TelemOLD

Manual Técnico

Índice

1 Introdução 4
1.1 Objetivos 4
1.2 Arquitetura 4
1.3 Principais Entidades 6
1.3.1 Doente 6
1.3.2 Equipamento 6
1.3.3 Parâmetro 7
1.3.4 Evento 7
1.3.5 Alerta 7
1.3.6 Mensagem 7
2 Aplicação Móvel 8
2.1 Descrição 22
2.2 Arquitetura 22
2.3 Interface 22
2.4 Serviço 22
2.5 Motor de Aquisição 23
2.5.1 Base 23
2.5.2 Dispositivos 23
2.6 Base de dados 23
2.7 Comunicação 23
3 Aplicação Web 24
3.1 Descrição 24
3.2 Arquitectura 24
3.3 Mapa da Aplicação 26
3.4 Componentes/Tecnologias 25
3.5 Gráfico ..... **Erro! Marcador não definido.**
3.5.1 Introdução 27
3.5.2 Entidades e Relações 29
3.5.3 Funções 29
3.5.4 Configurações 33
4 Serviço de Comunicação 34
4.1 Descrição 34

4.2 Arquitetura 34
4.3 Protocolo de Comunicação 34
4.3.1 Operações 34
4.3.2 Sintaxe 35
4.4 Configurações 37
6 Serviço de SMS 38
6.1 Descrição 38
6.2 Verificação de Alertas por Enviar 38
6.3 Processamento dos Alertas 38
6.4 Envio dos Alertas 38
6.5 Instalação 39
7 Trabalho Futuro 40
7.1 Sistema de Classificação 40
7.2 Aplicação Web 40
7.3 Serviços de SMS 40
7.4 Aplicação Móvel 40
8 Anexos 41

1 Introdução

1.1 Objetivos

O projeto TelemOLD visa a implementação um sistema de monitorização remota de doentes submetidos a Oxigenoterapia de Longa Duração (OLD).

Estes doentes encontram-se ligados a sensores de monitorização da saturação de oxigénio no sangue (SPO_2), da frequência cardíaca (FC) e a um acelerómetro que permite determinar o nível de esforço do doente através de Equivalentes Metabólicos de Tarefas (*Metabolic Equivalent of Tasks* i.e. METS) e da contagem do número de eventos por segundo (COUNTs).

Os sensores, oxímetro e acelerómetro, são colocados no doente e enviam, através de *bluetooth*, os sinais recolhidos para um telemóvel associado ao doente, que processa os dados recebidos e envia, através de 3G ou GPRS, para um servidor que os coloca numa base de dados.

A existência de um sistema web de monitorização dos dados recolhidos permite visualizar, através de um browser, toda a informação associada ao doente (dados administrativos; dados clínicos, diagnósticos, prescrições de oxigénio e ventilação e os dados monitorizados).

Para além da visualização através da web dos dados recolhidos é, também possível, implementar um sistema de alertas que, com a análise dos dados recolhidos, pode assinalar ocorrências e enviar SMS de aviso aos clínicos de situações anormais (ex.: valores excessivos ou diminutos, etc.).

1.2 Arquitetura

A arquitetura do projeto TelemOLD resume de uma forma simples aos pontos enunciados abaixo:

1. Utilização de sensores de monitorização (oxímetros, acelerómetros) colocados no doente, que enviam sinais, por *bluetooth*, para um telemóvel.

2. O software instalado no telemóvel recebe os sinais dos sensores (descritos em 1), faz um processamento local, regista os sinais processados numa base de dados local e envia periodicamente, por GPRS/3G, esses valores para um servidor que se encontra à escuta de mensagens provenientes dos seus clientes (telemóveis). A comunicação realizada entre cliente e servidor segue um protocolo estipulado (ver capítulo 5.3) que permite a identificação do tipo de mensagem (identificação do cliente, comunicação periódica de sinais, logs de dispositivos e eventos, alertas e feedback das comunicações) que o cliente pretende transmitir e os repectivos dados. É uma aplicação cliente-servidor mas em que o cliente não necessita que o servidor esteja online, uma vez que se tal não acontecer o telemovel continua o seu processamento de sinais e o seu registo na base de dados local. Desta forma, o processamento de sinais é independente da comunicação.

