images.
   - Creating and managing Dockerfiles.
   - Using Docker Compose to orchestrate multiple services.
   - Concepts of inter-container persistence.
3. Introduction to Python applied to Django
   - Creating and managing virtual environments with venv and requirements.txt.
   - Installing and managing packages with pip.
   - Primitive data types, lists, dictionaries, and flow control.
   - Functions, comprehensions, classes, modules, and packages.
   - Advanced concepts: generators, decorators, and asynchronous programming.
4. Web fundamentals
   - Client-server model, HTTP protocol, requests, and responses.
   - Methods, status codes, headers, and cookies.
   - Concepts of sessions and frontend-backend communication.
5. Django architecture and configuration
   - Project structure and the MTV pattern.
   - Project and application creation.
   - Main configuration files and environment variables.
   - URL configuration and routing.
6. Views and templates
   - Functional views and context rendering.
   - Templates, inheritance, and inclusion.
   - Best practices for file organization and component reuse.
7. Models and ORM
   - Definition of models, fields, and relationships (1–N, N–M).
   - Migrations and database schema evolution.
   - Django Admin: registration, customization, and permissions.
   - Class-based views (CBVs), generic views, QuerySets, and custom managers.
   - Query optimization and filtering.
8. Forms and validation
   - Creation and handling of forms.
   - ModelForm, formsets, and data validation.
   - Integration with templates and views.
9. Authentication and authorization
   - User and permission management.
   - Implementation of authentication, registration, and login.
   - Route protection and security best practices.
10. Advanced topics and deployment
   - Introduction to Django REST Framework: serializers, viewsets, and routers.
   - Building RESTful APIs and service integration.
   - Configuration of static and media files.
   - Preparing and deploying applications in production using Docker Compose.

Recommended reading

1. Mark Lutz. Learning Python: Powerful Object-Oriented Programming, O'Reilly Media , 2025
2. Antonio Melé. Django 5 By Example: Build powerful and reliable Python web applications from scratch. Packt Publishing. 2024
3. Python documentation. https://docs.python.org/3/. 2025
4. Django documentation. https://docs.djangoproject.com/. 2025

Teaching and learning methods

The course combines the theoretical exposition of concepts with a strong practical component. Learning is based on solving exercises, developing prototypes, and applying project-based learning methodologies. Continuous monitoring is carried out using GitLab for collaboration, code submission, and review.

Assessment methods

1. Normal period - (Regular, Student Worker) (Final)
   - Practical Work - 10%
   - Projects - 40%
   - Final Written Exam - 50% (Minimum of 7/20 to approve.)
2. Other periods - (Regular, Student Worker) (Supplementary, Special)
   - Final Written Exam - 100%

Language of instruction

1. Portuguese
2. English

Electronic validation

José Luís Padrão Exposto, Nuno Romeu Cardoso Sequeira	Tiago Miguel Ferreira Guimaraes Pedrosa	Isabel Maria Lopes	José Carlos Rufino Amaro
13-10-2025	15-10-2025	15-10-2025	01-11-2025