



Ricardo Cordeiro de Sousa

# Desenvolvimento de uma App para Android para fins médicos

Setembro 2018



UNIVERSIDADE DE COIMBRA





FCTUC FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

Mestrado Integrado Engenharia Eletrotécnica e de Computadores

# Desenvolvimento de uma App para Android para fins médicos

Ricardo Cordeiro de Sousa

Orientador:

Prof. Doutor Rui Pedro Duarte Cortesão

Júri:

Prof. Doutor Rui Paulo Pinto da Rocha (Presidente)

Prof. Doutor Marco Alexandre Cravo Gomes

Prof. Doutor Rui Pedro Duarte Cortesão (Orientador)

Dissertação para obtenção do Grau de Mestre em  
Engenharia Eletrotécnica e de Computadores.

Coimbra, setembro 2018



# Agradecimentos

Ao professor, Rui Pedro Duarte Cortesão, o meu agradecimento, pela orientação ao longo deste percurso.

A minha família, em especial aos meus pais, por tornarem este caminho possível.

Finalmente, mas não menos importantes, queria agradecer a alguns amigos que me ajudaram neste processo, sem nenhuma ordem em especial: Bruce, JP, Diana, Sílvia, André, Barry, Dona Bina, Luís, Daniel, Pedro e finalmente ao Jacinto, o meu obrigado.



# Resumo

A tecnologia na área da saúde é um estímulo para conquistar novas fronteiras e desafios. Num mundo em que a esperança média de vida tem vindo a aumentar, é necessário estar preparada para os desafios que tal conquista possa trazer.

Um dos desafios principais são as doenças crónicas. A incidência destas doenças tem vindo a crescer de tal forma, que a Organização Mundial de Saúde já as considera a epidemia silenciosa. Cancro, Diabetes, Asma, Epilepsia e HIV são apenas alguns exemplos. Certas doenças como a Asma ou a Diabetes não são consideradas altamente mortais, mas diminuem consideravelmente a qualidade de vida do doente.

Uma ferida é uma quebra na barreira protetora da pele e em alguns casos pode ser considerada uma doença crónica. Em casos normais estes ferimentos são tratados naturalmente, mas em casos mais complicados é necessária assistência médica, podendo desencadear um processo longo e complicado. Em tais casos, é necessária uma monitorização constante para certificar que o processo de cicatrização decorre sem complicações. O sucesso deste processo requer uma documentação cuidada do estado da ferida bem como do próprio doente.

A manutenção destes registos é uma tarefa longa e complexa. Com este problema em mente, foi desenvolvido um sistema capaz de gerir toda esta informação. O sistema foi desenvolvido usando tecnologias Web permitindo acesso em qualquer parte do mundo.

Este sistema pretende ajudar profissionais de saúde e consequentemente pacientes tornando todo este processo um pouco mais fácil.

**Palavras-chaves:** Ferida, Android, WebSockets, Sql Server





# Abstract

Technology can be an inspiration to expand the existing limits and conquer new barriers in medicine.

In a world where our lifespan is growing, it is necessary to carefully plan our future, expecting the known and unknown challenges in front of us.

One of the most concerning challenges is chronic diseases. These diseases are in a rapid growing pace, so much so that WHO (World Health Organization) considers them as a silent epidemic. Cancer, Diabetes, Asthma, Epilepsy and HIV are just a few examples. Some conditions like asthma or diabetes don't lead to a imitate death, but they take a toll on the quality of life of their patients. A wound is an injury to the living tissue and in some cases can be thought of as a chronic disease. In normal cases, it will heal by itself, however, there are cases where medical assistance is needed, leading to a complex process that can take years to fully heal.

In such cases, it is needed a constant follow-up to make sure the cicatrization process runs smoothly. The success of the cicatrization process requires a careful documentation, storing information about the patient and his wound.

Maintaining, saving and sharing such documentation can be troublesome.

With this problem in mind, it was developed a system capable of manage this type of information. This system was developed using the web's technologies allowing a full reach any time anywhere.

With this system, we want to help doctors, nurses and patients making the process a little bit easier.

**Keywords:** Wound, Android, WebSockets, Sql Server, IIS



“Se não tentares, nunca saberás”

**M.K.**



# Conteúdos

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	MOTIVAÇÃO E CONTEXTO	2
1.2	OBJETIVOS	3
1.3	ESTRUTURA DA TESE	3
<b>2</b>	<b>ESTADO DA ARTE</b>	<b>5</b>
<b>3</b>	<b>METODOLOGIA DE DESENVOLVIMENTO</b>	<b>7</b>
<b>4</b>	<b>REQUISITOS</b>	<b>9</b>
<b>5</b>	<b>ARQUITETURA DO SISTEMA</b>	<b>11</b>
5.1	CASOS DE USO	11
5.1.1	<i>Caso de Uso Paciente</i>	11
5.1.2	<i>Caso de Uso Restantes Funcionalidades</i>	12
5.2	DIAGRAMAS DE SEQUÊNCIA	12
5.2.1	<i>Diagrama de Sequência Login</i>	13
5.2.2	<i>Diagrama de Sequência Adicionar Paciente</i>	14
5.2.3	<i>Diagrama de Sequência Ver/Alterar/Remover Paciente</i>	15
5.2.4	<i>Diagrama de sequência Adicionar Ferida</i>	16
5.2.5	<i>Diagrama de sequência Ver/Alterar/Remover Ferida</i>	17
5.2.6	<i>Diagrama de Sequência Adicionar Consulta</i>	18
5.2.7	<i>Diagrama de Sequência Ver/Alterar/Remover Consulta</i>	19
5.2.8	<i>Diagrama de sequência das comunicações em tempo real</i>	20
5.3	COMPONENTES	20
5.3.1	<i>Aplicação móvel</i>	20
5.3.2	<i>Servidor</i>	21
5.3.3	<i>Base de Dados</i>	21
<b>6</b>	<b>IMPLEMENTAÇÃO MÓVEL</b>	<b>23</b>
6.1	ANDROID STUDIO	23
6.2	ACTIVITIES	24
6.3	ÁREA DA FERIDA	25
6.4	CÁLCULO DA ÁREA DA FERIDA	26
6.4.1	<i>Método de Monte Carlo</i>	26
6.4.2	<i>Guass's area formula</i>	27
6.5	COMUNICAÇÃO COM O SERVIDOR	28
6.5.1	<i>HTTP</i>	28
6.5.2	<i>Web Sockets</i>	29

<b>7</b>	<b>IMPLEMENTAÇÃO SERVIDOR E BASE DE DADOS .....</b>	<b>31</b>
7.1	SERVIDOR.....	31
7.1.1	<i>Signalr</i> .....	32
7.1.2	<i>Vídeo Chamada</i> .....	33
7.1.3	<i>Base de dados</i> .....	33
7.1.4	<i>Esquema da Base de dados</i> .....	34
<b>8</b>	<b>TESTE E MANUAL DE UTILIZADOR .....</b>	<b>35</b>
8.1	TESTES .....	35
8.1.1	<i>Testes unitários</i> .....	35
8.1.2	<i>Teste de Carga</i> .....	36
8.1.3	<i>Teste na ótica do utilizador</i> .....	36
8.2	MANUAL DE UTILIZADOR .....	37
8.2.1	<i>Login</i> .....	37
8.2.2	<i>Menu Principal</i> .....	37
8.2.3	<i>Lista de Pacientes</i> .....	37
8.2.4	<i>Adicionar Paciente</i> .....	38
8.2.5	<i>Adicionar Consulta</i> .....	38
8.2.6	<i>Calcular Área da Ferida</i> .....	38
8.2.7	<i>Notificações</i> .....	39
8.2.8	<i>Marcações</i> .....	39
8.2.9	<i>Historial de Feridas</i> .....	39
<b>9</b>	<b>CONCLUSÃO.....</b>	<b>41</b>
	<b>REFERÊNCIAS.....</b>	<b>42</b>



# 1 Introdução

A transmissão de informação é fundamental para qualquer civilização. É de tal forma importante que impulsionou vários desenvolvimentos tecnológicos, culminando na maior rede de partilha de informação, até à data, a *Internet*. Esta rede tornou-se de tal forma prevalente na nossa sociedade, que revolucionou a maneira como comunicamos.

A possibilidade de obter e transmitir informação facilmente, trouxe uma reforma a vários sectores permitindo melhoramentos de sistemas e serviços. Um dos sectores que mais beneficiou de tal reforma foi o sector da Saúde, melhorando cuidados médicos, gerando novos procedimentos, reduzindo a distância entre doentes e médicos, etc.

No entanto somos vítimas do nosso próprio sucesso. Com o aumento da esperança média de vida, outros desafios têm surgindo, como as feridas e doenças crónicas.

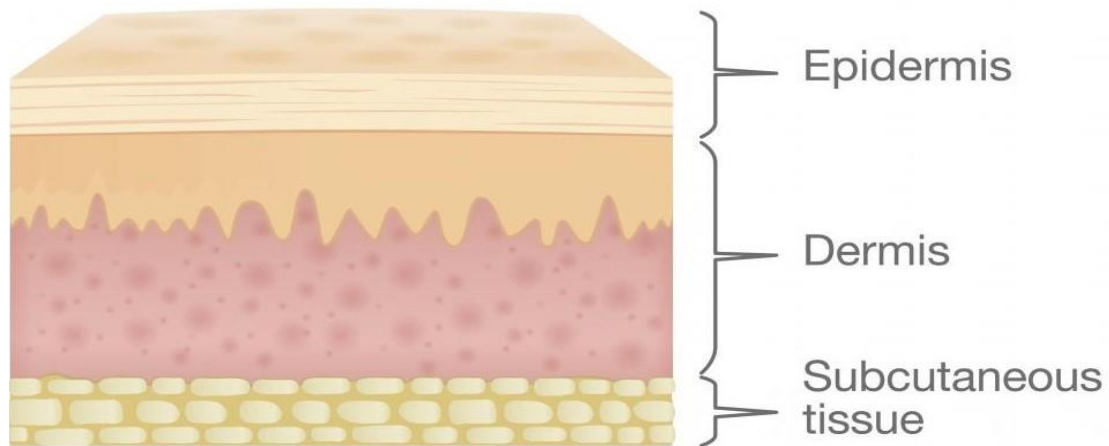
Embora doenças crónicas como o cancro, diabetes, epilepsia, obesidade sejam socialmente conhecidas existem outros fenómenos crónicos como as feridas crónicas que passam um bocado despercebidas. Foi estimado [1] que cerca de 6.5 milhões de pacientes americanos sejam afetados por estas feridas. Doenças como a diabetes e obesidade junto com uma população mais idosas, tornam estes casos mais frequentes.

Caso nada seja feito, existe a probabilidade de isto se tornar um fenómeno global com graves repercussões sociais e financeiras tanto para doentes como para a sociedade em geral.



## 1.1 Motivação e Contexto

A pele é o maior órgão do corpo humano e tem como principal função proteger o corpo do meio envolvente. É composta por 3 partes: A epiderme (parte superior), a derme (parte interior) e pela hipoderme (parte inferior). [1]



*Figura 1 Camadas da Pele*

Uma quebra nesta barreira defensiva é normalmente chamada de ferida.

Uma ferida pode ter variadas origens e conseqüentemente diferentes abordagens ao seu tratamento. Em certos casos o processo de cicatrização é algo complexo que exige um acompanhamento constante para prevenir possíveis complicações.

Nestas situações delicadas é necessário manter um registo evolutivo do processo de cicatrização (dimensões, dor, localização, odor, infecção, fotografia, etc.), facilitando o trabalho dos profissionais de saúde. O registo manual desta informação pode ser moroso, trazendo também alguns riscos como a sua perda ou a dificuldade da sua transmissão. O registo ficaria mais fácil e rápido se toda esta informação ficasse guardada num servidor, podendo ser consultada em qualquer lugar ou altura.

## 1.2 Objetivos

O principal objetivo deste trabalho é fornecer uma ferramenta aos profissionais de saúde que possa facilitar o registo e acompanhamento no processo de cicatrização de uma ferida.