3. O servidor anteriormente referido (registado e com acesso através da Internet) existe em execução através de um serviço Windows que escuta um porto IP e recebe os valores e eventos enviados pelo telemóvel segundo o protocolo de comunicação anteriormente referido. Este

serviço regista numa base de dados os valores recebidos e permite em tempo real visualizar quais os pedidos que estão a ser recebidos e o que está a ser processado.

4. Dedicada ao projeto existe uma base de dados (atualmente SQL Server) pronta para receber os dados recebidos pelo serviço Windows referido no ponto anterior e que armazena, para além destes valores, informação sobre os doentes, equipamentos, mensagens a enviar ao doente (é possível escrever uma sequência de caracteres que ficará registada na BD e será enviada, por um protocolo proprietário de comunicação entre o serviço Windows anteriormente descrito e o telemóvel, e apresentada no ecrã do telemóvel do doente), eventos ocorridos, alarmes (explicados de seguida) ocorridos, etc.

5. Caso os valores recebidos na base de dados acionem um alarme (um alarme é definido através da indicação do parâmetro a ser monitorizado, do valor a comparar, do tipo de comparação a efetuar (=, <, <=, >, >=) e duração que essa condição se verifique, ex.: "se *frequência cardíaca* for *menor* que *50* durante *120 segundos*" então é para despoletar um alarme) será enviado, por SMS, para um telemóvel, pré-estabelecido, uma mensagem de alerta (que se encontra definida e associada à caracterização do alarme);

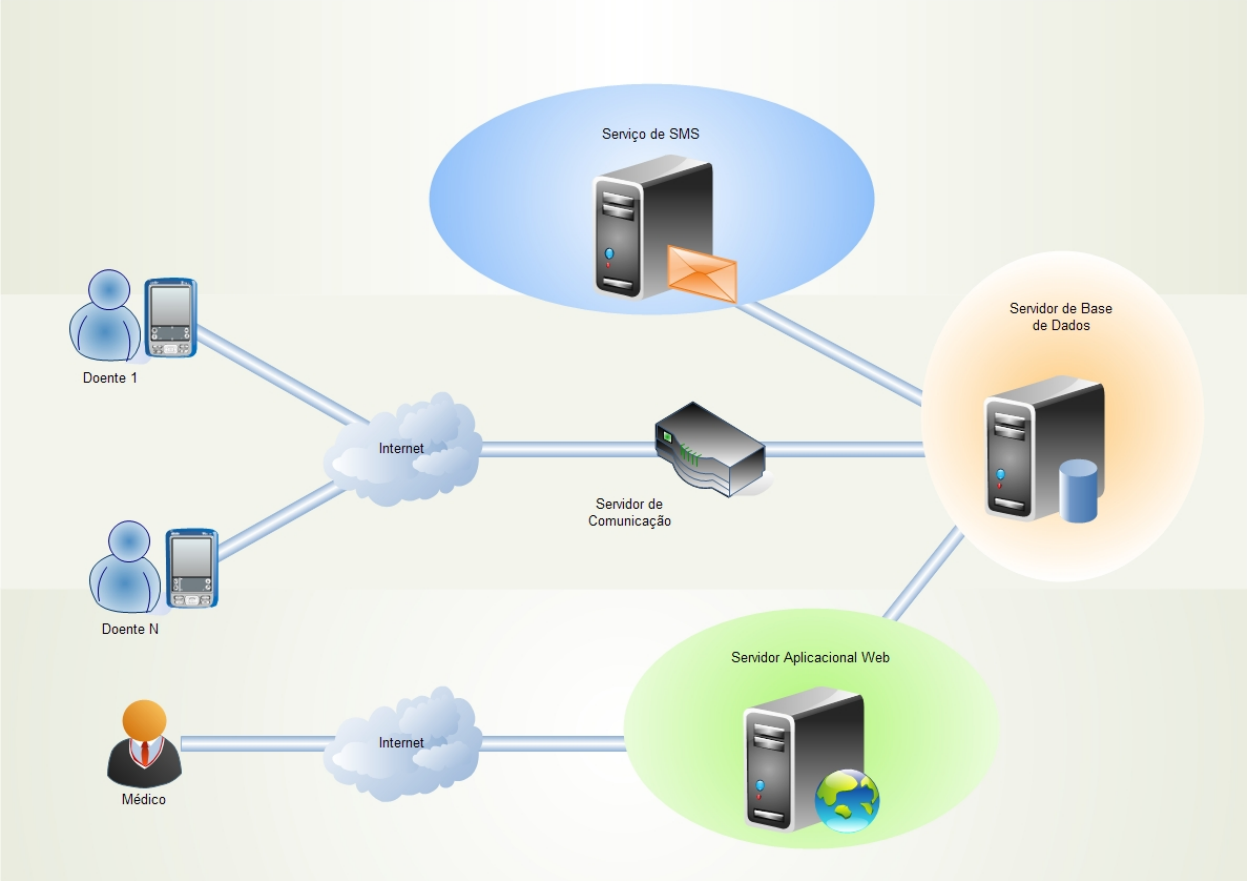
6. Sobre a base de dados existe um site web que permite a:

