

Road To DB

Wrap it up - 23/03/2022

Paulo Vieira

Desenvolvimento de *software*

- Não é um processo linear
- Para progredir no caos é necessário um processo sistemático
 - Orientado a um propósito
- Aprendizagem é individual, não colectiva
 - Reconhecer os sinais: explorar mais, compreender melhor, quando parar e quando continuar?

Desenvolvimento de *software*

- **Fase de análise:** exploração de possibilidades, experiências, tentativas
 - Fazer, “amontoar” linhas de código

Desenvolvimento de *software*

- **Fase de análise:** exploração de possibilidades, experiências, tentativas
 - Fazer, “amontoar” linhas de código
- **Fase de sistematização:** documentação, reflexão, organização, modularidade
 - Estudar, resumir, simplificar – escrever o “relatório”, ler e comentar o código

Desenvolvimento de *software*

- **Fase de análise:** exploração de possibilidades, experiências, tentativas
 - Fazer, “amontoar” linhas de código
- **Fase de sistematização:** documentação, reflexão, organização, modularidade
 - Estudar, resumir, simplificar – escrever o “relatório”, ler e comentar o código

Quando analisar e quando sistematizar?

Desenvolvimento de *software*

- Qual das competências é mais relevante?
 - Dominar/decorar todos os detalhes de todos conceitos?
 - Aprender, compreender e conseguir encontrar o que preciso na documentação, avançando com base no raciocínio, dedução e experimentação?

Para que serve o projecto?

Para que serve o projecto?

- *Sandbox* para sistematização de conceitos
- Organização para possível revisão de tópicos:
 - Tipos de dados
 - Instruções de controlo de fluxo e de repetição
 - Instâncias, tipos, classes, campos, propriedades e conceitos de OOP (modularidade, herança, encapsulamento e polimorfismo) – *static, private, public, protected*
 - Classes genéricas e Interfaces
 - Estruturas de dados
 - Enumerados
 - Introspecção, *Delegates, Nuget* e dependências
 - Atributos
 - JSON, leitura de ficheiros, serialização e deserialização, *ficheiros de configuração*
 - Programação assíncrona
 - *Dispose*
 - *ToString, Equals, IEnumerable, Linq*

Trabalho futuro

- *ADO .NET* para ligação à Base de dados
- *Entity Framework Core*
- Depois:

ASP .NET Core

(desenvolvimento de aplicações *web*)

Para que serve o relatório?

E o relatório?

- Fazer e explicar são duas formas úteis de validação de aprendizagem
 - Realçam pontos de falta de conhecimento e incerteza
 - Criação de hábitos de trabalho:
 - Tão ou mais importantes que a aprendizagem de tópicos tecnológicos
 - facilmente obtidos através de vídeos e tutoriais (na *web*)
 - através de copy+paste de soluções do *stack overflow*
- Memória futura de tópicos que são a base de qualquer projecto em que venham a ser colocados (em .NET Core)
 - Todos terão detalhes diferentes, mas os conceitos identificados estarão inevitavelmente presentes

Para que servem as
perguntas?

```
*****  
*** Welcome to the nightmarish Road To Database! ***  
*****
```

Challenges:

0. Does it work? What does it do?
1. Study the solution and projects. Identify what does what! Comment the code at your will.
2. Consider the other six db models. What do you need to integrate them in the solution? Check if you can get it done.
3. Add a Find option in the menu that receives an ID and shows the selected entity only.
4. Find any bugs and document them. If you can, correct them and document the solution.
5. Consider some topics highlighted in the comments, study and summarize them (hint: Reflection, Enumeration).
6. If you needed to change the JSON files location, how many times would you modify the path? Suggest a correction for this issue.
7. Questions:
 - a) Console: How do we write in color?
 - b) How to map db models to json files? Explain how it is done.
 - c) To integrate other db models, do you need to modify the manager?
 - d) What is the purpose of the ICrud interface and IHasPrimaryKey interface? How are they used?
 - e) Is there a difference between "Operation" and "Entity"? What do those concepts mean?
 - f) Why do we need "newOfT" in the Manager?
 - g) What is the purpose of the "new()" restriction on the top?
 - h) Why do we need IEnumerable<T> implementation in the Manager?
 - i) Is "Entity" really needed? What was its purpose?
 - j) What is the usefulness of singleton design pattern in the Manager?
 - k) Is the path attribute the only way of solving the path issue? Suggest an alternative.

∴ Deliver your report by e-mail. Deadline: 23th March 2022 @9AM

E as
perguntas?

```

*****
*** Welcome to the nightmarish Road To Database! ***
*****

Challenges:

0. Does it work? What does it do?
1. Study the solution and projects. Identify what does what! Comment the code at your will.
2. Consider the other six db models. What do you need to integrate them in the solution? Check if you can get it done.
3. Add a Find option in the menu that receives an ID and shows the selected entity only.
4. Find any bugs and document them. If you can, correct them and document the solution.
5. Consider some topics highlighted in the comments, study and summarize them (hint: Reflection, Enumeration).
6. If you needed to change the JSON files location, how many times would you modify the path? Suggest a correction for this issue.
7. Questions:
  a) Console: How do we write in color?
  b) How to map db models to json files? Explain how it is done.
  c) To integrate other db models, do you need to modify the manager?
  d) What is the purpose of the ICrud interface and IHasPrimaryKey interface? How are they used?
  e) Is there a difference between "Operation" and "Entity"? What do those concepts mean?
  f) Why do we need "newOfT" in the Manager?
  g) What is the purpose of the "new()" restriction on the top?
  h) Why do we need IEnumerable<T> implementation in the Manager?
  i) Is "Entity" really needed? What was its purpose?
  j) What is the usefulness of singleton design pattern in the Manager?
  k) Is the path attribute the only way of solving the path issue? Suggest an alternative.

.: Deliver your report by e-mail. Deadline: 23th March 2022 @9AM

```

E as perguntas?

- Além da memória futura “ganha” ao escrever o documento, os primeiros passos são sempre essenciais – compila e funciona na minha máquina? O que se pretende neste projecto?
- Em qualquer projecto em que se trabalhe é relevante manter o “olho aberto” a potenciais erros e quebras de funcionalidade
- Comentar linhas de código implica uma leitura cuidada para compreender e uma nova leitura/pesquisa sobre tópicos desconhecidos
- Com o tempo, aumentará o entendimento geral do projecto e virá a capacidade de incorporar novas funcionalidades, acrescentando novos aspectos aos que já são conhecidos

Conclusões

- Resumo do módulo 3 - .NET Core/C# total
- Agregado de algumas boas práticas num contexto suficientemente generalizável – “canivete suíço”
- Cábula/Memória futura para os primeiros projectos em que ainda há pouca confiança e capacidade de concretização
- Possibilidade de extensão com outros conceitos/padrões a explorar individualmente após o final da formação, com possibilidade de incorporar novos desenvolvimentos para “pequenas experiências”