Para tal será desenvolvida uma aplicação móvel, um servidor e uma base de dados para que estes interajam entre si, fornecendo ao seu utilizador a respetiva informação.

## 1.3 Estrutura da Tese

Este trabalho foi estruturado recorrendo a capítulos. No capítulo número dois é feita uma abordagem ao mercado atual, identificando potenciais concorrentes.

No capítulo três, é discutida a metodologia no desenvolvimento do software. No capítulo quatro são apresentados os requisitos. De seguida temos o capítulo cinco, onde são apresentados todos os diagramas técnicos. O capítulo seis documenta todo o processo de desenvolvimento da aplicação móvel. No capítulo sete é abordado o desenvolvimento do servidor bem como da base de dados. O capítulo oito é dedicado aos testes executados e ao manual de utilizador. Finalmente no capítulo nove é a conclusão.



## 2 Estado da Arte

Neste capítulo serão abordadas e discutidas outras aplicações/métodos usados na área do tratamento e monitorização de feridas.

Na maioria dos hospitais os métodos atualmente usados para monitorizar feridas são básicos e em alguns casos antigos. A cicatrização e o tratamento destas feridas requerem um registo constante documentando a ferida, avaliando parâmetros como a sua área e percentagem de tecido vivo.

Os registos, quando feitos, usam o formato em papel e no caso de ser necessário documentar o ferimento, recorrendo geralmente a uma máquina fotográfica e a uma régua. Este tipo de método pode trazer erros ou incertezas nas medições tornando toda a documentação um processo pouco fiável.

Naturalmente, certas empresas tentam e tentaram capitalizar esta lacuna.

Empresas como PointClickCare [11], Tissue Analytics[12] e E-Kare[13] desenvolveram aplicações para ocupar este espaço no mercado.

Tissue Analytics e E-Kare usam as capacidades fornecidas pelo hardware do iPad, criando uma imagem 3D do ferimento. Imagem esta que é posteriormente analisada fornecendo as dimensões da ferida. A ferida pode também ser analisada recorrendo a um iPhone, sendo neste caso necessário recorrer a referência real.

As duas aplicações não estão, até ao momento, disponíveis em android e não mostram funcionalidades de tempo real, abrindo assim uma boa oportunidade.



# 3 Metodologia de Desenvolvimento

O desenvolvimento de um *software* é um processo complicado. É necessário seguir uma metodologia estruturada para que o projeto chegue a bom porto. Esta metodologia está dividida em camadas, sendo cada camada responsável por uma parte determinante na totalidade do processo. Cada desenvolvimento deve percorrer as seguintes etapas:

- Recolha de requisitos
- Arquitetura do sistema
- Implementação
- Teste
- Instalação
- Manutenção

Vários modelos foram desenvolvidos cada um deles com uma ideia distinta de como estas etapas devem interagir uma com as outras. Dois dos modelos mais usados são *Waterfall e Agile* [5].

O modelo *Waterfall* aborda todo o desenvolvimento como processo sequencial, onde só é possível passar para o nível seguinte caso o anterior esteja totalmente completo. As principais vantagens deste método são: o rigor no desenvolvimento e a facilidade de gerir todo o processo. No entanto este modelo não é facilmente adaptado a mudanças pois quebra toda a cadeia de desenvolvimento.

Por outro lado, temos o modelo Agile, que não segue uma abordagem rígida como a anterior. Este modelo prima na flexibilidade, desenvolvimento contínuo e rapidez. É um modelo que implementa capacidades individuais em vez do projeto total de uma só vez.

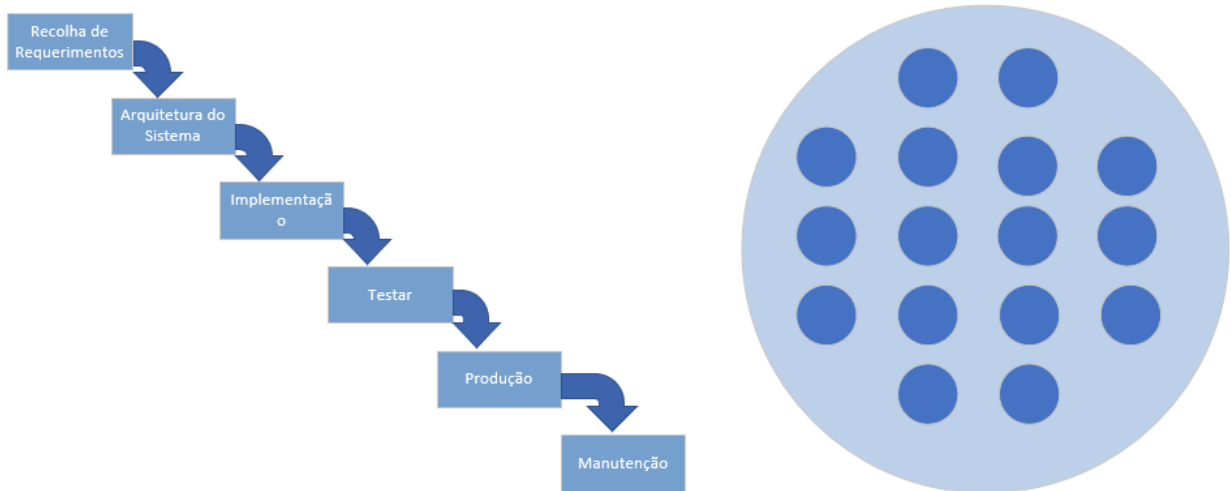


Figura 2 Waterfall vs Agile

A tese documentada neste documento trata-se de um projeto não comercial. Sendo este o caso não existe a definição concreta de cliente (alguém que paga para desenvolver o *software*). Sem cliente não existe a constante necessidade desenvolver novas capacidades para o sistema. Por este motivo foi decidido que a metodologia a usar seria *Waterfall*.

Todos os capítulos seguintes serão dedicados a uma parte do desenvolvimento.

# 4 Requisitos

Este capítulo é dedicado a 1ª fase do desenvolvimento, a recolha dos requisitos.

Requisitos são um conjunto de características que um *software* deve fornecer aos seus utilizadores. Devem ser detalhados e documentados, podendo até ser usados como base de um contrato.

Os requisitos abaixo listados foram elaborados com a ajuda do professor Rui Cortesão e do enfermeiro Luís Sousa.

<i>Número</i>	<i>Requerimento</i>	<i>Cotação</i>
1	O sistema deve permitir vários utilizadores cada um com uma conta própria protegida por palavra passe	1
2	O sistema deve permitir a introdução de um novo paciente guardando informações relevantes como o Nome, idade, altura, sexo, peso, morada, contacto, contacto de emergência, número de identificação	1
3	Cada paciente tem associado a si um historial Clínico e uma lista das respetivas feridas	1
4	O historial Clínico deve detalhar a medição atual, historial clínico, historial familiar e hábitos nocivos	1
5	Cada registo de ferida deve ser guardado com a sua respetiva localização no corpo	1
6	O sistema deve fornecer uma lista de todos os doentes no Sistema, permitindo a pesquisa por nome	1
7	Toda a informação armazenada relativa a cada paciente pode estar sujeita a mudanças	1
8	Uma ferida pode ter associada a si uma ou várias consultas	1



9	Cada consulta descreve o estado atual da ferida, listando algumas métricas como etimologia, presença de fístula, exsudado, tipo de dor, sinais de infecção e uma fotografia do ferimento.	1
10	A fotografia da ferida pode ser tirada em tempo real ou simplesmente selecionada da galeria	1
11	A fotografia pode ser analisada medindo a sua área	1
12	Cada ferida tem um historial fornecendo informação das consultas anteriores	1
13	O sistema deve ter um componente tempo real avisando os utilizadores de novas notificações	2
14	O sistema tem um sistema de mensagens em tempo real, podendo fazer videochamada	2
15	O sistema deve ter um calendário onde é possível adicionar marcações	2
16	O sistema deve permitir consultas as notificações	2

A cada requerimento está numerado e cotado. A numeração ajuda a sua identificação, enquanto a cotação revela o nível de importância (1 – Prioridade Alta, 2 – Prioridade Média, 3 Prioridade Baixa).

# 5 Arquitetura do sistema

Neste capítulo a lista anteriormente elaborada será convertida para documentos técnicos usando para isso UML (*Unified Modeling Language*)[6]. UML é uma linguagem uniformizada, desenvolvida para modelar os requisitos do software em linguagens mais técnicas, usando para esse fim uma pluralidade de métodos.

## 5.1 Casos de Uso

Em UML, os casos de uso são usados para especificar os detalhes do sistema, resumindo interações entre atores (pessoas ou objetos) e o próprio sistema.

