



**Universidade do Minho**  
Escola de Engenharia

Universidade do Minho  
Mestrado Integrado em Engenharia Informática

# Interoperabilidade Semântica

## Ficha 1

02 abril 2021



Tânia Rocha  
A85176

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Modelação das Bases de Dados</b>	<b>4</b>
2.1	Hospital . . . . .	4
2.2	Clínica . . . . .	5
<b>3</b>	<b>Implementação dos programas</b>	<b>6</b>
3.1	Aplicação Hospital . . . . .	7
3.2	Aplicação Clínica . . . . .	7
<b>4</b>	<b>Estabelecimento da Comunicação</b>	<b>8</b>
<b>5</b>	<b>Conclusão e Discussão</b>	<b>9</b>
<b>6</b>	<b>Referências Bibliográficas</b>	<b>10</b>

# 1 Introdução

A presente ficha enquadra-se na unidade curricular de Interoperabilidade Semântica, na qual foi proposta a construção de duas bases de dados e respetivos programas, correspondentes a um hospital e a uma clínica de exames. O principal objetivo centra-se na comunicação entre ambos os programas em contexto hospitalar.

Numa primeira instância, torna-se imprescindível a compreensão de mensagens do tipo *HL7*, uma vez que estas representam a estrutura adotada para a comunicação em situação hospitalar.

Para solucionar da melhor forma o desafio apresentado, será fundamental a criação de ambas as bases de dados relacionais na ferramenta *MySQL Workbench* e, posterior implementação dos programas na linguagem de programação JAVA. De modo a estabelecer a conexão entre ambos os programas foi utilizada a ferramenta de comunicação *Mirth*.

## 2 Modelação das Bases de Dados

Numa fase inicial foram desenvolvidas ambas as bases de dados (Hospital e Clínica), tendo em conta os constituintes principais necessários para o funcionamento de ambos os programas. De seguida, é visível o esquema lógico, bem como a explicação de cada um dos modelos.

### 2.1 Hospital

Para a modelação da base de dados do Hospital, foram consideradas as entidades Paciente, Consulta, Pedido e Registo Histórico. A seguir são listadas as entidades e respetivos atributos.

- **Pedido:** Para a entidade pedido, foram considerados necessários os atributos: **mensagem**, que representa a mensagem no formato *HL7*; **estado**, que descreve o estado do pedido, ou seja, se este está pendente, cancelado ou aceite; **data**, que informa a data de criação do pedido; o **relatório**, que permite guardar o relatório gerado após a ocorrência de um exame; o **código do exame** e a sua **descrição**. Nesta tabela foi também colocado o **id da Consulta**, uma vez que os pedidos de exame são feitos numa consulta e uma consulta pode fazer vários pedidos.
- **Consulta:** Para esta entidade apenas foram considerados a **descrição** da consulta, por exemplo "dermatologia" e o **id do Paciente** que representa o identificador do paciente que esteve na consulta.
- **Paciente:** Nesta entidade são tidos como atributos informações relativas ao paciente como, por exemplo nome, data de Nascimento, número de Processo, etc.
- **RegistoHistorico:** Esta entidade representa o histórico de pedidos. Nela estão presentes os atributos **mensagem** (mensagem *HL7*) e **estado** de um determinado pedido numa determinada data.

