

Software: **Gestão de Serviços Médicos**

AUTORES:	Carolina Araújo - 33348		
	Nilsa Melo - 33766		
DATA:	26 Novembro 2018		
DISTRIBUIÇÃO:	Alessandro Moreira	UFP	(1 cópia)
	Feliz Gouveia	UFP	(1 cópia)

DOCUMENT STATUS SHEET

DOCUMENT STATUS SHEET			
1. DOCUMENT TITLE: Software: Gestão de Serviços Médicos			
2. ISSUE	3. REVISION	4. DATE	5. REASON FOR CHANGE

DOCUMENT AMENDMENT SHEET

ISSUE	PAGE	COMMENT

Índice

1.	Introdução.....	7
1.1.	Objetivos.....	7
1.2.	Âmbito da aplicação.....	7
1.3.	Definições e acrónimos.....	7
1.4.	Referências.....	7
2.	Descrição geral.....	7
2.1.	Arquitetura da aplicação.....	8
3.	Descrição dos módulos.....	8
3.1	Web service 1.....	8
3.2	Web service 2.....	8
4.	Interface utilizador.....	9
4.1	Protótipo interface para o Utente.....	9
4.2	Protótipo interface para o Funcionário.....	9
5.	Escolhas do projeto.....	10
6.	Normas e boas práticas.....	10
6.1.	Documentação.....	11
7.	Descrição da API.....	13

Índice de Figuras

Arquitetura do software.....	8
Interface Utente.....	9
Interface Funcionário.....	9
Casos de Uso (WS1).....	11
Diagrama de Classes(WS2).....	11
Diagrama de Classes(WS1).....	12

1. Introdução

Este documento descreve o projeto de Arquitetura e o projeto Detalhado do software : Gestão de Serviços Médicos. Esta aplicação tem como finalidade ajudar na gestão administrativa dum ou mais clínicas, permitindo também aos utentes gerir as suas próprias consultas. De notar que deverá ser possível aplicar este programa para outras áreas.

1.1. Objetivos

Este documento segue as recomendações ESA-PSS-05-04 (Projeto de Arquitetura) e ESA-PSS-05-05 (Projeto Detalhado), para descrever projeto de Arquitetura e o projeto Detalhado do programa Gestão de Serviços Médicos. A aplicação necessita de um serviço de dados, cuja existência se assume e cujos requisitos serão também descritos.

1.2. Âmbito da aplicação

O software desenvolvido oferece um auxílio à Gestão de Serviços Médicos proporcionando assim uma maior interatividade entre o utilizador e a própria clínica.

1.3. Definições e acrónimos

WS1 - Web Service 1

WS2 - Web Service 2

1.4. Referências

ESA PSS-05-04, Guide to the software architectural design phase, Issue 1 Revision 1, March 1995.

ESA PSS-05-05, Guide to the software detailed design and production phase, Issue 1 Revision 1, March 1995.

2. Descrição geral

O software deve permitir ao utente gerir as suas consultas (marcar, editar e cancelar), tal como fazer pesquisas de disponibilidade de horários aplicando alguns filtros (especialidade, médicos e clínica). Em relação ao funcionário deve ser possível gerir informação administrativa como gerir médicos (inserir, editar e apagar), gerir consultas e fazer pesquisas. Numa última fase o programa deverá

estender para outros serviços como consultas com contabilistas, advogados, entre outros.

2.1. Arquitetura da aplicação

A arquitectura usada para o desenvolvimento do projeto é a BROKER, como podemos ver na figura 1.