### 5.1.1 Caso de Uso Paciente

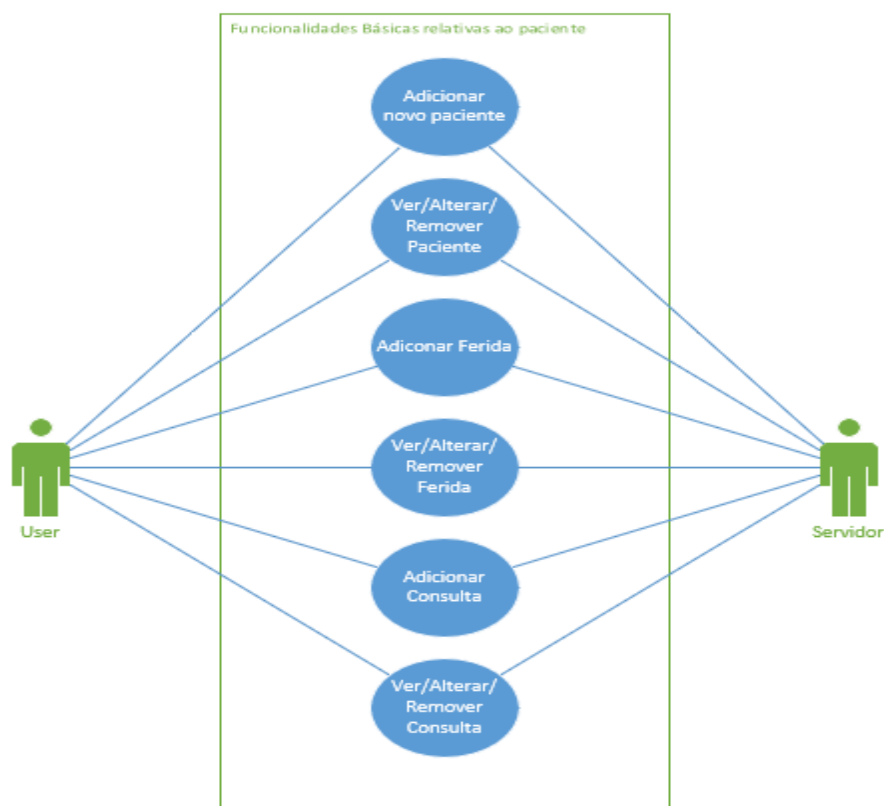


Figura 3 Caso de Uso Funcionalidades Básicas

### 5.1.2 Caso de Uso Restantes Funcionalidades

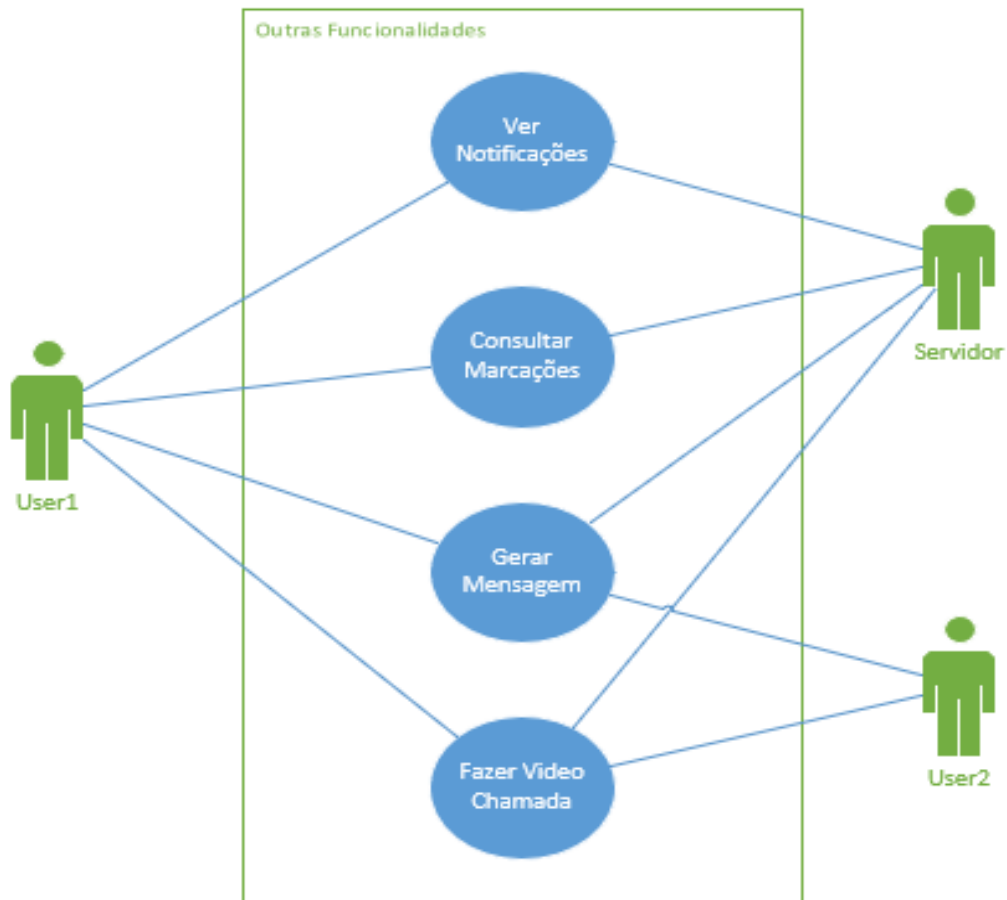


Figura 4 Caso de Uso Outras Funcionalidades

## 5.2 Diagramas de Sequência

Os casos de uso exprimem ações e seus atores, sem qualquer relação temporal nem um grande nível de detalhe. Para detalhar estas interações, atribuindo-lhes uma ordem cronológica, são usados diagramas de sequência [7].

Este diagramas temporais detalham os casos de uso, mostrando uma relação mais profunda entre o ator e os objetos que compõem o sistema.

## 5.2.1 Diagrama de Sequência Login

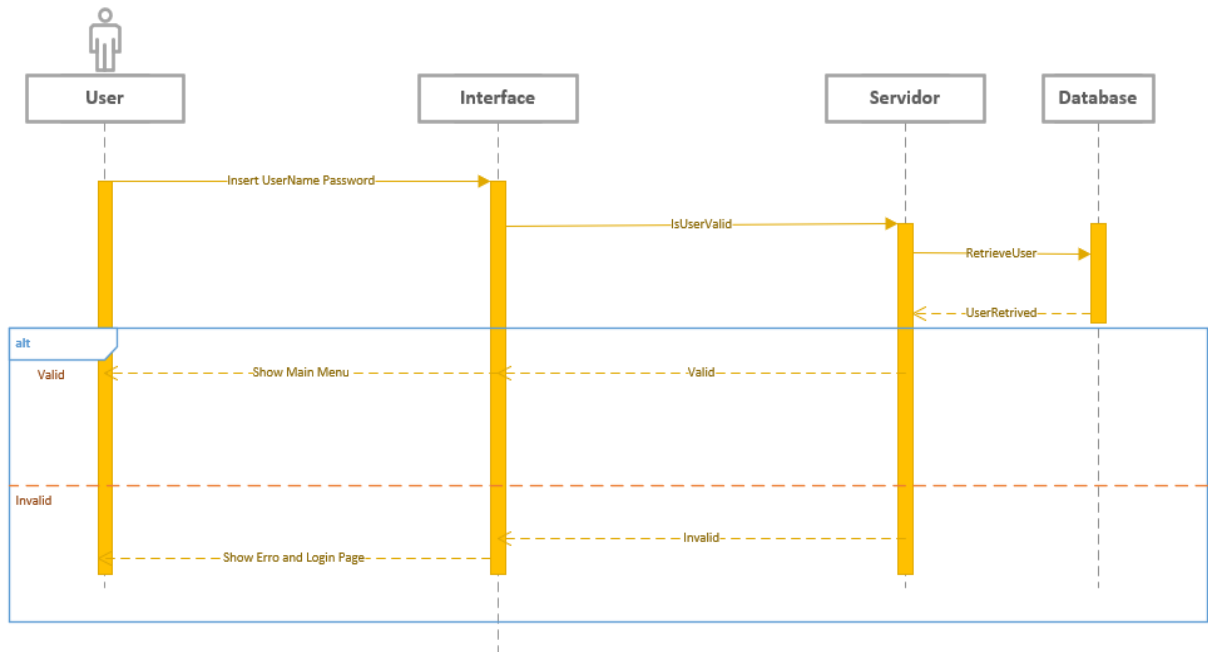


Figura 5 Diagrama de sequência login

O utilizador insere na aplicação o respetivo *Username* e *Password*. Por sua vez, a aplicação envia esta informação ao servidor que utiliza a base de dados para certificar a informação enviada. Caso a informação introduzida seja correta o utilizador será apresentado ao menu inicial caso contrário, uma mensagem de erro será gerada.

## 5.2.2 Diagrama de Sequência Adicionar Paciente

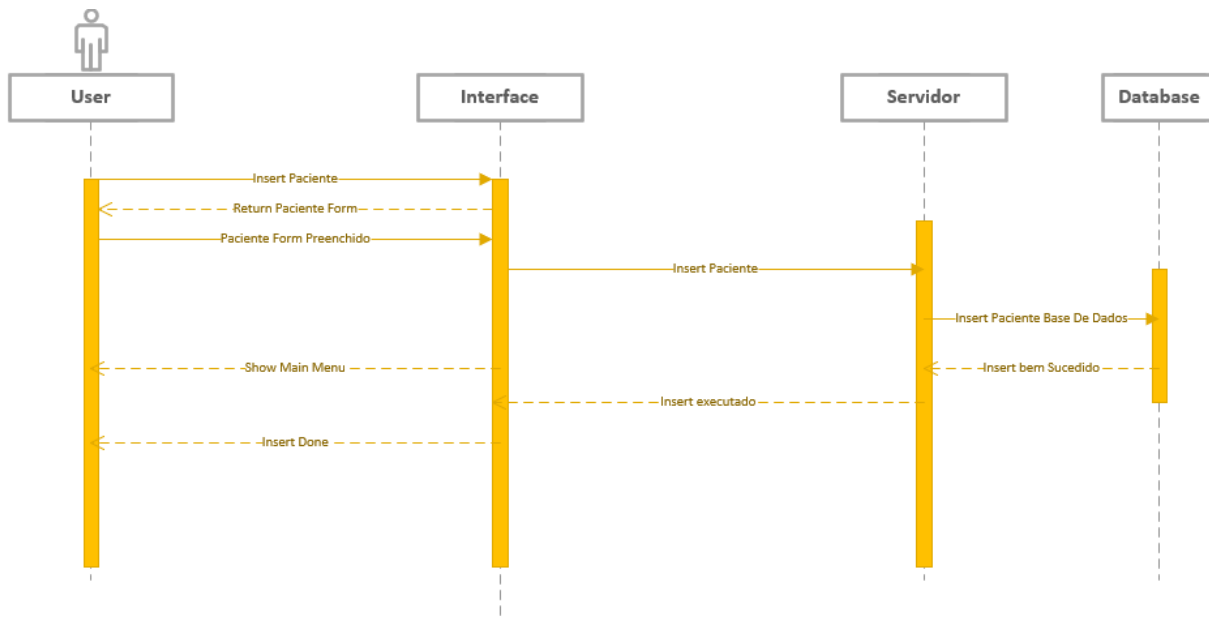


Figura 6 Diagrama de Sequência Adicionar Paciente

O utilizador selecciona a opção de adicionar um paciente. Posteriormente a aplicação apresentará um formulário ao qual o utilizador preencher. Concluindo a etapa anterior, a respetiva informação será enviada ao servidor que por sua vez a armazenará na base de dados. A conclusão deste processo, mostra uma mensagem ao utilizador avisando-o que o processo correu sem problemas.

### 5.2.3 Diagrama de Sequência Ver/Alterar/Remover Paciente

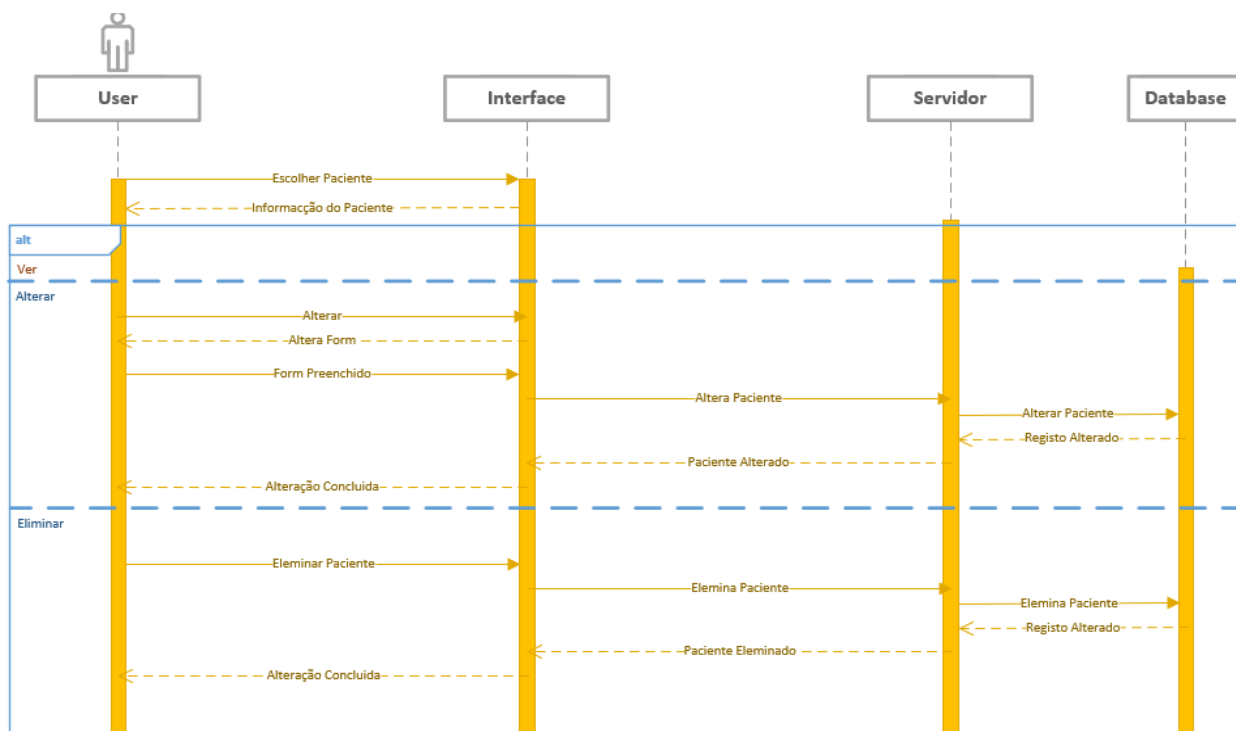


Figura 7 Diagrama de Sequência Ver/Alterar/Remover

O utilizador deve escolher um paciente. Posteriormente será apresentada toda a informação relativa a esse paciente. O utilizador pode consultar, alterar ou eliminar esta informação. Cada uma destas ações é comunicada ao servidor e posteriormente registada na base de dados. A finalização desta sequência é feita com uma mensagem ao utilizador, informando-o do sucesso ou fracasso da mesma

