
Ano Letivo 2019-20

Unidade Curricular DESENVOLVIMENTO DE APLICAÇÕES WEB

Cursos PROGRAMAÇÃO DE DISPOSITIVOS PARA A INTERNET
Tronco comum

Unidade Orgânica Instituto Superior de Engenharia

Código da Unidade Curricular 18411012

Área Científica FORMAÇÃO TÉCNICA, CIÊNCIAS INFORMÁTICAS

Sigla FT

Línguas de Aprendizagem Português (PT)

Modalidade de ensino Obrigatória

Docente Responsável Roberto Célio Lau Lam

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Roberto Célio Lau Lam	PL; TP	TP1; PL1	15TP; 45PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
2º	S1	15TP; 45PL	125	5

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação, tecnologias subjacentes ao funcionamento da WEB e dominar conhecimentos sobre a inter-relação entre os diferentes operadores da WEB (ver Ficha da Unidade Curricular Produção Conteúdos Web).

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Objetivos:

Pretende-se transmitir aos alunos conhecimentos que permitam utilizar as tecnologias Web para a integração dos dispositivos da Internet (IoT). A utilização destas tecnologias (Web) permitirá aos alunos: a) conhecer e compreender o modo como se integram os IoTs na Web, b) Conhecer e compreender o potencial de crescimento da Web das Coisas (WoT), nomeadamente a integração/analise de dados provenientes dos IoTs.

Competências

Saber: a) desenvolver aplicações/rotinas interativas/dinamicas para integração máquina-máquina, humano-máquina na Web, b) desenvolvimento de aplicações/rotinas para coleção e armazenamento de dados na Web (servidores base de dados relacionais), c) integração de funcionalidades e serviços locais ou com servidores remotos e d) desenvolvimento de aplicações/rotinas para analise de dados armazenados.

Conteúdos programáticos

1. Introdução
2. Programação em Node.js
3. Serviços Web
4. Conceção e desenvolvimento de serviços Web (APIs) em Node.js/RESTful
5. Trabalhos práticos para integração de aplicações através serviços Web
6. Integração/participação em projetos com aplicações Web em desenvolvimento

Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

A introdução permitirá aos alunos perspetivar a Web como plataforma de interação/integração dos IoTs, para a grande maioria dos IoTs. O ponto 2 (Node.js) permitirá aos alunos desenvolver clientes e servidores Web, baseados na linguagem JavaScript. O ponto 3 fornecerá o conceitos chave dos modelos Service-Oriented Architecture (SOA) e Representational State Transfer (REST). O ponto 4 permitirá aos alunos conceber e implementar serviços Web sob RESTful. A implementação de aplicações, de interação/integração máquina-máquina e humano-máquina serão efetuadas pelos trabalhos propostos no ponto 5. O ponto 6 permitirá aos alunos envolverem-se em projetos extra UC, nomeadamente em projetos de extensão e voluntariado.

Metodologias de ensino (avaliação incluída)

Aulas teórico-práticas para a exposição resumida dos conteúdos e resolução de casos. As práticas para apoio e realização de casos práticos e exercícios. A plataforma da tutoria eletrónica da UALG será utilizada, com os seguintes objetivos: a) Publicação dos materiais das aulas práticas; b) Publicação de avaliações; c) Publicação de avisos da disciplina; d) Criação de um espaço de comunicação (fórum de discussão) para esclarecimento de dúvidas e estímulo da comunicação docente /alunos e alunos/alunos. Modo de Avaliação: Teste/exame escrito (30% nota final) e apresentação dos trabalhos práticos feitos em programação (70% nota final). Para obter nota de aprovação os alunos terão de obter pelo menos 7 valores (0-20) no teste/exame e 7 valores (0-20) na componente dos trabalhos de programação. Nota final = 30% teste/exame + 70% trabalhos (deverá ser superior a 9,5 valores). O aluno não deve exceder o número limite de faltas correspondente a 25% das horas de contacto previstas (RT 59/2015).

Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

A aprendizagem emprega um misto da metodologia Problem Based Learning (PBL) e o classico método expositivo. desenvolve-se em trabalho de grupo, servindo as necessidades de resolução dos problemas, sendo centrada no aluno e assumindo a diversidade de modelos pessoais de aprendizagem. Em cada ponto dos objectivos será exposto as tecnologias que permitem a resolução do caso apresentado (problema) que constituirá o elemento agregador do processo de aprendizagem pelos alunos.

