

| | | | | | |
|--------------------------|-------------------------------|-------------------|-----------------|---|------|
| Unidade Curricular | Desenvolvimento de Aplicações | | Área Científica | Ciências da Computação | |
| Licenciatura em | Engenharia Informática | | Escola | Escola Superior de Tecnologia e de Gestão de Bragança | |
| Ano Letivo | 2023/2024 | Ano Curricular | 3 | Nível | 1-3 |
| 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 | 9119-706-3101-00-23 | |

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) José Luís Padrão Exposto, Daniel Jose Lopes Gouveia, Tiago Filipe Lino Santos

Resultados da aprendizagem e competências

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

1. Utilizar uma framework de aplicações no desenvolvimento de uma aplicação informática
2. Identificar as diversas vertentes de utilização da framework de aplicações e compreender o funcionamento dos seus principais componentes
3. Utilizar ambientes integrados de desenvolvimento para criar soluções, gerir itens, compilar as soluções e efetuar depuração de aplicações
4. Compreender os conceitos de evento e manipulador de evento e aplicá-los na manipulação de eventos gerados pelo sistema operativo e pelo utilizador
5. Saber organizar o desenvolvimento de aplicações em camadas arquiteturais
6. Utilizar controlos de janelas e caixas de diálogo para a visualização e interação com informação complexa
7. Utilizar ligações de dados nos elementos de interação com o utilizador
8. Gerir trabalho em equipa com ferramentas colaborativas e de controlo de versões

Pré-requisitos

Antes da unidade curricular o aluno deve ser capaz de:

1. Dominar a criação de modelos de programação orientada por objetos
2. Conhecer o funcionamento e utilização de estruturas de dados abstratas

Conteúdo da unidade curricular

Universal Windows Platform. Linguagem C#. Padrões de arquiteturas de software. Acesso a bases de dados. Linguagem XAML. Windows 10 apps. Controlo de versões e colaboração.

Conteúdo da unidade curricular (versão detalhada)

1. Universal Windows Platform
 - Desenvolvimento de aplicações para o Windows.
 - Evolução das APIs do Windows.
 - Convergência de plataformas.
 - Targeting de aplicações. IU adaptável. Código adaptável. Extensões de SDKs.
 - Espaços de nome e assemblies. Bibliotecas . NET Standard.
 - Planificação de aplicações. Objetivos. Requisitos. Representação de dados. Mapa da aplicação.
 - Desenho de uma aplicação UWP. Anatomia de uma aplicação.
 - Padrões de navegação. Elementos de comandos. Padrões de páginas. Elementos de conteúdo.
2. Linguagem C#
 - Entrada e saída de dados.
 - Tipos de dados. Alocação de entidades. Classificação de tipos de dados. Boxing e unboxing.
 - Tratamento de exceções.
 - Arrays, strings e tipos enumerados.
 - Classes. Propriedades. Herança e Interfaces. Agregação. Destruutores.
 - Coleções e Generics. Listas e dicionários. Iteração de elementos. Filas e Pilhas.
 - Manipulação de eventos. Eventos e Delegados. Expressões lambda.
 - Programação assíncrona. Operações com predominio de E/S e CPU.
3. Padrões de arquiteturas de software
 - Arquiteturas de software. Princípios de arquiteturas de software.
 - Aplicações monolíticas. Aplicações de microserviços avançadas.
 - Arquitetura de camadas clássica. Arquitetura Clean.
 - Padrão repositório e unidade de trabalho.
 - Padrão Modelo-Vista-VistaModelo.
4. Acesso a bases de dados
 - SQL Server e Bases de dados Azure.
 - EntityFramework Core.
5. Linguagem XAML
 - Estrutura básica.
 - Esquemas e espaços de nomes.
 - Extensões de markup.
 - Eventos.
 - Dicionários de recursos. Recursos e estilos.
6. Desenvolvimento de aplicações UWP
 - Criação de projetos.
 - Ciclo de vida de uma aplicação.
 - Linguagem XAML. Schemas e espaços de nomes. Extensões de Markup. Dicionários de recursos. Estilos.
 - Controlos de Layout.
 - TextBlock, TextBox, ListView, GridView, Hub, CommandBar, Flyouts e Content Dialogs.
 - Controlos de navegação.
 - Controlos de coleções: Data Templates, Data Context e Data Binding. Extensões {x: Bind}.
 - Gestão de ficheiros.
 - Controlos definidos pelo utilizador.
7. Controlo de versões e colaboração
 - Controlo de versões com o Git. Fluxo do Git. Git e GitLab.
 - Instalação do Git.
 - Comandos do Git.
 - Ramificação e fusão.

Bibliografia recomendada

1. C# 7. 0 com Visual Studio - Curso Completo, Henrique Loureiro, FCA, 2017

Bibliografia recomendada

2. Universal Windows Platform documentation. <https://docs.microsoft.com/en-us/windows/uwp/>. 2020
3. Windows universal samples. <https://github.com/Microsoft/Windows-universal-samples>, 2020
4. Git HowTo, <https://githowto.com/>, 2019

Métodos de ensino e de aprendizagem

Aprendizagem baseada em projetos com projetos de software em grupo e exposição teórica de conceitos focada em momentos oportunos para evolução do projeto.

Alternativas de avaliação

1. Época Final - (Ordinário, Trabalhador) (Final)
 - Projetos - 60% (Projeto final)
 - Exame Final Escrito - 40% (Mínimo 7 valores)
2. Época de Recurso/Especial - (Ordinário, Trabalhador) (Recurso, Especial)
 - Exame Final Escrito - 100%

Língua em que é ministrada

1. Português
2. Inglês

Validação Eletrónica

| | | | |
|--------------------------|---|--------------------------|--------------------------|
| José Luís Padrão Exposto | Tiago Miguel Ferreira Guimaraes Pedrosa | Lúisa Maria Garcia Jorge | José Carlos Rufino Amaro |
| 16-10-2023 | 25-10-2023 | 25-10-2023 | 31-10-2023 |