    a) Introdução/visualização da informação (administrativa e clínica) associada a cada doente;

    b) Visualização gráfica (em tempo real ou à posteriori) da evolução dos diversos parâmetros medidos a cada doente;
    c) Realizar pesquisas globais ou individualizadas dos dados monitorizados, eventos e/ou alarmes;

    d) Configuração do sistema (utilizadores, equipamentos, parâmetros, tabelas fixas, etc.);

Apesar do sistema implementado se debruçar sobre doentes em tratamento de Oxigenoterapia de Longa Duração (OLD), ao longo da sua evolução futura pretende-se que este seja sempre o mais genérico possível e que permita, no futuro, a aquisição de diversos sinais biológicos, de uma forma modular. Esta capacidade de configuração/adaptação do sistema esteve patente durante todos os processos de desenvolvimento dos diversos componentes.



1.3 Principais Entidades

De acordo com os objetivos estabelecidos para o sistema e os diferentes componentes descritos na arquitetura geral, o sistema está construído em função de seis principais entidades.

1.3.1 Doente

Os doentes representam a principal fonte de aquisição dos indicadores que posteriormente poderão ser alvo de monitorização médica. Através da sua informação clínica salvaguardada nas características desta entidade, é possível aceder a uma panóplia de dados que permitem uma visão detalhada de cada doente. Com os seus dados pessoais, informação clínica referente aos médicos que seguem o doente (médico assistente, médico pneumologista, médico de família), diagnósticos e prescrições como principais propriedades do Doente, traça-se um historial clínico do doente, passível de uma análise cuidada e facilmente monitorizada.

1.3.2 Equipamento

Os equipamentos representam as entidades responsáveis por auxiliar o doente no processo de aquisição de valores a serem monitorizados. Em função das características do sistema de monitorização são definidos os equipamentos necessários para que cada doente possa possuir o seu perfil de monitorização. É possível uma enorme diversidade de equipamentos e adaptar a sua associação de acordo com o que se pretende monitorizar num determinado doente, podendo definir características de monitorização/aquisição distintas para cada doente. Cada equipamento possui um identificador único de modo a que seja facilmente identificável no processo de comunicação entre os diversos componentes que integram o sistema de monitorização.

1.3.3 Parâmetro

Os parâmetros são os indicadores que serão alvo de possível análise médica. Estão fortemente relacionados com a entidade Equipamento, uma vez que é mediante o(s) equipamento(s) que um doente possui que é possível descortinar quais os parâmetros que se podem monitorizar nesse mesmo doente.

1.3.4 Evento

Os eventos representam acontecimentos importantes que necessitam ser guardados. Permitem categorizar eventos referentes aos equipamentos e que podem desvirtuar a monitorização que está a ser realizada num determinado período.

1.3.5 Alerta

A entidade alerta corresponde a uma representação de um aviso que se pretende que surja quando ocorre um determinado acontecimento, seja mediante uma configuração prévia de um parâmetro referente a um doente ou devido a um aviso acionado pelo próprio.

1.3.6 Mensagem

A entidade mensagem é responsável por proporcionar um canal de comunicação unidirecional entre o médico e o doente.