Bibliografia principal

Documentação das aulas teórico-práticas
Alex R. Young, Marc Harter-Node.js in Practice, Manning, 2015.
Ethan Cerami, Web Services, O'REILLY, 2002.
Ethan Brown, Web Development with Node & Express, O'REILLY, 2014.
Basarat Ali Syed, Beginning Node.js, Apress, 2014.
RESTfull Web Services, https://www.crummy.com/writing/RESTful-Web-Services/RESTful_Web_Services.pdf (acesso em 29/07/2019).
D.D. Guinard and V. M. Trifa, Building the Web of Things, Manning, 2016.
w3schools.com Node.js and Raspberry Pi, https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp (acesso em 29/07/2019).
Powers S., Learning Node, O'REILLY, 2nd edition, 2016
Create a Node.js web app in Azure, <https://docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-nodejs> (acesso em 29/07/2019)

Academic Year 2019-20

Course unit PRODUCTION OF WEBSITE CONTENT

Courses PROGRAMMING OF INTERNET DEVICES
Tronco comum

Faculty / School INSTITUTE OF ENGINEERING

Main Scientific Area CIÊNCIAS INFORMÁTICAS,FORMAÇÃO TÉCNICA

Acronym FT

Language of instruction Portuguese (PT)

Teaching/Learning modality Required

Coordinating teacher Roberto Célio Lau Lam

Teaching staff	Type	Classes	Hours (*)
Roberto Célio Lau Lam	PL; TP	TP1; PL1	15TP; 45PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
0	15	45	0	0	0	0	0	125

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Master the knowledge about programming and technologies underlying the WEB and the interrelationship between different WEB operators

The students intended learning outcomes (knowledge, skills and competences)

Objectives

It is intended to provide students with knowledge that enables the use of Web technologies for the integration of Internet devices (IoT). Using these technologies (Web) will enable students to: a) Know and understand how IoT's integrate into the Web, b) Know and understand the growth potential of the Web of Things (WoT), including data integration / analysis from IoT's.

Skills

Know: a) develop interactive / dynamic applications / routines for machine-to-machine, human-machine web integration, b) develop applications / routines for web data collection and storage (relational database servers), c) integration of local or remote server features and services and d) development of applications / routines for analyzing stored data.

Syllabus

1. Introduction
2. Programming in Node.js
3. Web Services
4. Design and Development of Web Services (APIs) in Node.js / RESTful
5. Practical work for application integration through Web services
6. Integration / participation in projects with web applications under development.

Demonstration of the syllabus coherence with the curricular unit's learning objectives

The introduction will allow students to view the Web as an IoT interaction / integration platform for the vast majority of IoTs. Point 2 (Node.js) will allow students to develop clients and web servers based on the JavaScript language. Point 3 will provide key concepts from the Service-Oriented Architecture (SOA) and Representational State Transfer (REST) models. Point 4 will allow students to design and implement web services under RESTful. The implementation of machine-machine and human-machine interaction / integration applications will be carried out by the work proposed in point 5. Point 6 will allow students to engage in extra-UC projects, namely extension and/or volunteering projects.

Teaching methodologies (including evaluation)

Theoretical-practical classes for the brief exposition of the contents and resolution of cases. Practices to support and carry out practical cases and exercises. The UALG electronic tutorial platform will be used, with the following objectives: a) Publication of the materials of the practical classes; B) Publication of evaluations; C) Publication of discipline notices; D) Creation of a communication space (discussion forum) to clarify doubts and stimulate teacher communication / students and students / students. Assessment Type: Written exam / exam (30% final grade) and presentation of the practical work done in programming (70% final grade). To obtain approval grade, students must obtain at least 7 values (0-20) in the test / exam and 7 values (0-20) in the programming component. Final grade = 30% test / exam + 70% programming component (should be more than 9,5). The student should not exceed the limit number of absences corresponding to 25% of the planned contact hours (RT 59/2015).

Demonstration of the coherence between the teaching methodologies and the learning outcomes

Learning employs a mix of the Problem Based Learning (PBL) methodology and the classic expository method. It develops in group work, serving the needs of problem solving, being student-centered and assuming the diversity of personal learning models. At each point of the objectives will be exposed the technologies that allow the resolution of the case presented (problem) that will be the aggregating element of the learning process by the students.

Main Bibliography

Classroom documentation

Alex R. Young, Marc Harter-Node.js in Practice, Manning, 2015.

Ethan Cerami, Web Services, O'REILLY, 2002.

Ethan Brown, Web Development with Node & Express, O'REILLY, 2014.

Basarat Ali Syed, Beginning Node.js, Apress, 2014.

RESTfull Web Services, https://www.crummy.com/writing/RESTful-Web-Services/RESTful_Web_Services.pdf (acesso em 29/07/2019).

D.D. Guinard and V. M. Trifa, Building the Web of Things, Manning, 2016.

w3schools.com Node.js and Raspberry Pi, https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp (acesso em 29/07/2019).

Powers S., Learning Node, O'REILLY, 2nd edition, 2016

Create a Node.js web app in Azure, <https://docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-nodejs> (acesso em 29/07/2019)