



ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

PEI: 2º ANO _1º SEMESTRE

NCAR-1º MILESTONE

Trabalho realizado por:

Hugo Ribeiro – 8200441

Lúcia Pereira – 8200278

Pedro Teixeira – 8200489

ÍNDICE

Índice de Ilustrações	2
Introdução	3
Estudo do domínio	3
Estrutura do projeto	3
Abordagem do Problema	3
Documento das marcações – XSD_XML	4
Documento das peritagens – XSD_XML	5
BaseX e API REST	7
Postman	8
Execução do Projeto	11
Considerações finais	11

ÍNDICE DE ILUSTRAÇÕES

Código 1 - Exemplo de uma restrição definida no ficheiro commonTypes.xsd	4
Código 2 – Exemplo de código .xml que valida o código postal da morada	4
Código 3 - Exemplo do elemento complexo com restrições de enumeração que definem o estado da bateria do veículo.	5
Código 4 - Exemplo .xml que para este parâmetro numa dada situação está em conformidade.....	5
Código 5 - Exemplo do elemento complexo com restrições de enumeração que definem o estado da transmissão do veículo	6
Código 6 - Exemplo .xml em que define o estado da transmissão do motor com identificação de dois problemas (Car not working on Petrol, MIL light glowing)	6
Ilustração 1 - Consultar os dados de uma peritagem	8
Ilustração 2 - Apresentar as peritagens realizadas/não realizadas num determinado dia para um dado parceiro.....	9
Ilustração 3 - Adicionar Expertises a DataBase	10

INTRODUÇÃO

A informatização dos dados referente aos processos e trabalhos nas empresas veio facilitar a sua acessibilidade, capacidade de armazenamento e organização em termos internos.

A “NCar” é uma empresa que tem como principal atividade a realização de inspeções mecânicas aos automóveis dos seus clientes, no entanto o processo de comunicação de peritagens é bastante complexo devido à dificuldade de comunicação entre os demais sistemas informáticos dos parceiros.

Desta forma, é necessário desenvolver um vocabulário XML e uma REST API de forma a facilitar a integração dos dados entre os parceiros.

ESTUDO DO DOMÍNIO

Este trabalho passou por uma fase de contextualização onde mediante uma pesquisa de atividades com modelo de negócio similares podemos perceber melhor sobre os processos de negócio que prestam serviços de inspeções de veículos com base em marcações. Podemos constatar que neste tipo de negócios existem diferentes parceiros que estão ligados à empresa-mãe, dispersos por diferentes regiões do país. Neste sentido a comunicação e partilha de informação uniformizada torna-se fulcral para agilizar todo o processo de negócio e prestar um serviço aos clientes com eficácia e eficiência.

ESTRUTURA DO PROJETO

- **/XSD_XML** – Contem os documentos XSD, “resevations.xsd”, “expertises.xsd”, “commonType.xsd”, utilizados na validação das marcações e peritagens feitas por um perito num determinado dia, bem como os documentos XML que servem de exemplo para este trabalho.
- **/webapp** – podemos encontrar todos os files que contém o código em xquery da API desenvolvida em BaseX,
- **/postman**- onde podemos encontrar a coleção do Postman com pedidos HTTP pré-definidos (com duas portas diferentes)
- **/baseX** – pasta de instalação do baseX
- **/data** – onde conta a base de dados no Base

ABORDAGEM DO PROBLEMA

No geral foram apresentados dois documentos XML distintos um para as marcações outro para as peritagens e respetivos XSD que os validam. Surge um terceiro documento XSD que serve de suporte à definição/restrições de alguns tipos de elementos que são comuns aos dois documentos principais. O domínio do *namespace* que abrange todo o projeto é

<https://www.NCar.com/>.

DOCUMENTO DAS MARCAÇÕES – XSD_XML

Para abordar o problema apresentamos a solução de criar um ficheiro XSD com a estrutura que consideramos correta para levar a cabo a apresentação de dados referentes a um conjunto de marcações a fim de agendar inspeções/peritagens a veículos. Foi feito um documento XML que serve de exemplo, que foi validado pelo XSD. No documento XML “**reservations.xml**”, o elemento root **reservas(<reservations>)** tem como elementos filhos a **identificação do parceiro (<partnerName>)** e a **identificação do perito (<expertiseName>)** que são representados por um código alfanumérico que correspondem à sua identificação; a **data (<date>)** e a **reserva (<reservation>)**. O elemento **<reservation>** é onde está compreendida toda a informação de todos os trabalhos a ser efetuados por um perito nessa determinada data.