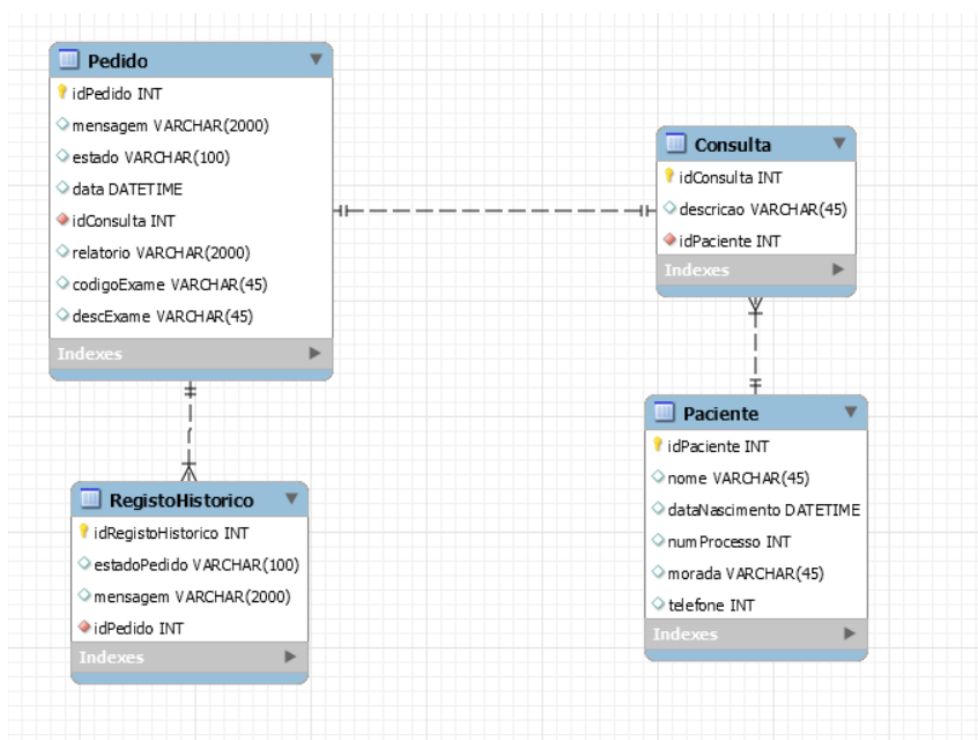


Figura 1: Modelo Lógico Hospital

## 2.2 Clínica

Para a modelação da base de dados da Clínica, foram consideradas as entidades Paciente, Exame, Pedido e Registo Histórico. A seguir são listadas as entidades e respetivos atributos.

- **Pedido:** Para esta entidade, foram considerados necessário os atributos: **mensagem**, que representa a mensagem no formato *HL7*; **estado**, que descreve o estado do pedido, ou seja, se este está pendente, cancelado ou aceite; **data**, que informa a data de criação do pedido; o **relatório**, que permite guardar o relatório gerado após a ocorrência de um exame; o **código do exame** e a sua **descrição**.
- **Exame:** Para esta entidade apenas foi considerado o **código do ato**, o **ato** (exame), um **id externo** que representa pedidos de consultas externas, ou seja, de diferentes hospitais. Como chaves estrangeiras é apresentado o **id do Paciente** que representa o identificador do paciente que efetuou o exame e o **id do Pedido**, uma vez que um pedido contém vários exames.
- **Paciente:** Nesta entidade são tidos como atributos informações relativas ao paciente como, por exemplo nome, **data de Nascimento**, **número de Processo**, etc.
- **RegistoHistorico:** Esta entidade representa o histórico de pedidos. Nela estão presentes os atributos **mensagem** (mensagem *HL7*) e **estado** de um determinado pedido numa determinada data.

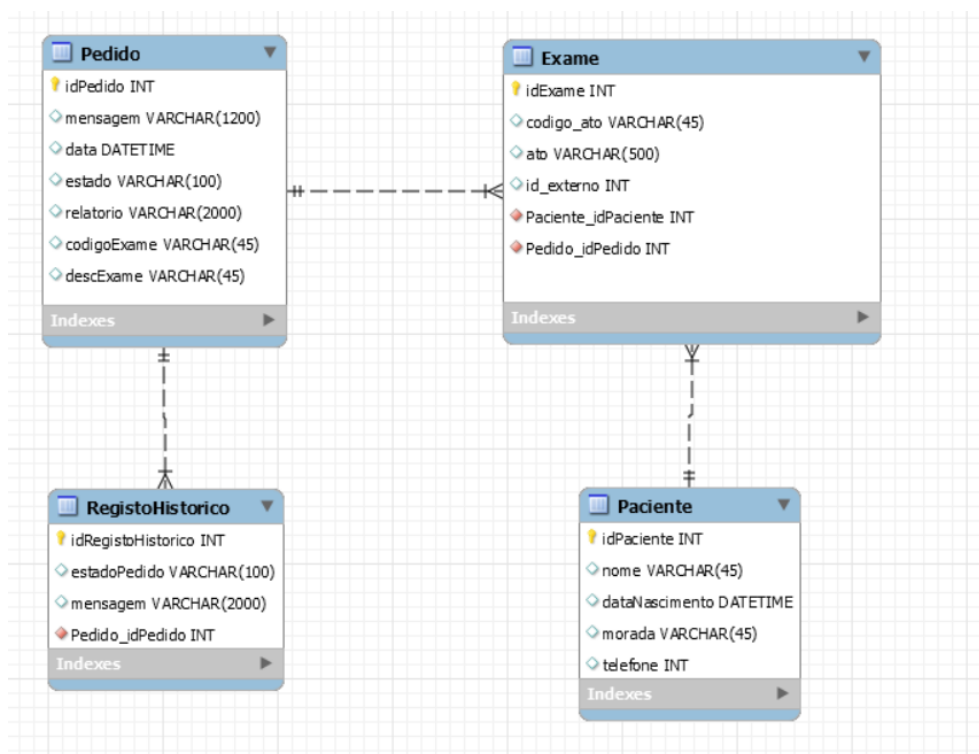


Figura 2: Modelo Lógico Clínica

### 3 Implementação dos programas

Para o desenvolvimento de ambas as aplicações utilizou-se a linguagem de programação JAVA. É de salientar que, a ligação do Java ao *schema* da base de dados foi realizada através do jar JDBC, com o objetivo de existir a permissão da inserção e atualização diretamente na base de dados. Realça-se ainda o facto de ambas as aplicações possuírem uma interface via terminal.