1.3.7 Utilizador

A entidade referente ao utilizador do sistema permite identificar os diversos utilizadores que interagem com o sistema e os seus respectivos papéis (clínico/médico, administrador).

2 Base de Dados

2.1 Descrição

A base de dados deste projeto foi pensada de forma a permitir a concretização de 5 objetivos específicos:

1. Armazenar toda a informação, relativamente a doentes e aos valores que lhes sejam monitorizados;
2. Fornecer vistas sobre essa informação, que forneçam uma camada de abstração a aplicações externas que necessitem de a aceder;
3. Disponibilizar automatismos que desencadeiem operações em função da informação que vai sendo inserida no sistema, mais concretamente o que diz respeito ao registo dos parâmetros de actividade na base de dados em função das regras que podem ser estipuladas para a geração de alertas automáticos a serem despoletados como notificações do Sql Server.
4. Fornecer procedimentos que encapsulem a forma de acesso e manipulação da informação armazenada;
5. Disponibilizar serviços, que permitam notificar aplicações externas, em função das alterações sofridas pela informação armazenada.

O objetivo do sistema TelemOLD consiste na telemonitorização de doentes, no tratamento de oxigenoterapia de longa duração, neste âmbito, surgem dois contextos distintos que se relacionam, que são: Doentes e Equipamentos.

Desta forma, no contexto dos doentes, é necessária a existência de infraestruturas que permitam guardar:

- Os seus **Dados Administrativos**, ou seja, o seu número de doente e de utente, o seu nome, data de nascimento, sexo, estado (ativo ou falecido), telefone, morada, localidade, código e respetiva localidade postal e eventuais observações a efetuar relativas a estes dados;

- A sua **Informação Clínica**, isto é, qual o seu medico assistente, pneumologista, de família, o seu centro de saúde e eventuais observações clínicas;

- Os seus **Diagnósticos**, nomeadamente, os diagnósticos que lhe foram associados, ordenados por ordem de relevância;

- As suas **Prescrições de Oxigenoterapia**, ou seja, para determinada atividade (Esforço, Repouso e Sono), qual deve ser o seu debito de oxigénio (em L/min) e para que período é valida a prescrição. Relativamente a esta informação, deve ser ainda mantido um histórico das várias prescrições passadas ao doente ao longo do tempo;

- As suas **Prescrições de Ventilação**, isto é, no tratamento de oxigenoterapia é necessária a prescrição de um equipamento de ventilação (CPAP ou BiPAP) e a indicação das respetivas pressões associadas ao aparelho. Sobre esta informação, deve também ser mantido um histórico das prescrições passadas.

As prescrições de oxigenoterapia e ventilação são aspetos fundamentais para a análise clínica pneumológica do doente.

- Os seus **Alertas**, ou seja, neste tipo de sistema é importante que em função de determinadas ocorrências, se possam despoletar alertas para o mundo exterior com informação detalhada sobre qual a ocorrência e o porque da sua ocorrência, desta forma deve ser possível atribuírem-se ao doente alertas que sejam despoletados em função de um dado parâmetro (saturação de oxigénio, frequência cardíaca, COUNT, MET ou outros), que satisfaça determinada condição, durante um determinado período de tempo contíguo. Deve ainda ser possível que os vários alertas do doente possam ser estabelecidos para períodos bem definidos no tempo. Cabe ao médico decidir se o alerta deve ou não gerar o envio de uma SMS para o médico de serviço (caso seja associada uma mensagem ao alerta este será enviado por SMS, se não existir mensagem associada, a SMS não será enviada e quando o alerta ocorrer, ficará apenas registado no sistema).

- As suas **Mensagens**, ou seja, através deste sistema deve também ser disponibilizada a possibilidade do envio de SMS's para o telemóvel associado ao doente por parte da equipa clínica, sempre que seja necessário. O sistema deve ainda manter um histórico destes dados com informação sobre, se a SMS já foi enviada e em que instante, e/ou se já foi lida.

