

Unidade Curricular	Desenvolvimento e Tecnologias de Software		Área Científica	Informática	
Mestrado em	Informática		Escola	Escola Superior de Tecnologia e Gestão de Bragança	
Ano Letivo	2022/2023	Ano Curricular	2	Nível	2-2
Tipo	Semestral	Semestre	1	Créditos ECTS	6.0
Horas totais de trabalho	162	Horas de Contacto	T - - TP 60 PL - - TC - - S - - E - - OT - - O - -	Código	5060-710-2102-00-22

T - Ensino Teórico; TP - Teórico Prático; PL - Prático e Laboratorial; TC - Trabalho de Campo; S - Seminário; E - Estágio; OT - Orientação Tutoria; O - Outra

Nome(s) do(s) docente(s) Rui Pedro Sanches de Castro Lopes

Resultados da aprendizagem e competências

No fim da unidade curricular o aluno deve ser capaz de:

1. Planificar projetos integradores de desenvolvimento de software
2. Cooperar, organizar e conceber estratégias de resolução de problemas tecnológicos
3. Analisar e conceber ferramentas, abordagens e metodologias com potencial de resolução de problemas complexos
4. Reconhecer e utilizar as diferentes áreas do conhecimento como meio de incorporação e interpretação dos problemas sociais na área da tecnologia

Pré-requisitos

Não aplicável

Conteúdo da unidade curricular

O desenvolvimento de software deve ser contextualizado nos problemas que se pretendem desenvolver e depende de um conjunto de ferramentas e APIs adequadas no âmbito de uma, ou mais, linguagens de programação. Cada problema deve ser bem percecionado, seguindo-se opções relativas à metodologia de desenvolvimento. Os estudantes devem ter a capacidade de implementar metodologias que permitam perceber o problema à luz do estado-da-arte.

Conteúdo da unidade curricular (versão detalhada)

1. Contexto dos problemas de desenvolvimento de software
 - Investigação em desenvolvimento de software
 - Metodologias de levantamento do estado-da-arte
 - Reflexão e investigação sobre desenvolvimento aplicado
 - Grandes áreas de intervenção
 - Recolha e análise de dados em contexto
2. Conceção do(s) projeto(s) de desenvolvimento
 - Definição do problema a investigar
 - Desenho do(s) projeto(s) de intervenção e abordagem a desenvolver na prática
 - Definição de requisitos investigativos
3. Especialização multi-linguagem
 - Estudo e adoção de linguagens de programação
 - Estudo e adoção de APIs
4. Do conceito ao protótipo

Bibliografia recomendada

1. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects, Arnon Axelrod, Apress, ASIN: B07FKGVQP6, 2018.
2. Release It! : Design and Deploy Production-Ready Software, Michael T. Nygard, Pragmatic Bookshelf, ASIN: B079YWMY2V, 2018
3. Java Unit Testing with JUnit 5: Test Driven Development with JUnit 5, Shekhar Gulati and Rahul Sharma, Apress, ASIN: B077GDZ216, 2017
4. Continuous Delivery with Docker and Jenkins: Create secure applications by building complete CI/CD pipelines, 2nd Edition, Rafal Leszko, Packt Publishing, ASIN: B07SJKHJZ7, 2019
5. Developer Testing: Building Quality into Software, Alexander Tarlinder, Addison-Wesley, ASIN: B01LHSV9Z1, 2016

Métodos de ensino e de aprendizagem

Trabalho de projeto, trabalho em grupo e discussão argumentativa. Reflexão individual sobre temas e questões decorrentes das experiências de aprendizagem realizadas.

Alternativas de avaliação

- Projeto com relatório escrito - (Ordinário, Trabalhador) (Final, Recurso, Especial)

Língua em que é ministrada

Inglês

Validação Eletrónica

Rui Pedro Sanches de Castro Lopes	José Luís Padrão Exposto	José Eduardo Moreira Fernandes	Paulo Alexandre Vara Alves
11-10-2022	12-10-2022	29-10-2022	03-11-2022