## 5.2.4 Diagrama de sequência Adicionar Ferida

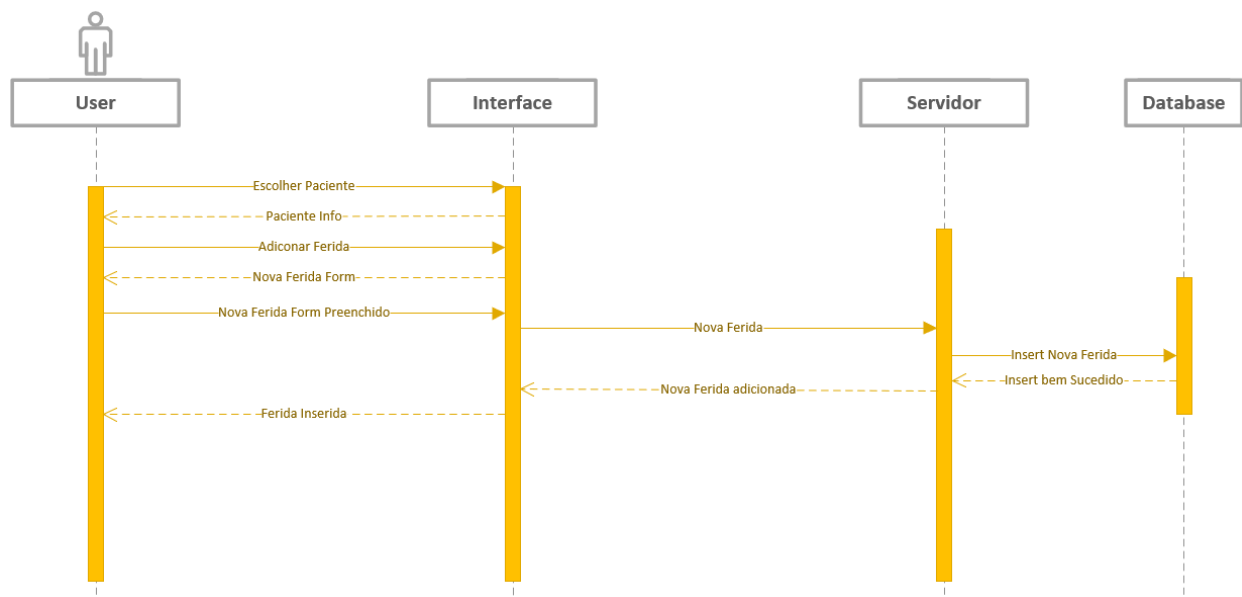


Figura 8 Diagrama de sequência Adicionar Ferida

O utilizador escolhe um paciente. De seguida é lhe mostrada toda a informação relativa ao paciente escolhido. O utilizador deve seleccionar a opção de introduzir nova ferida. A aplicação devolve o formulário de introdução de ferida, que deve ser preenchido. Depois do preenchimento do formulário, este é enviado para o servidor que armazena a sua informação na base de dados. É enviado ao utilizador uma mensagem informando-o do resultado da operação.

## 5.2.5 Diagrama de sequência Ver/Alterar/Remover Ferida

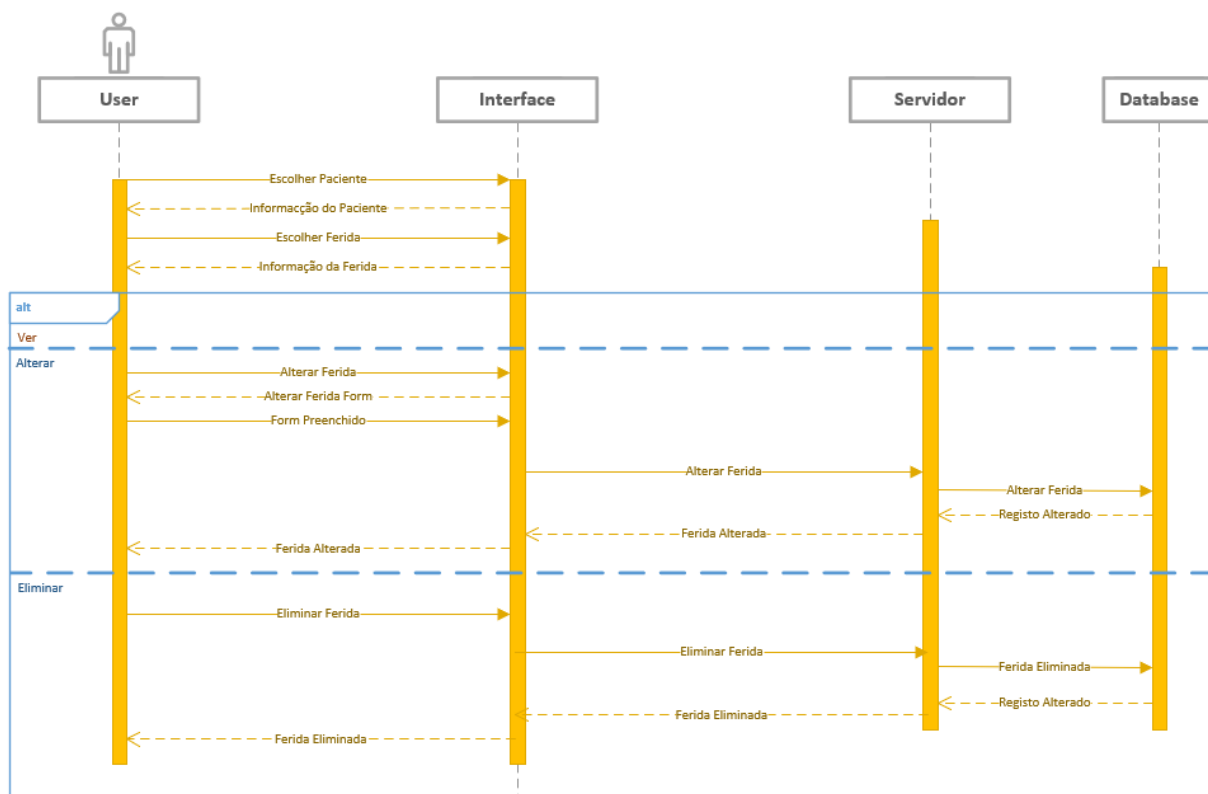


Figura 9 Diagrama de sequência Ver/Alterar/remover Ferida

O utilizador deve escolher um paciente, é lhe apresentado toda a informação relativa ao paciente incluindo as suas feridas. O utilizador deve seleccionar a ferida que deseja consultar, sendo posteriormente devolvido toda a informação relativa à ferida seleccionada. O utilizador pode optar por consultar, editar ou eliminar a informação apresentada. As opções de eliminar ou editar são transmitidas para o servidor, que por sua vez as armazena na base de dados. É enviado ao utilizador uma mensagem informando-o do resultado da operação.



## 5.2.6 Diagrama de Sequência Adicionar Consulta

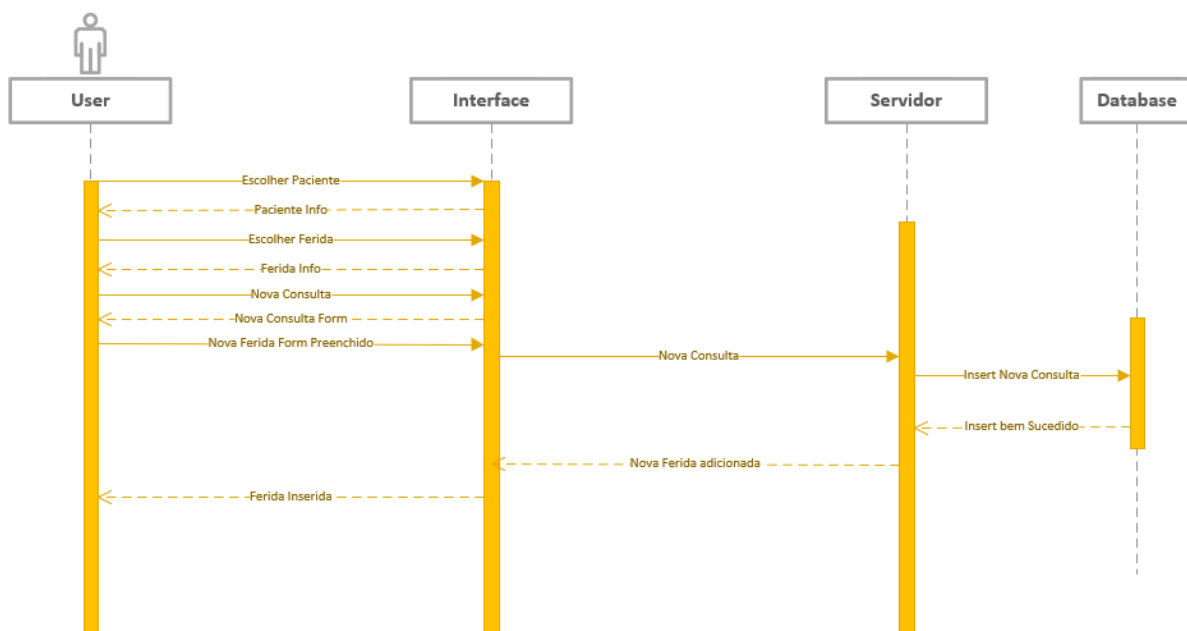


Figura 10 Diagrama de sequência Adicionar Consulta

O utilizador deve seleccionar um paciente. De seguida será devolvida toda a informação relativa ao paciente seleccionado. Na informação do paciente está presente uma lista de todas as suas feridas. O utilizador deve seleccionar a ferida que deseja. Posteriormente deve seleccionar a opção de adicionar uma consulta nova. Será apresentado um formulário para a consulta, formulário este que deve ser preenchido pelo utilizador. Depois de o formulário estar preenchido este será enviado ao servidor, que por sua vez guardará esta informação na base de dados. É enviado ao utilizador uma mensagem informando-o do resultado da operação.