Com o intuito de construir mensagens *HL7* recorreu-se à biblioteca *Hapi*, visto que esta permite a criação de todos os segmentos fundamentais para uma mensagem:

- **MSH:** O segmento *HL7* MSH (*Message Header*) está presente em cada tipo de mensagem *HL7* e define a origem, a finalidade, o destino da mensagem e certas especificações de sintaxe, como delimitadores e conjuntos de caracteres. É sempre o primeiro segmento na mensagem *HL7*.
- **PID:** O segmento PID transpõe as informações de identificação do paciente. Este segmento contém identificação permanente do paciente e informações demográficas que, na maioria das vezes, não são alteradas com frequência.
- **PV1:** O segmento PV1 apresenta informações básicas de pacientes internados ou ambulatoriais. Este, comunica informações com base numa conta ou visita específica.
- **ORC:** O segmento de pedido comum (ORC) é usado para transmitir campos que são comuns a todos os pedidos (todos os tipos de serviços solicitados). Na maioria das vezes, é usado para representar uma "sessão de pedido" para um único paciente.
- **OBR:** O segmento OBR transmite informações sobre um exame, estudo / observação de diagnóstico ou avaliação que é específico para um pedido ou resultado. É usado com mais frequência em mensagens ORM (pedido) e ORU (resultado de observação).
- **OBX:** O segmento OBX é usado para transmitir uma única observação ou fragmento de observação. Ele representa a menor unidade indivisível de um relatório. O seu objetivo principal é levar informações sobre observações em mensagens de relatório.

Como exemplo de recurso à biblioteca *Hapi*, é visível na figura seguinte a criação do *header* da mensagem.

```
private void createMshSegment(String currentDateTimeString) throws DataTypeException {
    MSH mshSegment = _ormMessage.getMSH();
    mshSegment.getFieldSeparator().setValue("|");
    mshSegment.getEncodingCharacters().setValue("~\\&");
    mshSegment.getSendingApplication().getNamespaceID().setValue("AppHospital");
    mshSegment.getSendingFacility().getNamespaceID().setValue("Hospital");
    mshSegment.getReceivingApplication().getNamespaceID().setValue("AppClinica");
    mshSegment.getReceivingFacility().getNamespaceID().setValue("Clinica");
    mshSegment.getDateTimeOfMessage().getTimeOfAnEvent().setValue(currentDateTimeString);
    mshSegment.getMessageControlID().setValue(getSequenceNumber());
    mshSegment.getVersionID().getVersionID().setValue("2.4");
}
```

Figura 3: Criação do *header* da mensagem *HL7*

A combinação de todos estes segmentos gera as mensagens *HL7*. Por fim, as mensagens são armazenadas em ficheiros.

```
MSH|^~\&|AppHospital|Hospital|AppClinica|Clinica|20210326103235||ORU^R01^ORU_R01|123420210326103235|P|2.4
PID|||123||^Joseph Joestar||||America
PV1||O
ORC|NW||||||20210326103235
OBR|2||M123^Exame ao Torax
OBX|1|TX||Exame ao Torax
OBX|2|TX||Encontra-se tudo bem com o paciente. Não há sinais fora do normal no Torax.
OBX|3|TX||Relatório validado por: Joaquim Alberto
```

Figura 4: Exemplo de uma mensagem armazenada em ficheiro

### 3.1 Aplicação Hospital

Esta aplicação permite, via terminal, registar pedidos de exame médico. Para tal, são pedidas informações do exame a realizar, como o código e a descrição, assim como o identificador do paciente que vai realizar o exame. Posteriormente, com base nestas informações e com recurso à biblioteca *Hapi*, são gerados os vários segmentos das mensagens (explicitados anteriormente), com exceção do segmento *OBX* que está presente apenas após o exame. E, como referido acima, as mensagens são armazenadas em ficheiros, sendo que cada ficheiro contém no nome o identificador do pedido.