- Os seus **Equipamentos**, isto é, deve ser possível persistir no sistema os equipamentos atualmente associados ao doente (telemóvel, oxímetro, acelerómetro ou outros) e as suas respetivas configurações, isto é:

  - Para o telemóvel, qual o período de comunicação, ou seja, com que frequência deve o telemóvel enviar informação recolhida no doente para o sistema central;

  - Para o oxímetro, qual o período de amostragem, ou seja, com que frequência deve o oxímetro colher amostras do doente e enviar para o telemóvel;

  - Para o acelerómetro, qual o período de amostragem, ou seja, com que frequência deve o acelerómetro colher amostras do doente e enviar para o telemóvel e qual o modelo de regressão a usar para o cálculo dos METS (CRouter ou Freedson) *[1].

- Os seus **Registos Clínicos**, ou seja, os valores que são colhidos no doente ao longo da sua monitorização clínica. Associado a cada um destes valores temos sempre informação sobre o parâmetro a que este está agregado, o instante em que foi colhido e se é fidedigno ou não.

No contexto dos equipamentos, é necessária a existência de infraestruturas que permitam guardar:

- **Equipamentos**, cada equipamento é definido pela sua descrição, tipo e estado (ativo ou inativo);

*[1] Informação adicional sobre estes modelos pode ser consultada no *Paper Novel Method - METS & Counts*

- **Parâmetros de Equipamentos**, vão permitir caracterizar o modo de funcionamento de cada equipamento, ex.: o período de comunicação do telemóvel, o período de amostragem do acelerómetro, MAC address utilizado para a identificação do telemóvel, etc.
  Como foi dito anteriormente, pretende-se um sistema genérico, que trabalhe independentemente do tipo de equipamento utilizado, permitindo-se a utilização de múltiplos equipamentos em simultâneo.
  Foi, por isso, necessário criar um mecanismo de caracterização dos equipamentos suficientemente genérico e independente dele, não se desejando definir de forma rígida (*hardcoded*) os seus parâmetros de configuração.
  Numa primeira abordagem simplista poderemos pensar nos parâmetros como pares (<característica>,<valor>).
  Na atual implementação cada parâmetro é descrito pela sua descrição, pela indicação de se é ou não um parâmetro que identifica univocamente o equipamento e pela indicação de se é ou não modificável, isto é, se o utilizador não técnico o pode alterar, não dependendo de características estruturais dele.
  Associado a um equipamento existe sempre um conjunto de um ou mais parâmetros, com valores pré-definidos que podem depois ser alterados (os parametrizáveis) de acordo com o doente em questão.

- **Eventos**, cada equipamento pode lançar um ou mais eventos, durante a monitorização de um doente. Os eventos devem ser registados, criando-se um histórico, para que posteriormente sejam apresentados nos gráficos, em conjunto com os valores monitorizados. Existem alguns tipos de eventos que estão diretamente relacionados com os parâmetros a serem monitorizados, dando informação sobre o valor que foi lido e apresentando indicações sobre como agir na apresentação dos resultados: os traçados dos valores podem ser interrompidos, ou outro tipo de situação, a ser definida futuramente (ex.: alteração da cor ou tipo de traçado, etc.)

2.2 Nomenclatura Utilizada

Todos os objetos criados na Base de Dados seguem uma nomenclatura própria que se rege segundo as normas da Cast (visa a uniformização dos nomes dos objetos das bases de dados da empresa, com o propósito de facilitar possíveis e eventuais integrações de outras aplicações, bem como a sua respetiva manutenção. Esta politica, permite reduzir significativamente o peso dos trabalhos de ETL – *extract, tranform, load* – em cenários de integração).

Na tabela abaixo encontra-se uma explicação de como são constituídos os nomes dos tipos de objetos principais:

