



Universidade do Minho
Escola de Engenharia

Universidade do Minho
Mestrado Integrado em Engenharia Informática

Interoperabilidade Semântica

Ficha 2

6 de maio de 2021



Tânia Rocha
A85176

Conteúdo

1	Introdução	3
2	REST API	4
2.1	Implementação dos programas	4
2.1.1	Aplicação Hospital	5
2.1.2	Aplicação Clínica	6
3	SOAP	7
4	Conclusão e Discussão	9

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.

De modo a garantir a comunicação entre as duas aplicações, bem como a representação dos valores presentes nas bases de dados, tornou-se fundamental a criação de uma REST API. É de salientar ainda que ambas as bases de dados utilizadas são as mesmas criadas aquando da resolução da ficha anterior.

Todavia, para esta ficha, pretende-se a permuta da comunicação via mensagens *HL7* pela criação de dois *JSON servers* funcionais a partir de pedidos *POST*, *GET* e *PUT*.

Como supramencionado, para solucionar da melhor forma o desafio apresentado, será imprescindível a utilização de ambas as bases de dados relacionais criadas à priori na ferramenta *MySQL Workbench* e posterior implementação dos programas na linguagem de programação *JavaScript*. A fim de estabelecer a conexão entre ambos os programas foram usados dois *JSON servers*.

2 REST API

2.1 Implementação dos programas

Numa primeira fase inicializou-se a implementação da aplicação da Clínica e do Hospital de uma forma semelhante à estratégia utilizada na ficha anterior. Ambas com um módulo de interface Web permitido pela *framework NodeJS* com conexão às suas respectivas bases de dados *MySQL*, com o intuito de, posteriormente, armazenar nas mesmas todas as informações e dados relativos a pedidos e exames a efetuar e/ou efetuados.

Para além do referido, a cada aplicação encontra-se também associado um *json-server*, onde são guardados os pedidos e exames a fim de garantir/possibilitar a comunicação entre as duas aplicações. Esta comunicação, com suporte a esta *framework*, é efetuada através de pedidos GET, POST e PUT (a partir de ambas aplicações).

A título de exemplo, enquadra-se a situação de geração de um pedido por parte do Hospital. Esse pedido, para além de armazenado na sua base de dados, é também inserido no *json-server* do Hospital, através de um pedido POST.

Já do lado da Clínica, aquando da listagem dos pedidos, é efetuado um pedido GET ao *json-server* do Hospital, o que permite a obtenção dos pedidos efetuados. Por sua vez, estes são guardados na base de dados da Clínica e é realizado um pedido POST para inserir estes pedidos no *json-server* da Clínica, de modo a que este contenha também todas as informações necessárias para continuar a efetuar tarefas.

Este procedimento garante que a Clínica é capaz de aceitar e/ou cancelar pedidos realizados pelo hospital, bem como fazer um pedido do tipo PUT para atualizar o estado dos mesmos, tanto no *json-server* da Clínica como no do Hospital.

Este comportamento das aplicações que tem vindo a ser descrito é constante para a realização e listagem de exames, assim como para a criação dos respetivos relatórios.

De seguida encontra-se representada uma esquematização subjacente ao funcionamento das aplicações e respetivos pedidos aos *json-server*, construída de forma a proporcionar uma melhor compreensão da estratégia arquitetada.