A aplicação também permite fazer o cancelamento dos pedidos efetuados, bem como a visualização do estado dos mesmos e o histórico de consultas. Por último, possibilita a visualização dos relatórios de exame recebidos pela clínica.

```
Insira 1 para visualizar o histórico de consultas
Insira 2 para efetuar pedido
Insira 3 para cancelar pedidos
Insira 4 para visualizar estado dos pedidos
Insira 5 para visualizar o relatório recebido da clínica
Insira 6 para sair
```

Figura 5: Menu da aplicação Hospital

### 3.2 Aplicação Clínica

Nesta aplicação existem as opções de visualização do estado dos pedidos recebidos da aplicação Hospital, bem como a aceitação ou cancelamento dos mesmos e, posterior, escrita e publicação do relatório após o exame. Neste caso, já é utilizado o segmento *OBX* na criação das mensagens, pois este permite o registo do relatório. Tal como feito na aplicação Hospital, todas as mensagens *HL7* geradas são armazenadas em ficheiros.

```
Insira 1 para ver estado dos pedidos
Insira 2 para cancelar pedidos
Insira 3 para aceitar pedidos
Insira 4 para escrever e publicar relatório de exames
Insira 5 para sair
```

Figura 6: Menu da aplicação Clínica

## 4 Estabelecimento da Comunicação

Com principal foco no estabelecimento da comunicação entre as duas aplicações desenvolvidas tornou-se fundamental o recurso à ferramenta Mirth.

O Mirth é um mecanismo de interface *HL7* de plataforma cruzada, open-source, que permite o envio bidirecional de mensagens *HL7* em diferentes protocolos standard entre sistemas e aplicações sob múltiplos modos de transportes.

Deste modo, criaram-se quatro canais (um par de canais de envio e receção de mensagens para cada aplicação). Relativamente aos canais de envio, estes encontram-se conectados a uma directoria que recolhe informação que é transmitida segundo o protocolo TCP. Já, os canais de receção funcionam de forma inversa, visto que apresentam um canal TCP com a mesma porta e endereço que os canais de envio, porém é onde ocorre a constante escuta da informação enviada e onde estão conectados a uma directoria para a qual escrevem essa mesma informação. Por fim, é de realçar que a informação trocada (enviada e recebida) entre estes canais são ficheiros de texto contentores de mensagens *HL7*.



## 5 Conclusão e Discussão

No final da implementação deste projeto, o grupo que desenvolveu o mesmo, concluiu que foram desenvolvidas tudo que se tinha como objetivo e de uma maneira simples e que engloba as ferramentas que se tinham como objetivo utilizar para a implementação.

Tendo como objetivo deste trabalho a implementação de duas aplicações, nomeadamente, hospital e clínica, é de alta importância que se tenha em conta um dos principais requisitos que estas devem ter, isto é, troca de dados e informação entre as mesmas relativamente aos pacientes, tipos de exames, médicos, entre outros. Daqui é então conclusivo que a Interoperabilidade é de maior importância e prioridade aquando do desenvolvimento de aplicações neste contexto.

Para tal foi então utilizado mensagens HL7 (Health a o grupo Level Seven) que se trata de um protocolo de transmissão de mensagens de contexto hospitalar, sistemas administrativos e bases de dados médicas. Este protocolo é utilizado em contexto de situações reais pois permite manter um equilíbrio entre as plataformas a contactar no sentido em que as mesmas sofrem constantes evoluções e alterações. Após uma pequena discussão em grupo foi possível concluir variadas vantagens da utilização deste protocolo, nomeadamente, permitir desenvolver e publicar padrões aprovados pela ANSI, permite promover a utilização standard dentro dos domínios da área da saúde e dos controladores e reguladores a nível internacional, promover serviços de certificação e aceitação internacional.

Esta ferramenta foi permitida utilizar como sendo open-source com o auxílio da biblioteca HAPI, também open-source, o que, mais uma vez, demonstra a facilidade da sua utilização.

A outra ferramenta utilizada, o Mirth, permitiu o auxílio na manutenção de interoperabilidade das aplicações permitindo fazer a comunicação entre as mesmas. Foram concluídas algumas vantagens desta aplicação como por exemplo, o mesmo ser open-source o que ajuda a alcançar outros projetos open-source, nomeadamente, a biblioteca HAPI como utilizada, permite manter segurança devido à sua natureza open-source que leva a uma maior comunidade que trabalha e analisa em conjunto vulnerabilidades que a ferramenta possa ter, a sua extensibilidade é também uma grande vantagem visto que pode ser facilmente utilizada para se integrar noutros projetos também na área da saúde. Uma das maiores vantagens é também o caso de ser uma ferramenta comprovada, isto é, já processou milhares de mensagens e continua em uso em centenas de ambientes e produção.

Apesar de algumas dificuldades encontradas quanto à utilização do Mirth, é possível concluir que a tarefa foi implementada com sucesso e ainda permitiu o conhecimento das ferramentas anteriormente indicadas e da sua importância no contexto hospital.

## 6 Referências Bibliográficas

<https://hl7-definition.caristix.com/v2/HL7v2.5/Segments>

<https://hapifhir.github.io/hapi-hl7v2/>