## 5.2.7 Diagrama de Sequência Ver/Alterar/Remover Consulta

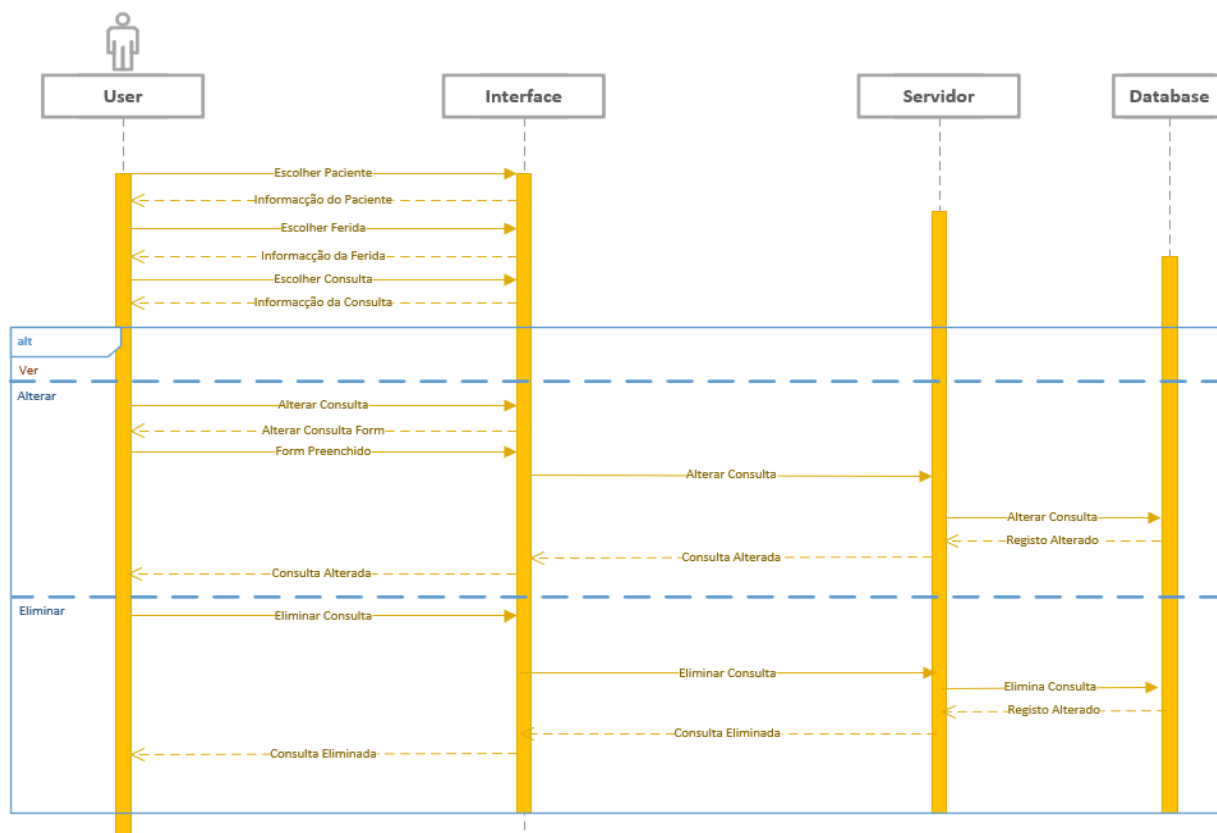


Figura 11 Diagrama de Sequência Ver/Alterar/Remover Consulta

O utilizador deve seleccionar um paciente. De seguida será devolvida toda a informação relativa ao paciente seleccionado. Na informação do paciente está presente uma lista de todas as suas feridas. O utilizador deve seleccionar a ferida que deseja. De seguida será apresentado ao utilizador toda a informação referente a ferida seleccionada. O utilizador pode consultar, editar ou eliminar esta informação. As opções de eliminar e editar são enviadas ao servidor, que as guardará na base de dados. É enviado ao utilizador uma mensagem informando-o do resultado da operação.

## 5.2.8 Diagrama de sequência das comunicações em tempo real

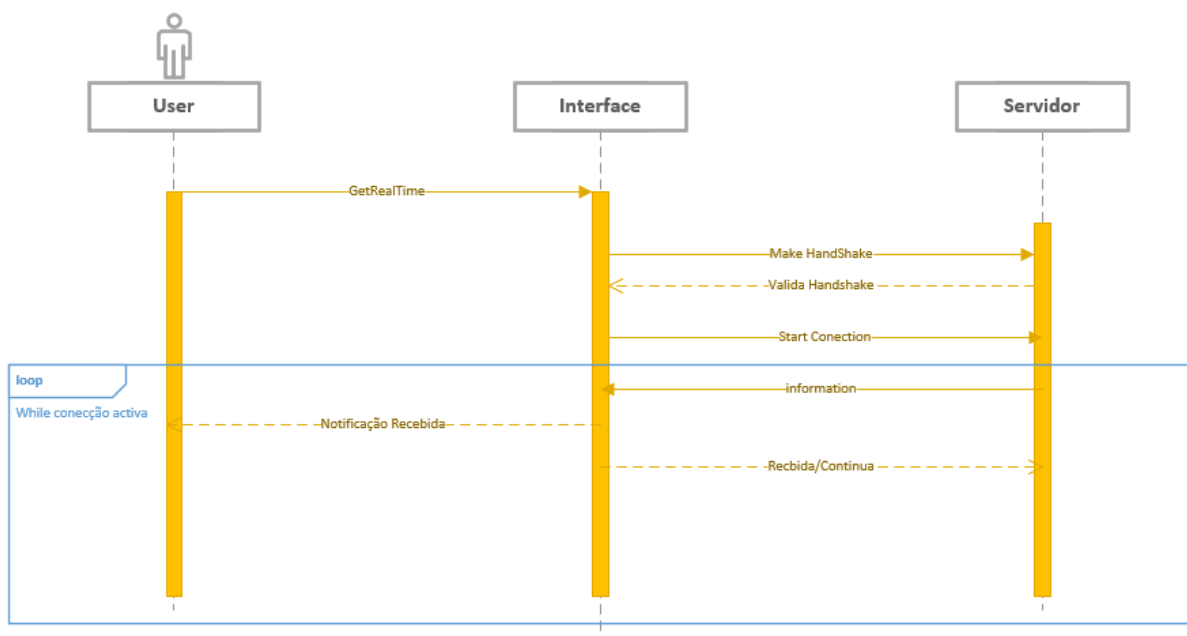


Figura 12 Diagrama de Sequência Comunicações em tempo real

## 5.3 Componentes

Os diagramas revelam que são necessários três componentes base para que o sistema funcione. Será necessário:

- Uma aplicação móvel.
- Um servidor.
- Uma base de dados.

### 5.3.1 Aplicação móvel

Aplicação móvel é a ponte entre o utilizador e o servidor. Existem múltiplos sistemas operativos como *Android*, *IOS*, *Microsoft*, etc. Consultando o gráfico em [2], é de notar que o sistema operativo Android, atualmente, ocupa grande parte do mercado (80%) seguido pelo IOS (20%), os restantes sistemas ocupam a restante parte. Como queremos que a aplicação esteja disponível ao maior número possível de utilizadores, a escolha cai em Android ou IOS.

Como não temos a disposição um Mac nem um iPhone foi decidido desenvolver aplicação usando *Android Studio* [8], a ferramenta oficial para o desenvolvimento deste tipo.

### 5.3.2 Servidor

O servidor é responsável por comunicar diretamente com a aplicação fornecendo-lhe variados tipos de informação. Existindo a possibilidade de evoluir a aplicação para uso em web, ficou decidido que o servidor seria um Web server em Windows usando para tal a ferramenta do Windows IIS (*Internet Information Services*).

O IIS é um *WebServer* desenvolvido pela Microsoft que suporta múltiplos protocolos entre os quais HTTP e WS. O IIS [9] pode ser configurado para servir vários tipos de aplicações.

Será criada uma aplicação web usando uma *framework* denominada de ASP.NET MVC [10] (Detalhes na secção seguinte), que posteriormente será instalada no IIS e servirá de servidor.

### 5.3.3 Base de Dados

Uma base de dados é uma coleção de data, que pode ser consultada eletronicamente. Esta base de dados pode ser gerida por múltiplos software. Como escolhemos um servidor Windows também iremos optar por um software desenvolvido pela Microsoft. A escolha foi SQL Server 2017.



# 6 Implementação Móvel

Este capítulo detalha o processo de desenvolvimento da aplicação móvel bem como do servidor.

## 6.1 Android Studio

Como foi mencionado em Aplicação móvel, a aplicação móvel será desenvolvida em android, usando para esse fim o Android Studio.

Uma aplicação android é dividida em vários módulos, sendo estes um conjunto de ficheiros e configurações que permitem a divisão do projeto

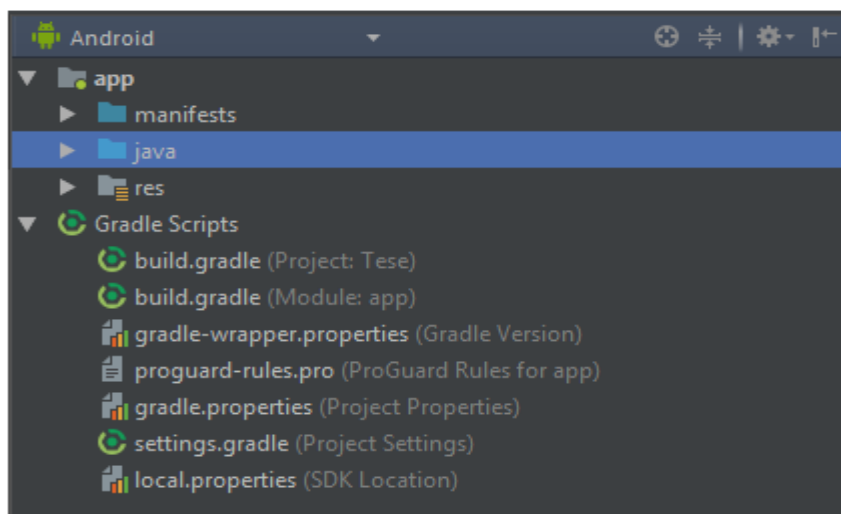


Figura 13 estrutura de pastas e ficheiros

A figura acima ilustra como o modulo está organizado. A pasta app encapsula:

- *Manifests*, onde é guardado o ficheiro de configuração, que detalha todo o hardware que a aplicação irá necessitar (Acesso à câmara, Internet, etc).
- *Java*, onde todo código e teste unitários estão guardados.
- *Res*, onde os ficheiros gráficos estão armazenados

A pasta *Gradle Scripts* engloba todas as configurações de compilação da aplicação

## 6.2 Activities

Um projeto pode ser dividido em dois tipos de classe: Classes “normais” e *Activities*. O segundo conjunto são classes nativas ao sistema que têm como principal função criar a interface com a qual o utilizador vai interagir. Estas classes possuem um ciclo de vida bem definido.

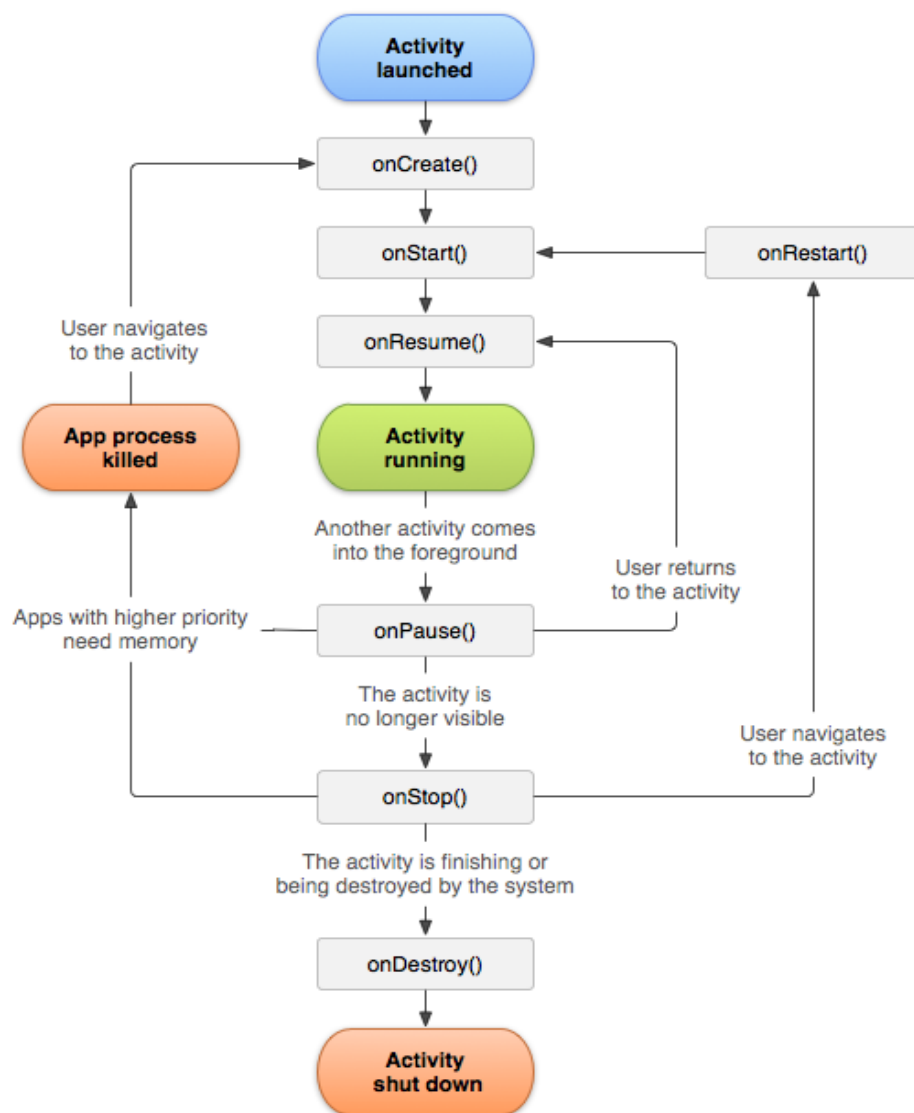


Figura 14 Ciclo de vida de uma activity [3]

## 6.3 Área da Ferida

Um dos requisitos mais complicados a nível de programação foi o cálculo da área da ferida. Este requerimento exige a deteção da área que queremos medir, mas também a sua conversão para unidades métricas.