Trata-se, do documento que contém um conjunto de marcações feitas onde disponibiliza também entre outras informações a morada (**<address>**), detalhes sobre o veículo (**<vehicleInfo>**), bem como o tipo de contato do cliente (**<typeOfContact>**). Neste documento podemos também verificar uma importação do documento “CommonTypes.xsd” onde se encontram definidas as restrições que caracterizam cada elemento acima mencionado. O *namespace* deste documento encontra-se definido como <http://www.NCar.com/commonTypes.xsd>.

Um dos elementos que constam neste “CommonTypes.xsd” é o **zipCodeType** como mostra a imagem seguinte:

```
1. <xs:simpleType name="zipCodeType">
2.   <xs:annotation>
3.     <xs:documentation>Regras que definem o codigo-postal</xs:documentation>
4.   </xs:annotation>
5.   <xs:restriction base="xs:string">
6.     <xs:pattern value="[0-9]{4}-[0-9]{3}"/>
7.   </xs:restriction>
8. </xs:simpleType>
```

CÓDIGO 1 - EXEMPLO DE UMA RESTRIÇÃO DEFINIDA NO FICHEIRO COMMONTYPES.XSD

```
1. <address>
2.   <street>Rua do Curral</street>
3.   <numDoor>12</numDoor>
4.   <zipCode>4610-156</zipCode>
5.   <city>Felgueiras</city>
6.   <coordinates></coordinates> <!-- caso não seja possível fornecer coordenadas basta apenas
7.     não chamar elemento coordinates ou chamar e não chamar latitude e longitude-->
8. </address>
```

CÓDIGO 2 – EXEMPLO DE CÓDIGO .XML QUE VALIDA O CÓDIGO POSTAL DA MORADA

DOCUMENTO DAS PERITAGENS – XSD_XML

Para abordar o problema apresentamos a solução de criar um ficheiro XSD com a estrutura que consideramos correta para levar a cabo a apresentação de dados referentes a um conjunto de peritagem feitas num dia por um determinado perito de veículos. Foi feito um documento XML que serve de exemplo, que foi validado pelo XSD.

Neste documento que demos nome tem na sua raiz a data <date>, o código do perito <expertiseName> (aqui deveríamos ter alterado para expertId), do parceiro <partnerName> e o conjunto de peritagens que foram feitas nesse dia pelo perito <expertise>.

Cada peritagem efetuada/realizada é lançada uma lista com o resultado da avaliação de vários parâmetros sobre o estado de um veículo. Essa lista <checkList> incorpora nela vários elementos que são os parâmetros predefinidos para cada peritagem que visam verificar as seguintes componentes de um veículo: <battery>, <oilLevel>, <engineTransmission>, <refrigerationSystem>, <engineSupport>, <engineSound>, <gases>, <gearBox>, <kms>.

Estas componentes são avaliadas e resultam num report de informação sobre o estado do veículo. As anomalias estão predefinidas como parâmetros como podemos verificar no XSD e caso existam são apontadas no documento que resulta no xml expertises, onde serão eventualmente alvo de futuras intervenções.

Seguidamente estarão apresentadas alguns excertos de código referentes aos parágrafos acima.

```
1. <xs:complexType name="batteryType">
2.   <xs:choice>
3.     <xs:element name="Status">
4.       <xs:simpleType>
5.         <xs:restriction base="xs:string">
6.           <xs:enumeration value="Weak" />
7.           <xs:enumeration value="Not Working" />
8.           <xs:enumeration value="Jump Start" />
9.           <xs:enumeration value="Discharging light glowing" />
10.          <xs:enumeration value="Intermittent functioning" />
11.        </xs:restriction>
12.      </xs:simpleType>
13.    </xs:element>
14.    <xs:element name="conformidade" fixed="yes"></xs:element>
15.  </xs:choice>
16. </xs:complexType>
```

CÓDIGO 3 - EXEMPLO DO ELEMENTO COMPLEXO COM RESTRIÇÕES DE ENUMERAÇÃO QUE DEFINEM O ESTADO DA BATERIA DO VEÍCULO.

```
1. <battery>
2.   <conformidade></conformidade>
3. </battery>
```