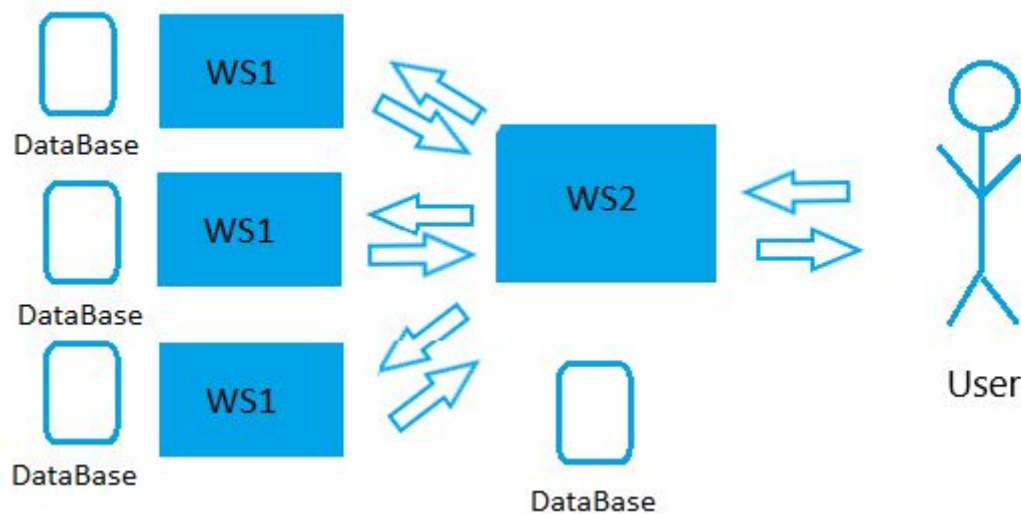


Figura 1 - Arquitetura do Software

Os módulos correspondem a grupos de funcionalidades, e são detalhados a seguir.

3. Descrição dos módulos

3.1 Web service 1

Cada web service 1 corresponde a uma clínica diferente onde teremos class, tais como, appointments, doctor, specialty, schedule. Todas estas funcionalidades permitem que seja possível a gestão administrativa da clínica. Os dados serão gravados na base de dados possibilitando assim ao sistema uma coerência de dados em tempo real.

3.2 Web service 2

O web service 2 corresponde a agregação da cadeia de clínicas. É neste serviço que vai ser possível a interação do utilizador com as funcionalidades do programa. Este serviço também terá a sua base de dados onde vai ser possível registar as clínicas.

4. Interface utilizador

4.1 Protótipo interface para o Utente

O protótipo de interface para o usuário é composto por várias seções:

- ClientInterface:** Um formulário de login com campos para "Username:" e "Password:", e um botão "Login".
- Username:** Uma seção de cabeçalho para a interface principal.
- Consultas:** Uma tabela com 7 colunas: "Clinica", "Medico", "Esp.", "Sala", "Data", "Hora" e uma coluna vazia. Abaixo da tabela há três botões: "Marcar", "Editar" e "Apagar".
- Pesquisar:** Um formulário de busca com campos para "Clinica:", "Especialidade:" e "Medico:", e um botão "Search".
- Results:** Uma segunda tabela idêntica à de "Consultas", com 7 colunas: "Clinica", "Medico", "Esp.", "Sala", "Data", "Hora" e uma coluna vazia.
- Marcar:** Um formulário lateral com campos para "Clinica:", "Medico:", "Especialidade:", "Data:" e "Hora:", e um botão "Marcar".
- Editar:** Um formulário lateral idêntico ao de "Marcar", com um botão "Editar".
- Apagar:** Um formulário lateral com campos para "Clinica:", "Medico:", "Especialidade:", "Data:" e "Hora:", e um botão "Apagar".

Figura 2 - Interface Utente

4.2 Protótipo interface para o Funcionário

EmployeeInterface

Username:

Password:

Login

Gerir Medicos

Consultas

Clinica	Medico	Esp	Sala	Data	Hora

Marcar Editar Apagar

Pesquisar

Clinica:

Especialidade:

Medico:

Search

Results

Clinica	Medico	Esp	Sala	Data	Hora

Marcar

Clinica:

Medico:

Especialidade:

Data:

Hora:

Marcar

Editar

Clinica:

Medico:

Especialidade:

Data:

Hora:

Editar

Apagar

Clinica:

Medico:

Especialidade:

Data:

Hora:

Apagar

Gerir medicos

Lista Medicos

Nome	Esp

Inserir Editar Apagar

Figura 3 - Interface Funcionário

5. Escolhas do projeto

6. Normas e boas práticas

6.1. Documentação

Serão representados aqui alguns dos modelos criados para a modelização do projeto.

