

Curso: Mestrado em Engenharia Biomédica**U.C.:** Processo Clínico Eletrónico**Ficha de Exercícios 04**

Docente:	Hugo Peixoto José Machado
Tema:	Introdução ao FHIR
Ano Letivo:	2024-2025 – 1º Semestre
Duração da aula:	2 horas

1. Instalação container HAPI FHIR

[1] Correr o seguinte comando:

```
docker run -p 8080:8080 \
  -e HAPI_FHIR_LOGGING_LEVEL=INFO \
  -e HAPI_FHIR_DELETE_MODE=SOFT \
  --name hapi-fhir \
  hapiproject/hapi:latest
```

Nota: Caso o comando devolva algum erro, tente perceber o problema e ultrapassar!

2. Criar um novo paciente (POST)

[1] Poderá criar um ficheiro denominado new_patient.json com o seguinte conteúdo:

```
{
  "resourceType": "Patient",
  "name": [{
    "use": "official",
    "family": "Silva",
    "given": ["João"]
  }],
  "gender": "male",
  "birthDate": "1985-03-10"
}
```

[2] Correr o seguinte comando para armazenar o paciente:

```
curl -X POST "http://localhost:8080/fhir/Patient" \
  -H "Content-Type: application/json" \
  -d @new_patient.json
```

[3] Instalar o POSTMAN e fazer extamente o mesmo procedimento que nas duas alíneas anteriores. Neste caso a criação do ficheiro será redundante!

[4] Crie 2 ou 3 novos pacientes. Teste várias opções e tente utilizar campos diferentes existentes na documentação.

3. Listar todos os pacientes existentes (GET)

[1] Utilize o seguinte comando para listar apenas um paciente da lista criada:

```
curl -X GET "http://localhost:8080/fhir/Patient?_count=1" -H "Accept: application/json"
```

[2] Utilize o POSTMAN para executar o mesmo comando.

[3] Qual o comando que deverá usar para listar todos os pacientes?

[4] Qual o id que foi atribuído aos pacientes criados?

4. Atualizar um paciente existente (PUT)

[1] Poderá criar um ficheiro denominado update_patient.json com o seguinte conteúdo:

```
{
  "resourceType": "Patient",
  "id": ":id",
  "name": [{
    "use": "official",
    "family": "Silva",
    "given": ["João", "Carlos"]
  }],
  "gender": "male",
  "birthDate": "1985-03-10"
}
```

[2] Correr o seguinte comando para armazenar o paciente:

```
curl -X PUT "http://localhost:8080/fhir/Patient/:id" \
  -H "Content-Type: application/json" \
  -d @update_patient.json
```

[3] Alterar algum campo no JSON acima a fazer o mesmo processo recorrendo ao POSTMAN.

5. Atualizar um paciente existente (PATCH)

[1] Poderá criar um ficheiro denominado patch_patient.json com o seguinte conteúdo:

```
[
  {
    "op": "replace",
    "path": "/name/0/family",
    "value": "Almeida"
  },
  {
    "op": "add",
    "path": "/telecom",
    "value": [{"system": "phone", "value": "912345678"}]
  }
]
```

[2] Correr o seguinte comando para armazenar o paciente:

```
curl -X PATCH "http://localhost:8080/fhir/Patient/:id" \  
  -H "Content-Type: application/json-patch+json" \  
  -d @update_patient.json
```

[3] Altere algum campo no JSON acima a fazer o mesmo processo recorrendo ao POSTMAN.

6. Quiz – Consulta de Histórico

[1] Ao atualizar os dados do paciente o que acontece às versões antigas? São perdidas? Conseguimos recuar no tempo? Como consultar as versões anteriores?

7. Criar uma observação para um paciente (POST)

[1] Poderá criar um ficheiro denominado observation.json com o seguinte conteúdo:

```
{  
  "resourceType": "Observation",  
  "status": "final",  
  "category": [{"coding": [{"system": "http://terminology.hl7.org/CodeSystem/observation-category", "code": "vital-signs"}]},  
  "code": {"coding": [{"system": "http://loinc.org", "code": "85354-9", "display": "Blood Pressure"}]},  
  "subject": {"reference": "Patient/:id"},  
  "valueQuantity": {"value": 120, "unit": "mmHg"}  
}
```

[2] Correr o seguinte comando para armazenar o paciente:

```
curl -X POST "http://localhost:8080/fhir/Observation" \  
  -H "Content-Type: application/json" \  
  -d @observation.json
```

[3] Utilizar o POSTMAN e fazer exatamente o mesmo procedimento que nas duas alíneas anteriores. Neste caso a criação do ficheiro será redundante!

[4] Crie 2 ou 3 novas observações para pacientes distintos. Teste várias opções e tente utilizar campos diferentes existentes na documentação.

8. Listar observações (GET)

[1] Utilize o seguinte comando para listar um paciente da lista criada:

```
curl -X GET "http://localhost:8080/fhir/Observation?subject=Patient/:id" -H "Accept: application/json"
```

[2] Utilize o POSTMAN para executar o mesmo comando.

9. Eliminar um paciente (DELETE)

[1] Correr o seguinte comando para eliminar o paciente:

```
curl -X DELETE "http://localhost:8080/fhir/Patient/:id"
```

Notas:

- a) O paciente deixa de estar acessível através do ID original.
- b) Se existirem outros recursos que fazem referência a esse paciente (Observations, Encounters, Conditions, etc.) pode devolver um erro a apagar.
- c) A maioria dos servidores não apaga fisicamente os dados, mas sim marca o paciente como “deleted” para garantir rastreabilidade.

[2] Verificar o histórico do paciente para validar se for apagado.

10. Desafio

[1] Implementar alguns dos comandos que foram desenvolvidos nas alíneas anteriores em python.