CÓDIGO 4 - EXEMPLO .XML QUE PARA ESTE PARÂMETRO NUMA DADA SITUAÇÃO ESTÁ EM CONFORMIDADE

```

1. <xs:complexType name="engineTransmissionType">
2.     <xs:choice>
3.         <xs:element name="Status" minOccurs="1" maxOccurs="unbounded">
4.             <xs:simpleType>
5.                 <xs:restriction base="xs:string">
6.                     <xs:enumeration value="Car not working on Petrol" />
7.                     <xs:enumeration value="Fuel Leakage From Injector" />
8.                     <xs:enumeration value="MIL light glowing" />
9.                     <xs:enumeration value="Misfiring " />
10.                    <xs:enumeration value="Not converting to CNG" />
11.                    <xs:enumeration value="Over heating" />
12.                    <xs:enumeration value="RPM fluctuating " />
13.                    <xs:enumeration value="RPM not increasing" />
14.                    <xs:enumeration value="Repaired" />
15.                    <xs:enumeration value="Seized" />
16.                    <xs:enumeration value="Sump Damaged" />
17.                    <xs:enumeration value="Turbo charger not working" />
18.                </xs:restriction>
19.            </xs:simpleType>
20.        </xs:element>
21.        <xs:element name="conformidade" fixed="yes"></xs:element>
22.    </xs:choice>
23. </xs:complexType>

```

CÓDIGO 5 - EXEMPLO DO ELEMENTO COMPLEXO COM RESTRIÇÕES DE ENUMERAÇÃO QUE DEFINEM O ESTADO DA TRANSMISSÃO DO VEÍCULO

```

1. <engineTransmission>
2.     <Status>Car not working on Petrol</Status>
3.     <Status>MIL light glowing</Status>
4. </engineTransmission>

```

CÓDIGO 6 - EXEMPLO .XML EM QUE DEFINE O ESTADO DA TRANSMISSÃO DO MOTOR COM IDENTIFICAÇÃO DE DOIS PROBLEMAS (CAR NOT WORKING ON PETROL, MIL LIGHT GLOWING)

BASEX E API REST

Após a construção de um vocabulário XML e validar a sua estrutura usando o XML Schema, foi necessária a implementação de uma interface web de forma a ser possível a troca de mensagens entre dispositivos. Para este efeito, utilizou-se uma API com arquitetura REST com recurso a funções na linguagem de consulta XQuery, para tal foi utilizado o HTTP service do BaseX, que nos permite fazer pedidos HTTP, os módulos estão presentes na pasta “/BaseX/webapp”.

Foram necessárias funções capazes de suportar a submissão e substituição de documentos e consulta de dados.

- ❖ `add.xq`: permite verificar se o código da marcação, do parceiro e do perito existem, e estão presentes todas as restantes funções/queries de adicionar, seja de reservas e peritagens.
- ❖ `expertrealized_not.xq`: esta função permite obter as peritagens realizadas e não realizadas num determinado dia para um dado parceiro.
- ❖ `getDatafromExpertise.xq`: esta função permite consultar os dados de uma peritagem
- ❖ `replace_expertises.xq`: esta função permite fazer a substituição de um documento das peritagens para um determinado dia
- ❖ `Reservationbydate.xq`: esta função permite consultar as marcações associadas a um perito para o dia atual.

POSTMAN

A ferramenta **Postman** é utilizada para desenvolver e testar APIs. Através desta plataforma foi possível enviar os pedidos necessários à API criada e consequentemente rever as respostas recebidas.

Após o desenvolvimento da API, começamos a trabalhar numa coleção de pedidos utilizando o software **Postman**, a coleção poderá ser importada para o postman usando o ficheiro json presente na pasta “/Postman”.

Desta forma, os testes para verificar as funcionalidades da API realizaram-se de forma mais concisa.

De seguida são apresentados alguns exemplos de pedidos feitos ao **Postman**, de maneira a testar todas as funções desenvolvidas e requisitadas para o desenvolvimento da API:

Neste exemplo é possível ver a consulta de dados de uma peritagem, para tal, é usado a path “getDatafromExpertise”, invocando a respetiva query que está presente no módulo getDatafromExpertise.xq. Neste caso é feita a consulta de dados de uma peritagem de acordo com um determinado “code”.