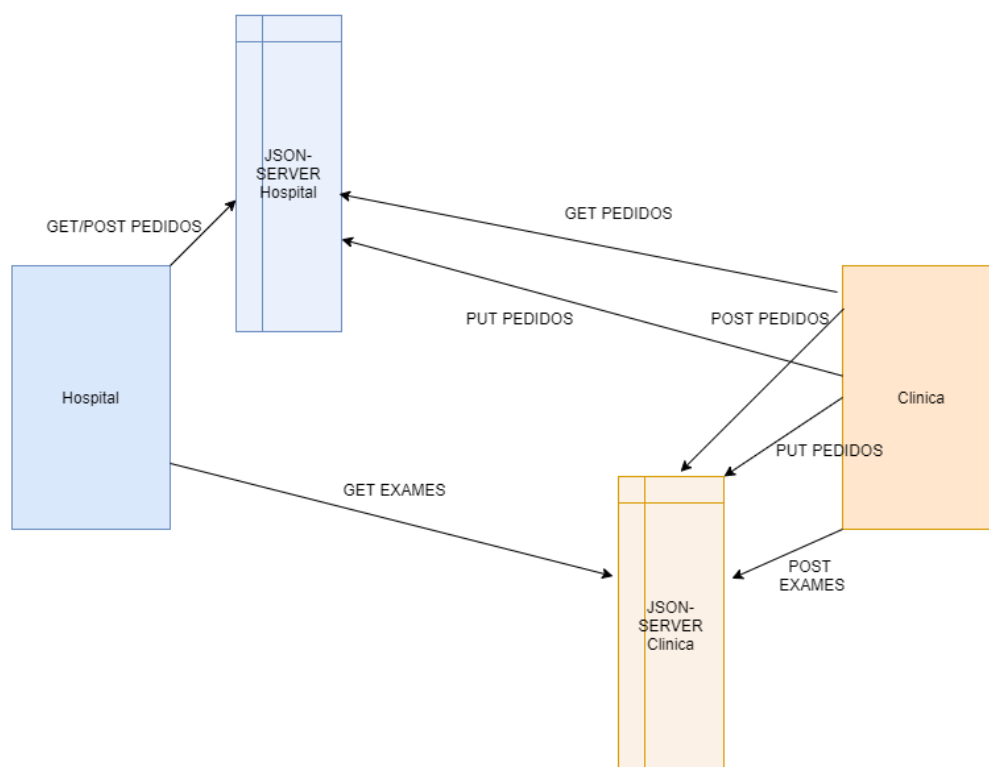


Figura 1: Esquema REST.

2.1.1 Aplicação Hospital

A aplicação do Hospital permite a solicitação de um exame juntamente com o agendamento do mesmo para um determinado dia e hora, é também capaz de garantir a disponibilização de uma lista de exames mediante a pesquisa por um determinado dia. No que diz respeito, à utilização de um serviço para receber notificação relativa aos exames e relatórios realizados, entende-se que o envio de uma mensagem via SMS resultaria numa ótima solução, pelo que optou-se por esta implementação.

Estas funcionalidades encontram-se representadas nas figuras apresentadas de seguida.

Opções Iniciais

Hospital



Figura 2: Página inicial aplicação Hospital

Inserir Novo Pedido +				
Estado	Data	Tipo Exame	Descrição de Exame	Relatório
aceite 2021-04-28	2023-5-5 10:30:00	AB2G	Exame de Cardiologia	
pendente 2021-04-28	2029-4-6 4:0:00	MG4F	Exame de Oftalmologia	
pendente 2021-04-28	2026-5-11 10:13:00	KJU7	Análises Clínicas	
Voltar				

Figura 3: Listagem de pedidos na aplicação Hospital

Inserir Novo Pedido +				
Estado	Data	Tipo Exame	Descrição de Exame	Relatório
aceite 2021-04-28	2023-05-05T09:30:00.000Z	AB2G	Exame de Cardiologia	Após análise de resultados é concluído que paciente não apresenta qualquer anomalia.
pendente 2021-04-28	2029-4-6 4:0:00	MG4F	Exame de Oftalmologia	
cancelado 2021-04-28	2026-5-11 10:13:00	KJU7	Análises Clínicas	
Voltar				

Figura 4: Listagem de Exames e respectivo relatório

2.1.2 Aplicação Clínica

A aplicação Clínica garante o carregamento da lista de todos os exames a realizar no presente dia, bem como a realização dos exames e consequente disponibilização dos relatórios subjacentes aos mesmos.

Estas funcionalidades encontram-se representadas nas figuras apresentadas de seguida.

Opções Iniciais

Clinica

Lista de Pedidos	Lista de Exames
------------------	-----------------

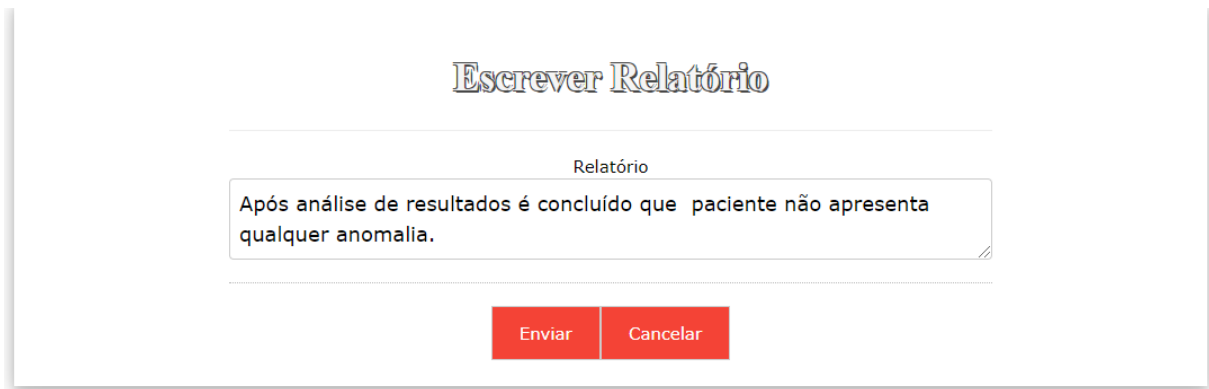
Figura 5: Página inicial aplicação Clínica