Numa abordagem mais simples, podendo ser melhorada numa versão posterior, a deteção da área a medir será feita pelo utilizador marcando na imagem os pontos que a constituem. O cálculo da área e sua posterior conversão para unidades métricas será feita recorrendo a uma referência. Esta referência pode ser qualquer objeto de dimensões conhecidas.

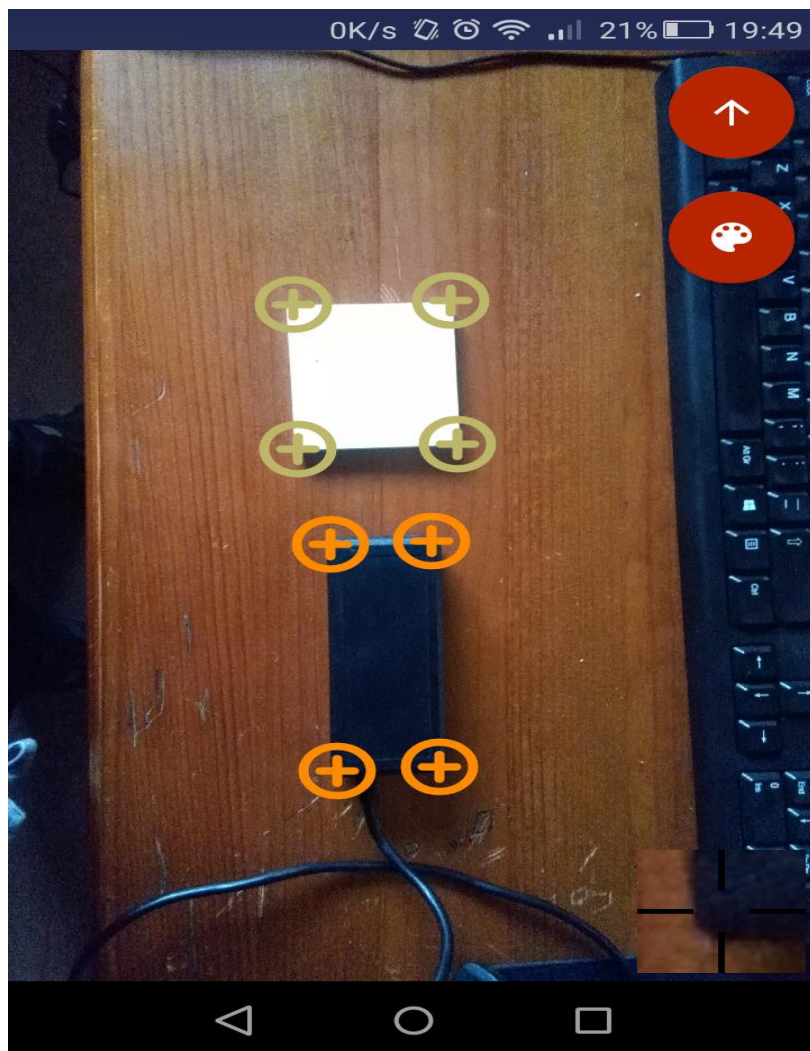


Figura 15 Cálculo da área



Assumindo que o utilizador deseja saber as dimensões do carregador, ele limita a área do carregador bem como a área da referência, introduzindo posteriormente o tamanho desta em centímetros.

## 6.4 Cálculo da área da ferida

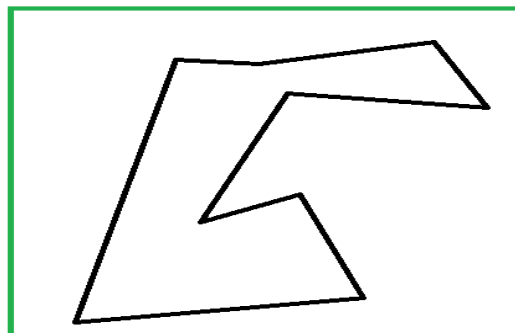
No exemplo anterior a área desejada é um retângulo, tornado o seu cálculo simples. No entanto, a área de uma ferida pode ser um polígono irregular, tornando o cálculo da sua área mais complicado.

Este cálculo pode ser abordado de várias maneiras. Neste projeto foram testados e aprofundados dois métodos:

- O método de Monte Carlo
- *Gauss's area formula*

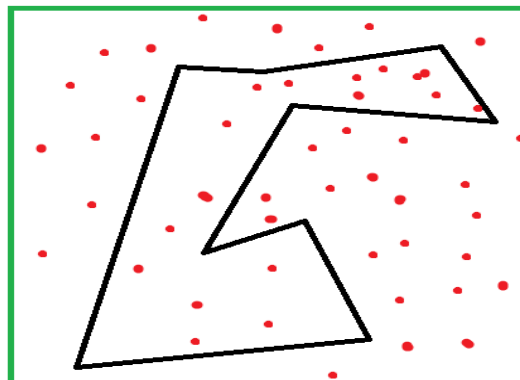
### 6.4.1 Método de Monte Carlo

Este método [4] resolve o problema das áreas recorrendo a números aleatórios. O sólido geométrico é encapsulado por um retângulo, servindo este de limite lateral e superior.



*Figura 16 Sólido dentro do retângulo limitador*

Depois de o limite estar definido, são gerados números aleatoriamente dentro do limite.



*Figura 17 Pontos Aleatórios*

Cada ponto calculado deve ser atribuído a uma categoria, dentro da figura geométrica ou fora. A área total da figura será igual à área do retângulo multiplicada pelo rácio entre o número de pontos dentro e o número total de pontos. A fiabilidade deste método vai aumento com o número de pontos utilizados. Trata-se de um método computacionalmente pesado.

#### 6.4.2 Gauss's area formula

*Gauss's area formula* é um algoritmo matemático desenvolvido para calcular a área de um polígono simples, usando para tal as coordenadas cartesianas de cada vértice. Este método pode ser descrito pela seguinte equação:

$$A = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right|$$

O sólido é decomposto em múltiplos triângulos usando como vértices dois pontos sequenciais do sólido e a origem do referencial. A área de cada um destes triângulos é  $\frac{1}{2} * \det \begin{pmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{pmatrix}$ , Logo a área total do sólido geométrico pode ser descrita como  $\frac{1}{2} * \sum_{i=1}^n \det \begin{pmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{pmatrix}$ .

Comparando os dois métodos obtemos:

Sólido	Gauss	Monte Carlo (1000)
Sólido A 4 vértices	1ms	150ms
Sólido B 8 vértices	1.5ms	250ms
Sólido C 16 vértices	1.7ms	400ms

A tabela mostra claramente que o método de Gauss é mais rápido do que o método de Monte Carlo para 1000 pontos. Um dos parâmetros não representados na tabela é precisão de cada iteração. No caso do método de Gauss a precisão depende simplesmente da referência, caso esta seja mal medida terá repercussões na área final.

Por outro lado, o método de Monte Carlo não apresenta grande precisão. Este fenómeno pode ser explicado com o tamanho dos polígonos de teste. Os polígonos usados apresentam grandes dimensões para os mil pontos usados. Aumentar a precisão corresponderia a um número de pontos maior tornando o algoritmo ainda mais lento.

## 6.5 Comunicação com o Servidor

Uma parte essencial de qualquer aplicação móvel é a interação com o servidor. Esta interação permite a aplicação mostrar ou guardar informação aos seus utilizadores. A *Internet* e os seus protocolos foram desenvolvidos com este propósito, transmitir informação rapidamente.

O projeto em causa aproveita os protocolos já desenvolvidos e documentados para poder gerir toda a informação necessária ao seu funcionamento.

### 6.5.1 HTTP

Um dos protocolos mais usados na *Internet* é o HTTP (*HyperText Transfer Protocol*). Este protocolo foi desenvolvido em 1991 com o principal objetivo de auxiliar outra ideia a *World Wide Web*. Ele viria a ajudar na transmissão de informação entre agentes, formando uma conexão cliente (aquele que pede) e servidor (aquele que fornece).

Deste de então este protocolo tem sofrido várias a iterações melhorando questões de performance bem como diminuindo as suas limitações.

Tomando partido deste protocolo foi desenvolvido um conjunto de métodos que usam HTTP juntamente com JSON para transmitir informação de e para o servidor.

JSON (*JavaScript Object Notation*) é um dos formatos mais usados para transmitir informação. É incorporado no corpo do pedido HTTP e é facilmente interpretado tanto pelo servidor como pelo cliente, aumenta a eficácia e facilidade de comunicação.

```
POST /api/Communication/Login HTTP/1.1
Content-Type: application/json; charset=utf-8
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; HUAWEI SCL-L21 Build/HuaweiSCL-L21)
Host: 2.82.7.186
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 41
{"Password":"Teste","UserName":"Ricardo"}
```

O exemplo acima mostra um pedido HTTP *Post* ao servidor. A cinzento o pedido que queremos fazer e o protocolo que vamos usar. A verde estão os *headers* responsáveis por transportar informação sobre o pedido e finalmente a amarelo temos uma *String* JSON, que neste caso transporta informação sobre o *Login* do utilizador.

## 6.5.2 Web Sockets

A *Internet* é um sistema global de computadores interligados que usam um conjunto de protocolos para partilhar informação entre si. Nos seus primórdios, a partilha de informação era algo “simples”, havia uma agente responsável por criar um pedido e algures havia um agente responsável por responder a este. No entanto, com o evoluir dos tempos estas noções deixaram de fazer sentido em alguns casos. Casos esses em que é necessária uma porta constantemente aberta ente cliente e servidor para que estes possam comunicar instantaneamente.

Surgem então as *WebSockets*. *WebSockets* são uma forma de comunicação bidirecional. O cliente inicia um acordo com o servidor através de um pedido HTTP. Se o servidor aceitar a ligação é evoluída para *WebSockets* permitindo, até um utilizador quebrar a ligação, comunicações em tempo real.

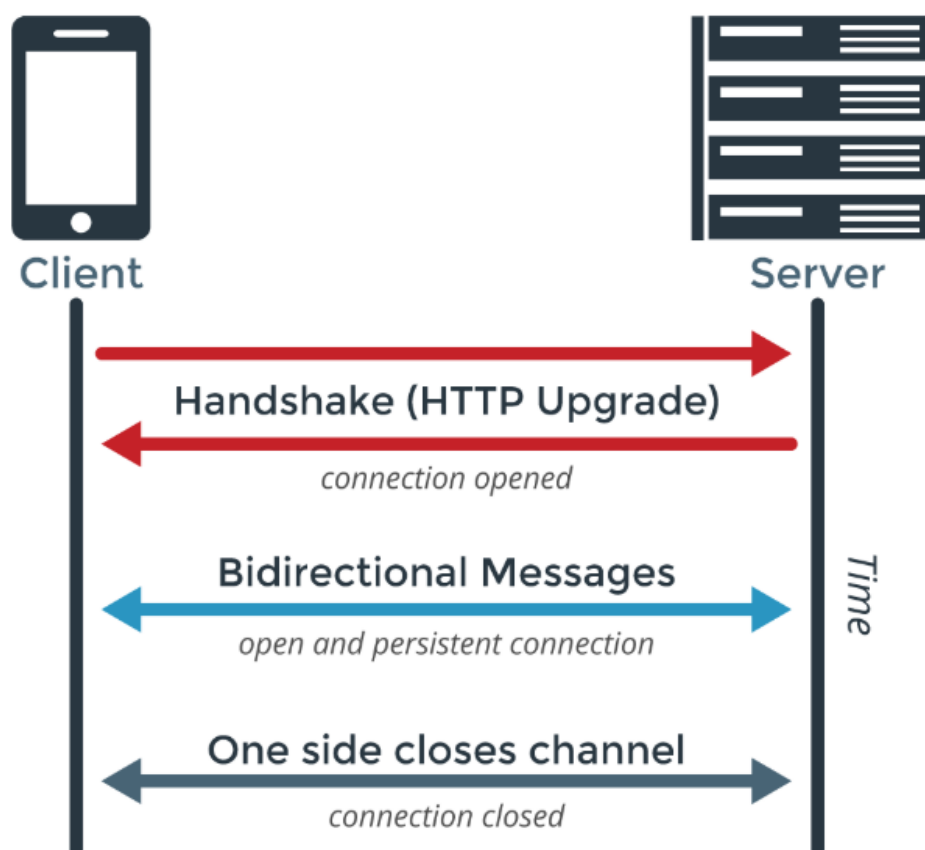


Figura 18 Diagrama WebSockets



# 7 Implementação Servidor e Base de Dados

A aplicação móvel necessita de um servidor que responda aos seus pedidos. Esta secção descreve o processo que foi percorrido para desenvolver um servidor funcional.