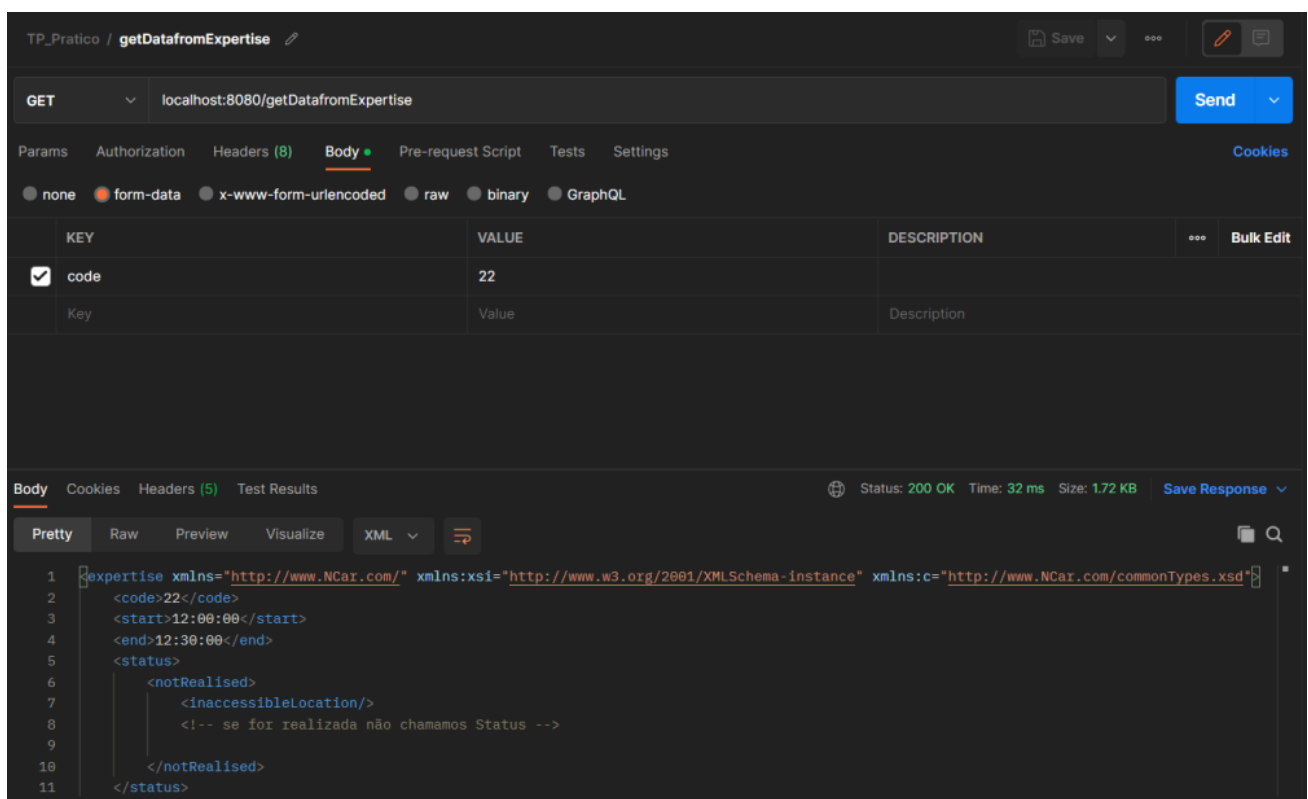


ILUSTRAÇÃO 1- CONSULTAR OS DADOS DE UMA PERITAGEM

Neste exemplo é possível verificar a consulta das peritagens realizadas ou não realizadas de um determinado dia, para tal utiliza a path “/RealizedExpertises” para invocar a respetiva query e fazer o respetivo pedido. Neste caso estamos a consultar as peritagens com as respetivas restrições:

expertiseName = A4S5

date = 2022 – 12 – 08

type = not_realized

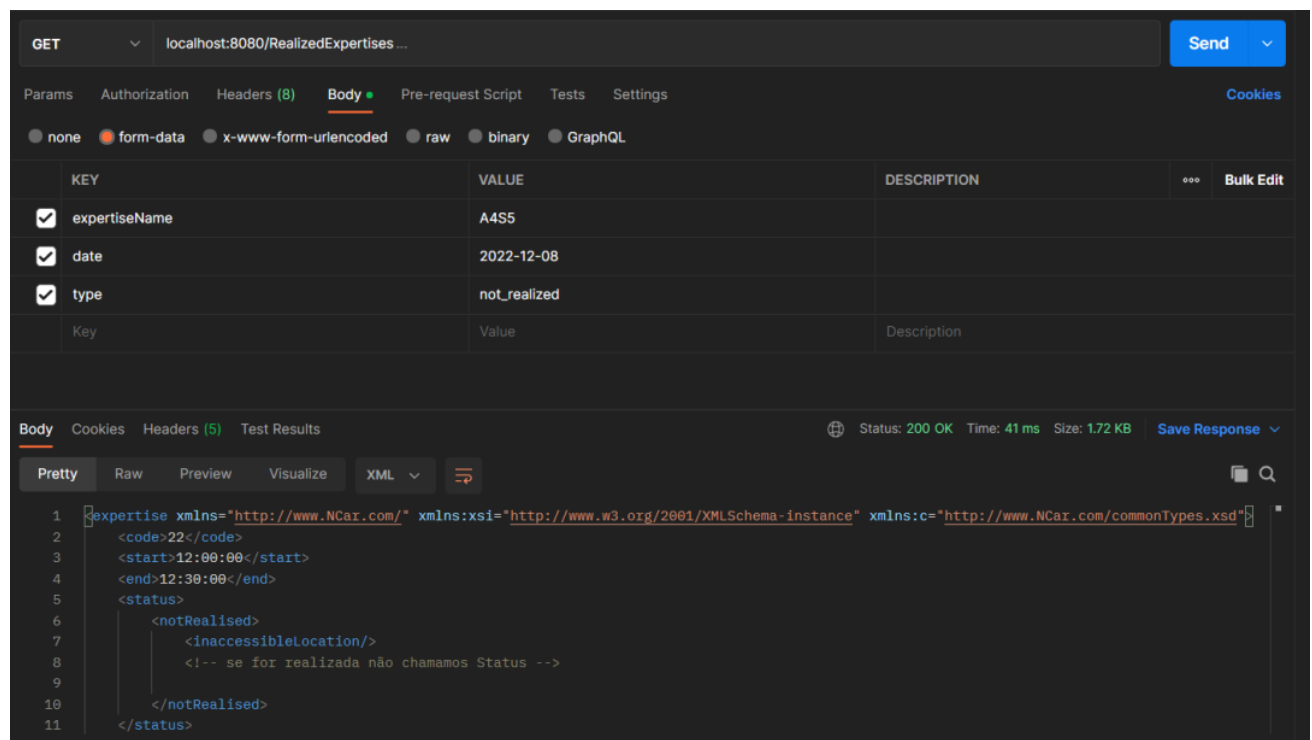


ILUSTRAÇÃO 2 - APRESENTAR AS PERITAGENS DE UM DETERMINADO DIA PARA UM DADO PARCEIRO

Neste exemplo é possível verificar a utilização de uma query com a path “.../add/Expertises”, presente no modulo “add.xq”, esta permite adicionar um conjunto de expertises à DataBase “NCar” e caso seja realizado com sucesso é gerado o output que esta presente na imagem abaixo, caso contrário será gerado “Erro ao Adicionar”.