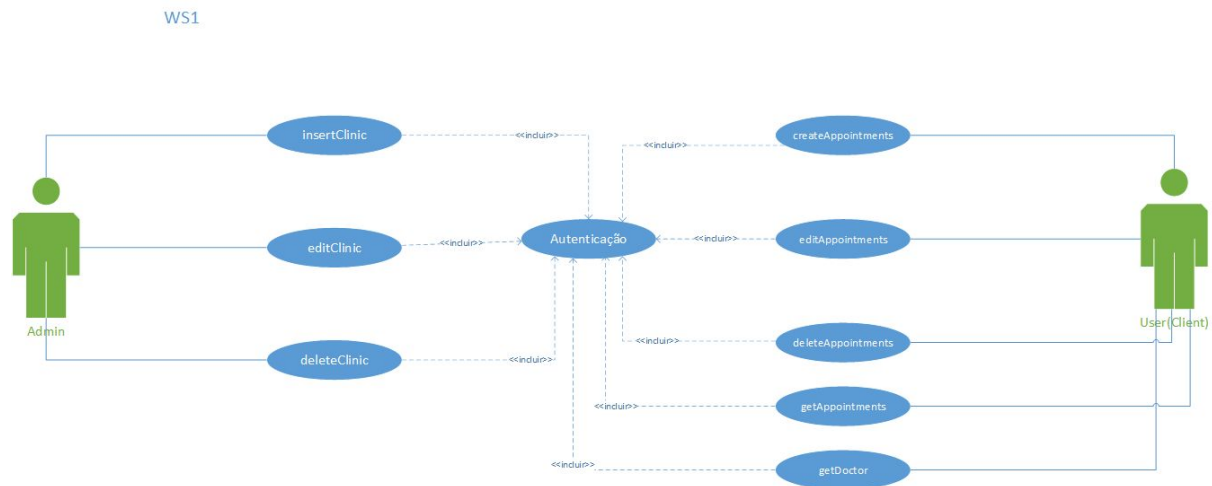


Figura 4 - Casos de Uso (WS1)

A modelização do WS1 teve por base a representação dos requisitos funcionais obtidos e a sua representação num diagrama de casos de uso (Figura 4).

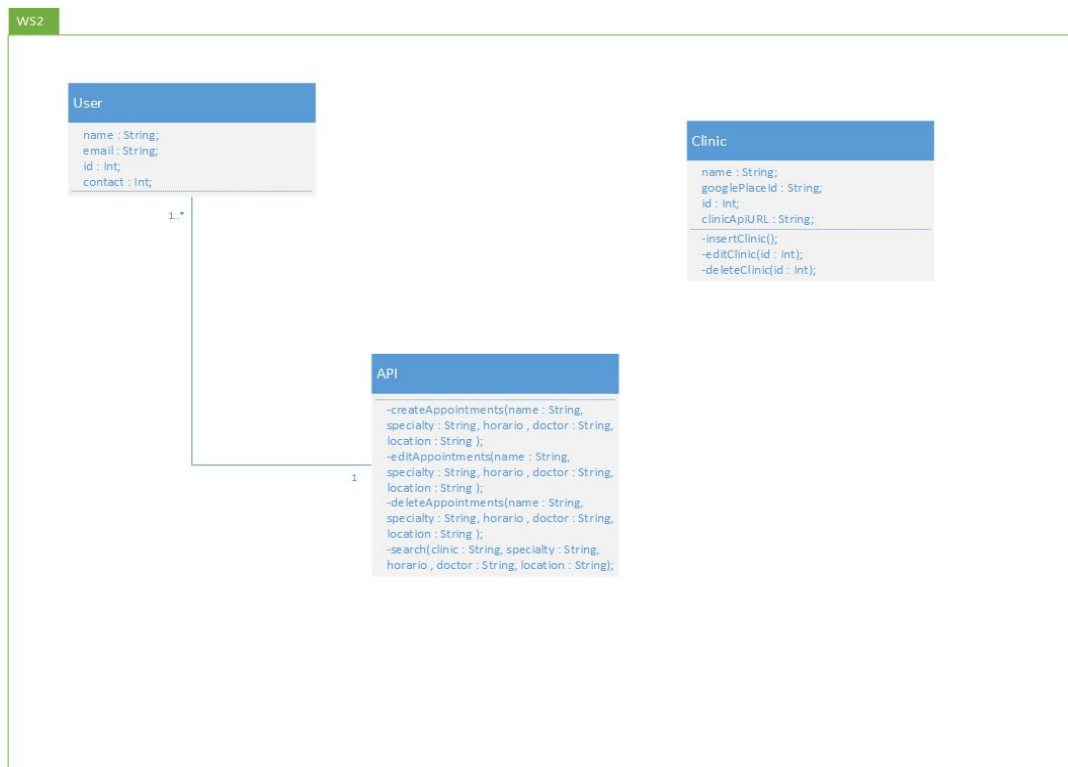


Figura 5 - Diagrama de Classes(WS2)