TIPO DE OBJECTO	CONSTITUIÇÃO DO NOME

Tabelas	É constituído por duas partes: a 1ª, por três letras que identificam o seu *schema*, escrito em maiúsculas; a 2ª parte, identifica a entidade ou associação de entidades representada, com as palavras separadas por _ e escritas com notação *PascalCase*.
Triggers	É constituído por três partes: a 1ª, identifica a tabela (em *PascalCase*); 2ª, a ação que executa; 3ª, a condicionante de execução (BfrIns – Before Insert, AftIns – After Insert, BfrUpd – Before Update, AftUpd – After Update, BfrDel – Before Delete e AftDel – After Delete). Exemplo: "DntAlerta_RegisterMessage_AftIns" (deve ser lido como, *trigger* sobre a tabela DNT_Alerta, para registo de mensagens, após a inserção de um alerta).
Stored Procedures	Os nomes dos *stored procedures* são constítuidos por quatro partes, separadas por *underscore* (_): 1. A primeira parte indica a ferramenta que irá utilizá-lo: a. *telemoldgen* indica que se trata de um *stored procedure* que será utilizado na geração automática de código das classes de acesso a dados; b. *telemold* indica que se trata de um *stored procedure* com funcionalidades especificas da aplicação telemold; c. Poderão ser criadas outras siglas para casos específicos, por exemplo um *stored procedure* que expandisse as funcionalidade do asp.net teria como sigla *aspnet*; 2. A segunda parte representa o nome da tabela principal (sem os underscores _), ex.: DntDoente; 3. A terceira parte é uma sigla que indica o tipo de operação que o *stored procedure* irá executar: a. Add – para realizar operações de Insert na tabela; b. Upd – para realizar operações de Update na tabela; c. Del – para realizar operações de Delete na tabela; d. Get – para realizar um operação de Select que devolve um único registo da tabela; e. List – para realizar operações de Select simples (com uma associação directa entre os argumentos do *stored procedure* e os campos das tabelas) sobre a tabela; f. Find – para realizar operações de Select complexas (onde os argumentos do *stored procedure* são manipulados antes de serem associados aos campos das tabelas) sobre a tabela; g. TabularList – quando o *stored procedure* estrutura o seu resultado num formato matricial; h. Check – quando o *stored procedure* se destinha a verificar a existência de informação na tabela Esta metodologia tem, ainda, a vantagem de ser percetível o tipo de resultado devolvido dos *stored procedure* (ex. um Check devolve um valor booleano, um List devolve uma lista de registos da tabela, etc.), que será útil na sua utilização

	Pode acrescer às siglas anteriores a indicação de qual o campo da tabela que estará envolvido no *stored procedure* através da concatenação da palavra *By* seguida do nome do campo. Por exemplo obter uma listagem de registos (*List*) de uma tabela baseada no campo *Codigo* será feita através de um *stored procedure* com a 3ª parte da forma *ListByCodigo*. 4. A última parte é opcional e será apenas utilizada em *overloads* de *stored procedures* (visto o SQLServer não suportar *overloads*). Caso exista terá o formato "OL{i}" onde {i} é o numero do *overload*. Por exemplo para criar um overload de um *stored procedure* que retorna uma listagem de registo baseados no campo tipo (a sigla da 3ª parte será *ListByTipo*), a 4ª parte será da forma OL1, obtendo-se um nome *…ListByTipo_OL1*. Esta última parte é usada para permitir definir comportamento adicional da ferramenta de geração de código .NET e portanto é comum apenas nas *stored procedures* onde a primeira parte seja telemoldgen. Como exemplo, para um *stored procedure* utilizado na ferramenta de geração de código (telemoldgen) que na tabela DNT_Doente (DntDoente) faz a listagem de registos (List) por código (ByCodigo) terá o seguinte nome: "telemoldgen_DntDoente_ListByCodigo".

Mais informação relevante, pode ser consultada no documento de normas da Cast, capítulo "Normas de Nomenclaturas dos Objectos".

Relativamente à organização das tabelas da base de dados, estas encontram-se agrupadas por 6 *schemas*, que as caracterizam de acordo com a sua informação da seguinte forma:

ABREVIATURA	NOME	DESCRIÇÃO
aspnet	ASP.NET	Agrupa as tabelas relacionadas com os serviços aplicacionais do ASP.NET (como o de autenticação, autorização, perfis, etc.).
telemold	TelemOLD	Agrupa *stored procedures* desenvolvidas à medida para a realização de uma atividade específica.
telemoldgen	TelemOLD Gen	Agrupa *stored procedures* desenvolvidas para efeitos de geração de código e que após a execução da ferramenta passaram a ser acedidas por código .NET na classe que representa a Tabela. Eliminar uma *stored procedure* neste *schema* implica elimina-la das classes .NET.
CNT	Contacto	Relativo aos contactos do doente.