## 7.1 Servidor

No ponto Servidor ficou definido que o servidor iria correr em Windows através do IIS usando uma aplicação web criada usando o *framework* ASP.NET MVC. Esta *framework* têm três principais componentes

- *Controller* - responsável por responder aos pedidos feitos a aplicação web.
- *Model* - representa toda a informação do sistema (Classes, Interfaces).
- *View* - este componente não é usado nesta tese. Será necessário para desenvolver uma aplicação web

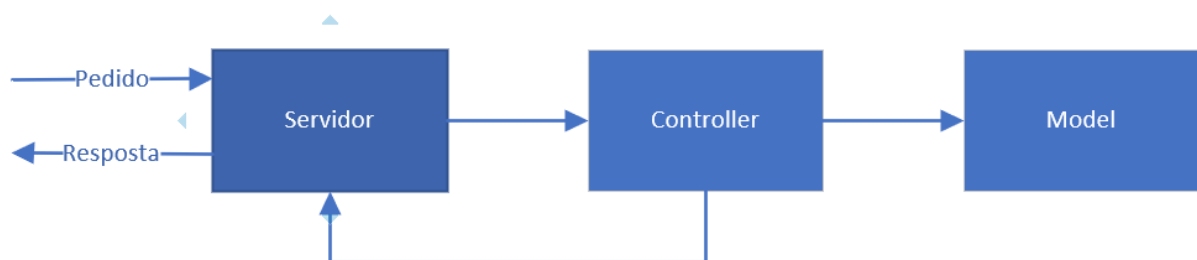


Figura 19 Relação entre Servidor, Controller e Model

### 7.1.1 Signalr

ASP.NET Signalr é uma biblioteca que torna possível o desenvolvimento de funcionalidades *real-time*. Esta biblioteca permite ao servidor enviar informação ao cliente, assim que ela é gerada.

Signalr suporta várias técnicas para lidar com as comunicações em tempo real:

- WebSockets
- Server-Sent Events
- Long Polling

Para a aplicação desenvolvida, o sistema foi configurado para usar *WebSockets*. A motivação por detrás desta escolha está relacionada com a maior facilidade em integrar este método com o sistema *Android*.

Esta biblioteca recorre ao uso de *hubs*, que atuam como controladores de comunicação.

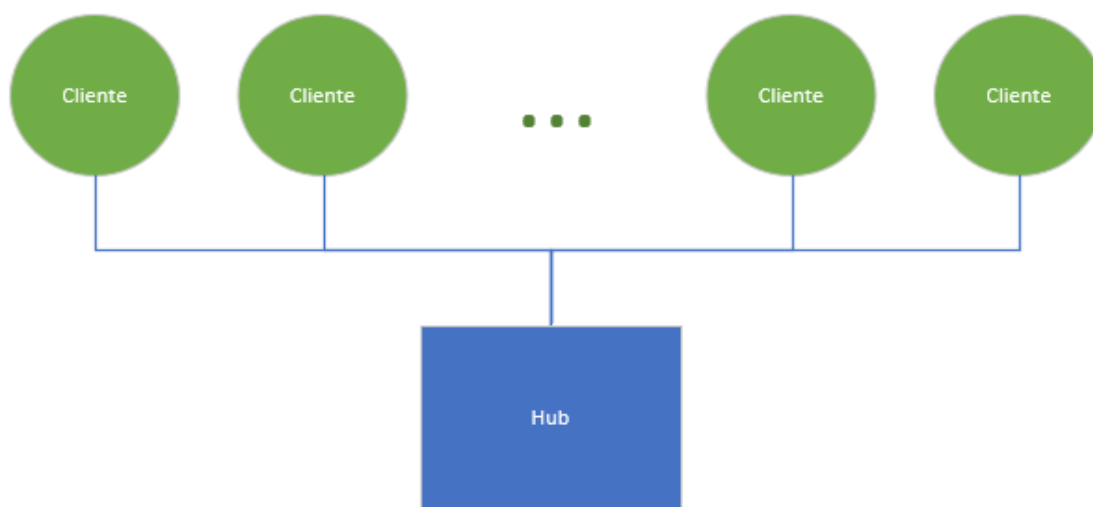


Figura 20 Signalr esquema

O *hub* é uma classe do sistema que permite aos seus clientes invocarem os seus métodos. Estes métodos são codificáveis permitindo uma ampla gama de possibilidades.

### 7.1.2 Vídeo Chamada

A telemedicina está a tornar-se uma grande área no campo da saúde. Ela possibilita a fácil comunicação entre paciente e profissional de saúde com uma enorme facilidade. Neste trabalho foi tentado implementar esta funcionalidade. A funcionalidade não ficou a funcionar, mas deixarei aqui o que foi feito e o que possivelmente possa ser melhorado.

WebRTC é um projeto *open source* que possibilita a comunicação em tempo real usando simples APIs. Para fazer *streaming* de data são precisos três pontos fulcrais:

- *MediaStream* – Responsável por armazenar a informação de vídeo e se for o caso sonora.
- *RTCPeerConnection* - API que transmite a informação. Processo mais complicado que envolve entre outras coisas *signal processing*, *codec handling*, *peer to peer communication* ...
- *RTCDataChannel* – transmite outro tipo de informação, usando um mecanismo semelhante as *WebSockets*.

Para tentar recriar este mecanismo foi desenvolvido um simples servidor Node.js. Este servidor corria uma aplicação que permitia dois utilizadores fazerem entre si uma vídeo chamada.

Usando a mesma teoria, tentou-se replicar o mesmo comportamento em Android. No entanto a transmissão apresentava perdas e não funcionava em grande parte das versões. Por este motivo foi decidido que não seria incluída nativamente. Uma solução, pouco elegante, para este problema passa por mostrar na aplicação uma vista direta do browser.

### 7.1.3 Base de dados

Para guardar os dados referentes a cada paciente e utilizadores, foi necessário desenvolver uma base de dados relacional.

Para criar a base de dados foi usado *SQL Server* com o auxílio do *SQL Management Studio(SSMS)*. O SSMS é um ambiente de desenvolvimento que permite configurar, monitorizar e gerir a base de dados e os seus respetivos componentes.



## 7.1.4 Esquema da Base de dados

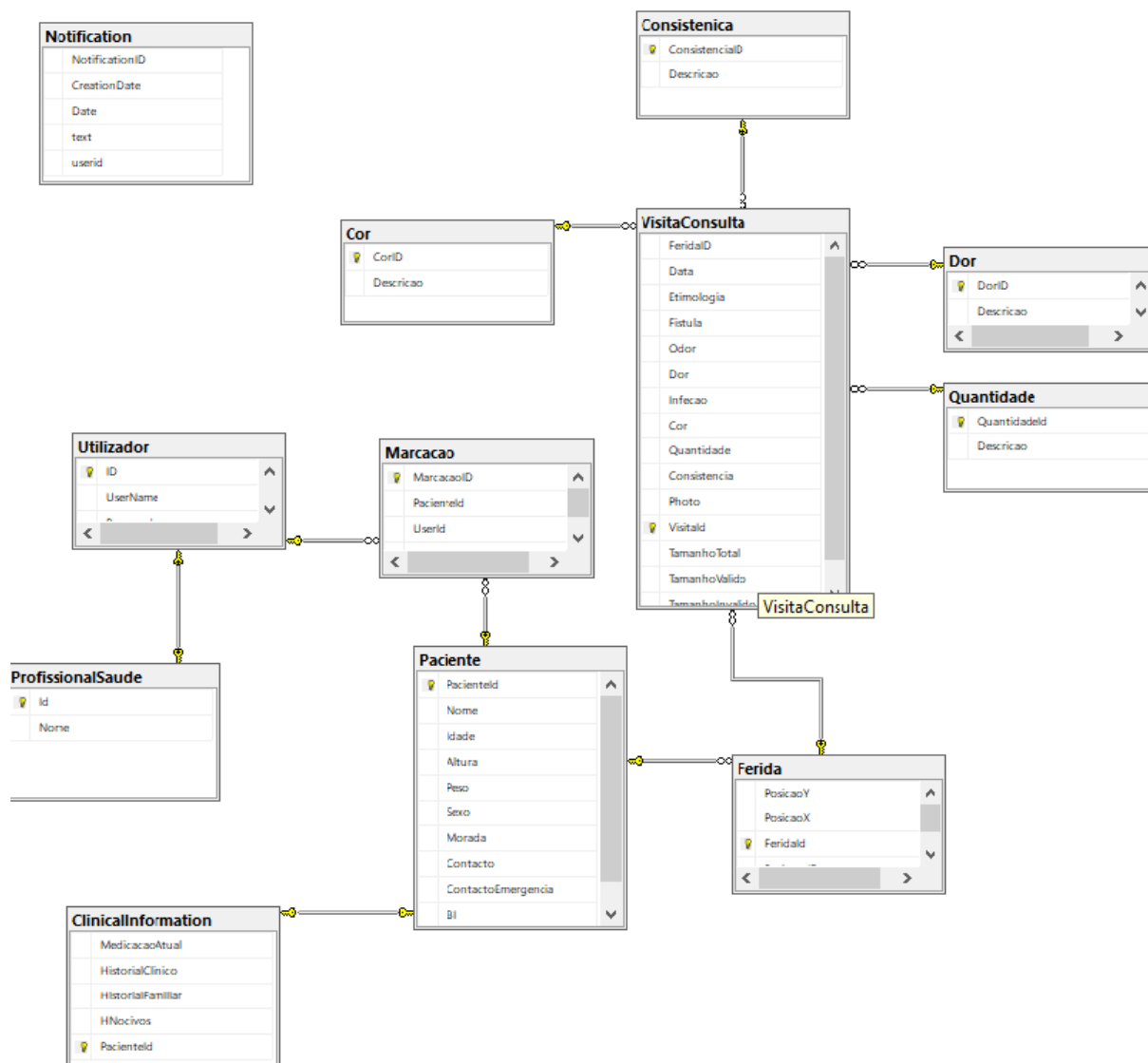


Figura 21 Esquema da Base de dados

Numa fase inicial, podendo ser alterado posteriormente, todos os utilizadores são profissionais de saúde.

Um paciente pode ter múltiplas feridas e cada uma respetivamente têm associada a si um conjunto de consultas.

Cada paciente pode ter várias marcações sendo cada uma associada a um utilizador específico.

# 8 Teste e Manual de Utilizador

Este capítulo é o final na fase de desenvolvimento do software. É composto por duas partes:

- Testar – Utilizar o software num cenário real, tentando percorrer todos os cenários, documentar os resultados e possíveis erros ou bugs.
- Produção – Depois de passar todos os testes, é necessário disponibilizar a aplicação ao público.

## 8.1 Testes

O desenvolvimento de teste para testar a aplicação é uma parte fundamental no processo de desenvolvimento. Sem ela existe a possibilidade de erros no software seguirem para produção, trazendo possíveis riscos para os seus utilizadores bem como a empresa responsável.