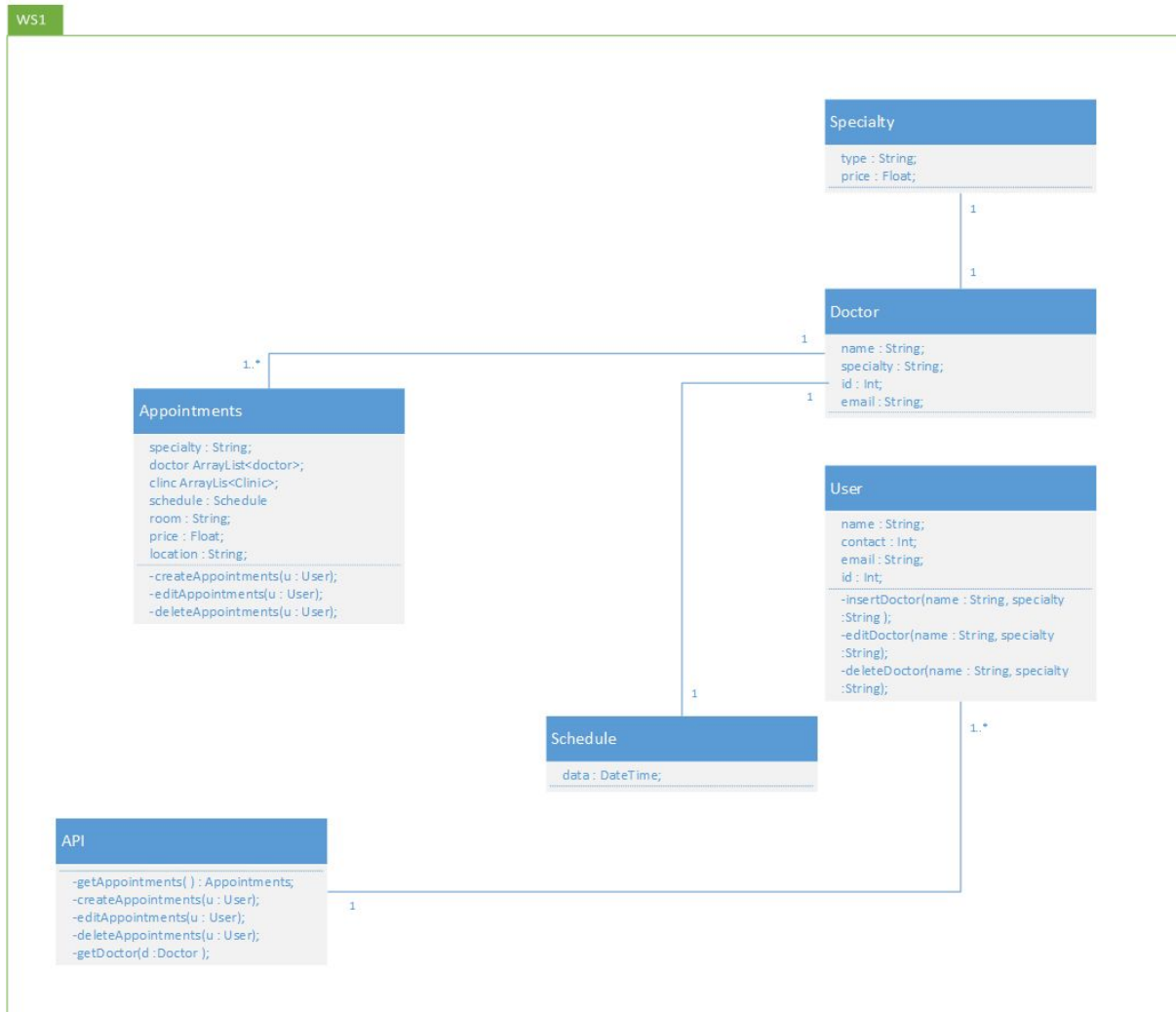


Figura 6 - Diagrama de Classes (WS1)

Todos os restantes modelos encontram-se em anexos.

7. Descrição da API

API REST é um tipo de arquitetura que define um conjunto de restrições e propriedades baseados em HTTP.

Num web service RESTful quando são feitos pedidos ao URI de recurso a resposta que pode estar num formato do tipo XML, HTML, JSON, etc... . Quando o HTTP tem os seguintes métodos GET, POST, PUT, DELETE e outros métodos HTTP CRUD pré-definidos.

Esta abordagem usa Controllers, que recebem pedidos HTTP do controller do spring e simplesmente delega esses pedidos para uma camada de serviços.