Pedidos Pendentes de Hoje					
Procure por data					
Estado	Data	Relatório	Tipo Exame	Descrição de Exame	Opções
aceite 2021-04-28	2023-5-5 10:30:00		AB2G	Exame de Cardiologia	Aceitar Cancelar
pendente 2021-04-28	2029-4-6 4:0:00		MG4F	Exame de Oftalmologia	Aceitar Cancelar
cancelado 2021-04-28	2026-5-11 10:13:00		KJU7	Análises Clínicas	Aceitar Cancelar
Voltar					

Figura 6: Listagem de pedidos na aplicação Clínica

Procure por data						
Código Exame	Exame	Médico	Data	Identificador do Paciente	Identificador do Pedido	Relatório
AB2G	Exame de Cardiologia	Alberto João Henriques	2023-5-5 10:30:00	1	1	Realizar Exame
Voltar						

Figura 7: Listagem de Exames e possibilidade de realizar o exame



Escrever Relatório

Relatório

Após análise de resultados é concluído que paciente não apresenta qualquer anomalia.

Enviar Cancelar

Figura 8: Escrever relatório do exame realizado

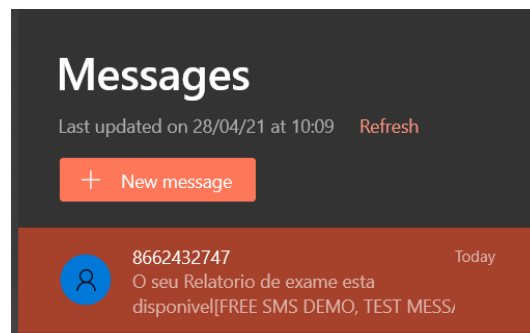


Figura 9: Notificação de exame realizado

3 SOAP

SOAP (*Simple Object Access Protocol*) é um protocolo baseado em XML para troca de informações num ambiente distribuído, ou seja, é utilizado para a troca de mensagens entre aplicações distribuídas pela rede. Aplicações (ou “*Web services*”) estas que possuem uma interface de acesso simples e bem definida.

SOAP revela-se como o elemento principal de infra-estrutura de Web Services, bem como um factor fundamental para o funcionamento dos mesmos. Para além do mencionado, o SOAP facilita a interoperabilidade entre objetos e componentes distribuídos e nem precisamos de saber como estão implementados.

Por outro lado, REST (*Representational State Transfer*) é um protocolo cliente/servidor sem estado, em que cada mensagem HTTP contém toda a informação necessária para compreender o pedido.

É de conhecimento geral que ambos os protocolos comportam inúmeros benefícios, todavia são as suas diferenças que funcionam como meio de seleção para a utilização de um face a outro dependendo da problemática.

A presente ficha assentou em REST devido às vantagens deste protocolo em prol do SOAP, não desfazendo das várias qualidades que o SOAP também apresenta. Pelo facto do SOAP ser um protocolo igualmente capaz e útil, irão ser descritas as modificações consideradas essenciais para proceder à transição de REST para SOAP.

Desta forma, uma das principais diferenças é a utilização de *XML* ao invés de *JSON*. A mudança de linguagem também seria um ponto a modificar, dado que o SOAP não interage da melhor forma com *JavaScript*.