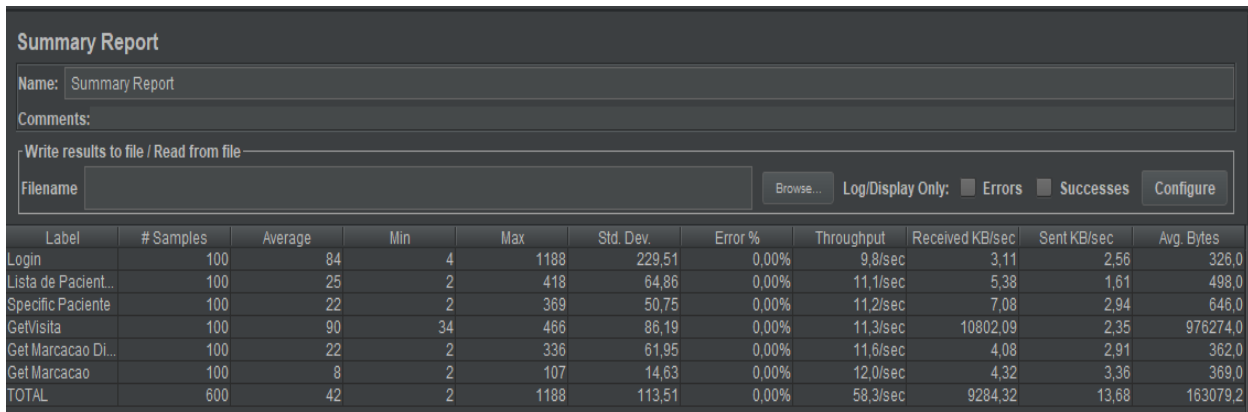
### 8.1.1 Testes unitários

Um teste unitário é um teste a parte indivisível de um programa, a função ou método.

Estes testes são levados a cabo durante a fase de desenvolvimento. Uma classe de testes com vários métodos é criada, cada um responsável por testar um método específico. São uma ferramenta importante para ajudar os programadores a descobrir erros ou bugs.

## 8.1.2 Teste de Carga

O outro tipo de teste que corremos denomina-se de teste de carga. Este teste consiste em testar as capacidades do sistema em responder a múltiplos pedidos.



The screenshot shows the 'Summary Report' window in JMeter. It includes fields for 'Name' (Summary Report), 'Comments', and a section for 'Write results to file / Read from file' with a 'Filename' input and a 'Browse...' button. There are also checkboxes for 'Log/Display Only: Errors' and 'Successes', and a 'Configure' button. Below this is a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	100	84	4	1188	229,51	0,00%	9,8/sec	3,11	2,56	326,0
Lista de Pacient...	100	25	2	418	64,86	0,00%	11,1/sec	5,38	1,61	498,0
Specific Paciente	100	22	2	369	50,75	0,00%	11,2/sec	7,08	2,94	646,0
GetVisita	100	90	34	466	86,19	0,00%	11,3/sec	10802,09	2,35	976274,0
Get Marcacao Di...	100	22	2	336	61,95	0,00%	11,6/sec	4,08	2,91	362,0
Get Marcacao	100	8	2	107	14,63	0,00%	12,0/sec	4,32	3,36	369,0
TOTAL	600	42	2	1188	113,51	0,00%	58,3/sec	9284,32	13,68	163079,2

Figura 22 Jmeter

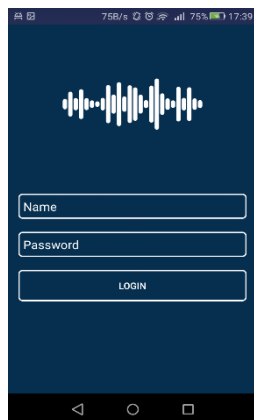
A figura acima é uma representação do relatório de carga do software JMeter. Este software permite entre outras coisas testar os pedidos e os seus tempos de resposta. Observando a figura acima podemos ver que para 100 pedidos todos eles responderam abaixo dos 100 ms. Nesta tabela so estão representados os pedidos que possuem uma grande taxa de uso. Pedidos como adicionar, alterar ou remover não foram testados pois estaticamente é pouco provável que aconteçam simultaneamente.

## 8.1.3 Teste na ótica do utilizador

O último teste realizado foi usando a perspetiva do utilizador. Para este teste foi pedido a um conjunto de pessoas que testassem o software livremente. Posteriormente foi lhes pedido que reportassem algum erro ou bug encontrado.

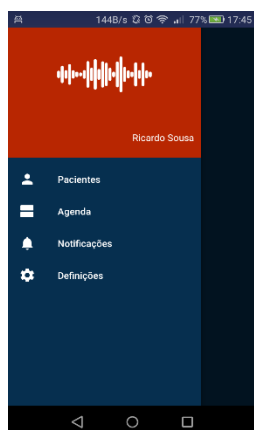
## 8.2 Manual de Utilizador

### 8.2.1 Login



Login – Como se trata de uma aplicação com informação sensível login tem de ser autenticado.

### 8.2.2 Menu Principal



Menu Principal – Aqui é possível aceder a todas as funcionalidades do sistema. Opção paciente onde é possível consultar toda a informação referente a todos os pacientes no sistema.

Agenda onde é possível consultar todos os compromissos.

Notificações onde podemos ver todas notificações que o sistema gera ao longo da sua execução.

Definições onde é possível alterar as definições de utilizador.

### 8.2.3 Lista de Pacientes



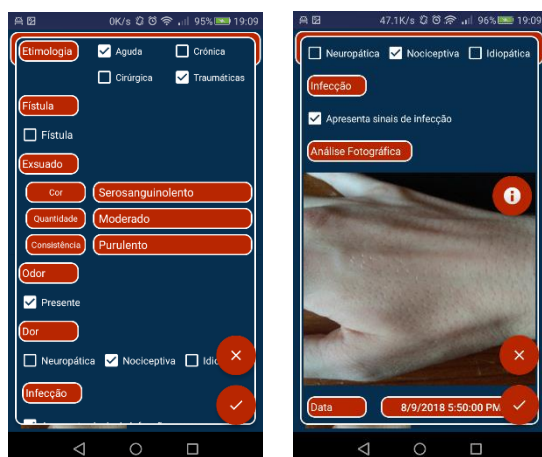
Lista de pacientes. Cada linha representa uma paciente diferente mostrando o nome, o número do B.I., sexo e o número de feridas. A lista pode ser filtrada pelo nome.

## 8.2.4 Adicionar Paciente



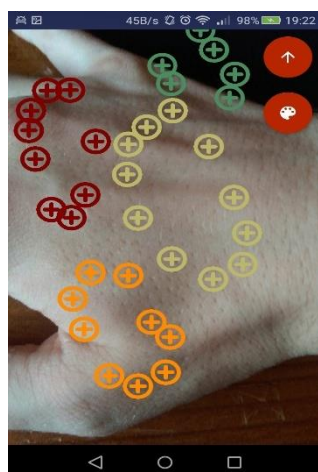
Para adicionar um utilizador é preciso preencher os três campos. Clicando em cada um dos botões um formulário ira aparecer.

## 8.2.5 Adicionar Consulta



Para adicionar uma consulta nova é necessário preencher todos os campos deste formulário fornecendo uma fotografia do ferimento direta ou *upload*.

## 8.2.6 Calcular Área da Ferida



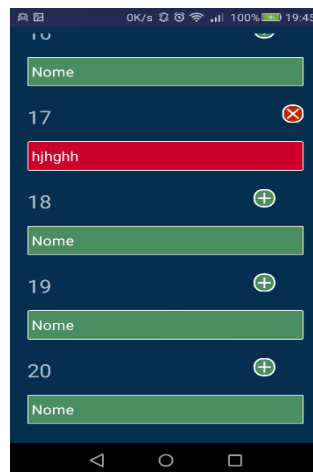
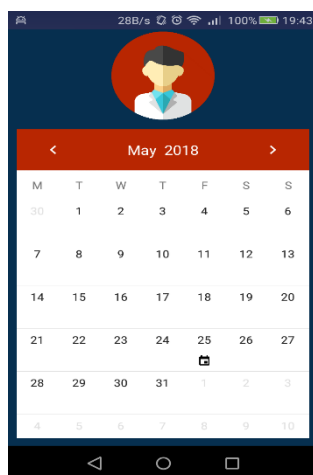
Selecionando o botão em cima da imagem em Adicionar Consulta este *activity* será chamada. Nela o utilizador recorrendo a um conjunto de pontos pode selecionar a área que deseja medir. Os pontos devem ser colocados por ordem de maneira ao método de calculo resultar.

## 8.2.7 Notificações



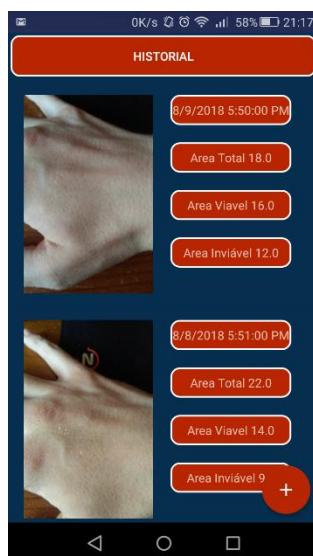
Caso um utilizador esteja ligado a aplicação qualquer tipo de ação gera uma notificação em tempo real que aparece na barra superior do ecrã. No caso de o utilizador não estar ligado ao sistema, será possível consultar a lista completa posteriormente.

## 8.2.8 Marcações



Um utilizador pode consultar com ajuda de um calendário os dias em que tem consultas. Clicando no dia aparece uma lista das 24 horas indicando quais as que estão disponíveis (verde) e as indisponíveis (a vermelho)

## 8.2.9 Historial de Feridas



Neste menu é possível consultar todo o historial de uma ferida, mostrando o dia em foi introduzida no sistema, bem como a sua área total.



## 9 Conclusão

O principal objetivo desta tese era desenvolver um sistema que cumprisse todos os requisitos abordados em [4]. Infelizmente, não foi possível tornar a comunicação em vídeo uma realidade totalmente funcional. No entanto todos os outros requisitos foram cumpridos.

A aplicação é funcional e esta pronta a ser utilizada, necessitando apenas de um servidor global.

No entanto, como qualquer outro projeto, pode ser melhorado. A funcionalidade de comunicação audiovisual seria uma mais valia para a aplicação. A análise da ferida pode ser evoluída para um novo patamar, fornecendo informação ao utilizador sobre o tipo tecido.

Uma aplicação totalmente dedicada ao paciente também seria uma mais valia.



# Referências

- [1] U. N. L. o. Medicine, “Human Skin Wounds: A Major and Snowballing Threat to Public Health and the Economy,” [Online].
- [2] statista, “Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018,” [Online]. Available: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.
- [3] Google, “Activity,” [Online]. Available: <https://developer.android.com/reference/android/app/Activity>.
- [4] “Findind Areas Using the Monte Carlo Method,” [Online]. Available: [http://mathonweb.com/entrtrain/monte/t\\_monte.htm](http://mathonweb.com/entrtrain/monte/t_monte.htm).
- [5] M. Lotz, “Waterfall vs. Agile: Which is the Right Development Methodology for Your Project?,” [Online]. Available: <https://www.seguetech.com/waterfall-vs-agile-methodology/>.
- [6] “<http://www.uml.org/>,” [Online]. Available: <http://www.uml.org/>.
- [7] “UML Sequence Diagram Tutorial,” [Online]. Available: <https://www.lucidchart.com/pages/uml-sequence-diagram>.
- [8] “<https://developer.android.com/studio/>,” [Online].
- [9] “IIS,” [Online]. Available: <https://www.iis.net/>.
- [10] “Learn About ASP.NET MVC,” [Online]. Available: <https://www.asp.net/mvc>.
- [11] “<https://pointclickcare.com/>,” [Online].
- [12] “<https://www.tissue-analytics.com/#page=home>,” [Online].
- [13] “<https://ekare.ai/>,” [Online].