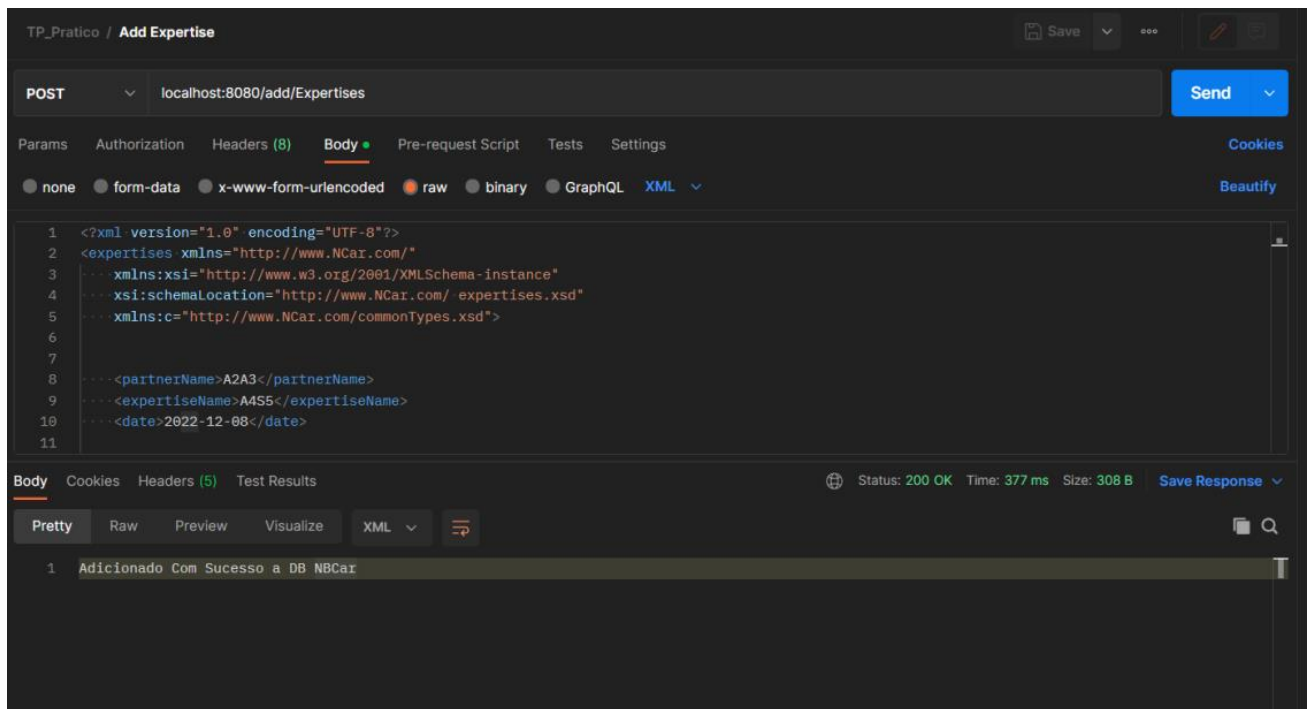


ILUSTRAÇÃO 3- ADICIONAR EXPERTISES A DATABASE

EXECUÇÃO DO PROJETO

Para inicializar o web service do BaseX, basta executar o ficheiro basexhttp.bat.

De seguida no Postman, importar coleção na pasta localizada na pasta “/Postman”.

Após a importação, pode executar as várias funcionalidades da API, utilizando os pedidos HTTP já pré-definidos.

CONSIDERAÇÕES FINAIS

Após a realização deste trabalho, concluímos que numa primeira fase, fizemos uma abordagem aos documentos XSD e XML de forma menos correta. Este trabalho teria sido facilitado se primeiro optássemos por construir o documento XML e só depois definir a sua estrutura através do XSD.

Posteriormente, foi necessário explorar a ferramenta BaseX onde encontramos algumas dificuldades na sua manipulação, nomeadamente na criação e associação da base de dados com os vários documentos XML que foram criados. A linguagem XQuery desenvolvida para respostas aos pedidos propostos no enunciado também gerou alguma dificuldade devido a algumas falhas de importação de domínios que constavam nos *namespaces*, o que na REST API – Postman, por vezes não gerava o resultado esperado.

No final das *queries* desenvolvidas e validadas percebemos que algumas *tags* do XML poderiam estar definidas de forma mais clara, nomeadamente as seguintes nomenclaturas: <reservationCode>, <expertId>, <partnerId> em detrimento daquelas que constam neste projeto, <code>, <expertiseName>, <partnerName>, respetivamente.

Devido à falta de tempo para esta *milestone*, fica aqui registada a vontade de melhorar estes aspetos menos positivos para a entrega final.

Para a elaboração deste relatório, abordamos a estrutura do projeto dando alguns exemplos de excertos dos documentos criados sendo que, no próprio projeto, podemos encontrar ao longo do seu desenvolvimento várias anotações / documentação do trabalho realizado. Igualmente se aplica às *queries* desenvolvidas no BaseX.

Apesar dos contratempos com a gestão da equipa e da ferramenta de trabalho colaborativa (GitHub), consideramos que aprendemos a implementar os conhecimentos adquiridos na Unidade Curricular de Processamento Estruturado de Informação.