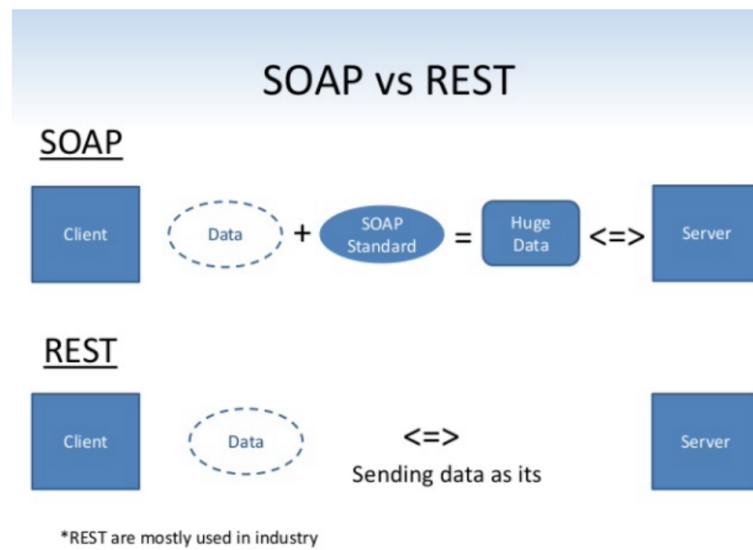


Figura 10: Web Services - SOAP e REST

4 Conclusão e Discussão

Após o desenvolvimento deste projeto foi possível, através de uma discussão em grupo, obter um conjunto de conclusões neste contexto.

Numa primeira parte, foram concluídas as principais diferenças entre as APIs desenvolvidas na fase anterior comparativamente com a corrente. No trabalho anterior a comunicação foi centrada nas mensagens de formato hl7 que continham a informação necessária para ambas as APIs envolvidas. Estas mensagens eram comunicadas através da ferramenta Mirth que por sua vez transferia estes ficheiros de mensagens para o respectivo remetente.

Por outro lado, nesta fase, foi utilizado como elo de comunicação ficheiros em formato json que podiam ter as suas informações partilhadas entre a clínica e o hospital através de protocolos GET, POST e PUT com o auxílio de servidores do tipo *json-server* gerados a partir dos respetivos ficheiros JSON.

Quando comparada a facilidade e o nível de trabalho exigido na implementação destas duas fases, foi concluído que a fase atual, a segunda fase, foi de uma implementação de menor dificuldade isto porque, no caso da utilização de mensagens por hl7 era necessário respeitar o formato e constituição das mesmas, o que não era uma tarefa trivial devido à quantidade de informação que estas continham numa formatação não tão legível como no caso de ficheiros do tipo json. Quando à comunicação efetuada por mirth na fase anterior, também é possível afirmar que, comparativamente com os protocolos de transferência utilizados com o auxílio de *json-servers*, era mais restrita e complicada, visto que a alternativa torna mais interpretável a informação comunicada e esta contém uma abrangente comunidade ativa que ajuda tanto na resolução de problemas como na sugestão de novas implementações. Para além disto, a própria ferramenta *json-server* é altamente indicada para o desenvolvimento de aplicações REST.

Ainda no contexto da implementação deste projeto, recorreu-se a novas aplicações auxiliares de envio de mensagens sms para os pacientes aquando o término dos respectivos relatórios de exames. Esta pequena implementação final da aplicação permitiu obter novos conhecimentos de como utilizar este tipo de sistemas em vida real, afirmando-se de grande importância no que se trata de rendimentos em contexto real hospital visto que permite que não ocorram casos de fraude de consultas não existentes, uma vez que há pacientes reais a serem notificados, e permite que receitas ou exames sejam enviados tecnologicamente o que permite uma diminuição da lotação em geral do hospital, entre outras melhorias no funcionamento do mesmo.

Numa outra nota, foi também sugerido para este projeto que se indicassem quais as diferenças quanto à implementação de uma API Rest com uma API SOAP. As APIs REST operam por meio de uma interface consistente com o objetivo de aceder a recursos identificados. É frequentemente utilizado quando pretendemos expor uma API pública na Internet. O SOAP, por outro lado, expõe os componentes da lógica da aplicação como serviços em vez de dados. O REST acede aos dados enquanto o SOAP executa operações por meio de um conjunto de mensagens padrões. Ainda assim, na maioria dos casos, REST ou SOAP podem ser usados para atingir o mesmo resultado, com algumas diferenças de implementação, nomeadamente, a facilidade de implementação na linguagem de programação *javascript*.

Após retiradas as conclusões e implementadas todas as tarefas propostas neste contexto, dá-se então por terminado este relatório, cujo desenvolvimento permitiu alargar os horizontes no que se trata da implementação de novos sistemas e ferramentas de desenvolvimento de aplicações, não só em contexto hospital, mas como também num contexto geral programacional.