DNT	Doente	Agrupa todas as tabelas que armazenam informação sobre doentes: alertas de doentes, sinais biológicos monitorizáveis, prescrições médicas, informações clínicas, mensagens da equipa médica e valores dos respetivos sinais observados.
EQP	Equipamento	Agrupa todas as tabelas que armazenam informação sobre equipamentos: parâmetros de equipamentos e respetivas associações, equipamentos associados a doentes e eventos gerados.
FNC	Funcionário	Agrupa as tabelas que contem informação sobre os membros das equipas médicas (médicos, enfermeiros, administrativos).
IGT	Integração	Agrupa os objetos cujo propósito existencial consiste essencialmente na integração com outras aplicações.

Os *schemas* CNT e FNC em situação normal seriam absorvidos pelo *schema* DNT, no entanto por motivos de compatibilidade com o sistema SIGUS foram organizados desta forma.

2.3 Tabelas

No âmbito da informação anterior, surge um conjunto de conceitos, representado por um conjunto de tabelas, que nos permitem chegar a um modelo sólido para a conceção base do sistema TelemOLD.

Nota: *A leitura do texto abaixo deve ser feita em conjunto com o Anexo B, por forma, a facilitar a interiorização dos conceitos nele contidos. Para informação de referência sobre os campos das tabelas e respetivos tipos sugere-se a consulta do Anexo E.*

Desta forma e seguindo esta linha de raciocínio, a tabela principal DNT_Doente (Doentes), permite listar todos os doentes do sistema, incluindo, a sua informação administrativa e clínica. Esta tabela relaciona-se com as seguintes:

- CNT_Codigo_Postal (Códigos Postais), esta tabela permite listar os códigos postais dos CTT e respetiva localidade postal (sendo compatível com a tabela respetiva do SIGUS).

- FNC_Funcionario (Funcionários), esta tabela permite listar os membros das equipas médicas (sendo compatível com a tabela respetiva do SIGUS).

- DNT_Diagnostico_Doente (Diagnósticos), esta tabela permite-nos obter a lista ordenada de diagnósticos associados a um doente;

- DNT_Diagnostico (Tipos de Diagnósticos) , esta tabela surge no contexto da anterior e permite-nos listar os diagnósticos que podem ser atribuídos a doentes;

- DNT_Oxigenoterapia (Prescrições de Oxigenoterapia), esta tabela permite-nos obter a lista de prescrições de oxigenoterapia associadas a um doente, com informação sobre o período da

prescrição, a atividade e o valor de débito de oxigénio (normalmente medido em L/min). Dentro da informação considerada inválida (através de *constraint* na base de dados e de controlo na aplicação web) para tabela incluem-se: períodos sobrepostos para a mesma atividade e débitos negativos;

- DNT_Actividade (Tipos de Atividades), esta tabela surge no contexto da anterior e permite-nos listar os tipos de atividades em que se podem realizar as prescrições de oxigénio a doentes;

- DNT_Ventilacao_Doente (Prescrições de Ventilações), esta tabela permite-nos obter uma lista de prescrições dos equipamentos ventilatórios associados a um doente, com informação sobre o período da prescrição, o ventilador e os seus valores pressóricos (normalmente medidos em cm H_2O). Dentro da informação considerada inválida (através de *constraint* na base de dados e de controlo na aplicação web) para esta tabela incluem-se: períodos sobrepostos e valores pressóricos negativos;

- DNT_Ventilacao (Tipos de Ventilação), esta tabela surge no contexto da anterior e permite-nos listar os tipos de ventiladores que podem ser prescritos a doentes;

- DNT_Parametro (Parâmetros), esta tabela permite-nos listar os parâmetros monitorizáveis num doente, assim como informação sobre: a cor para efeitos de desenho gráfico, a descrição, as casas decimais e a unidade para efeitos de formatação e apresentação, o mínimo e máximo para efeitos de validação/parametrização e o identificador cuja primeira letra representa o Equipamento a que pertence o parâmetro e a segunda letra serve para identificar o parâmetro;

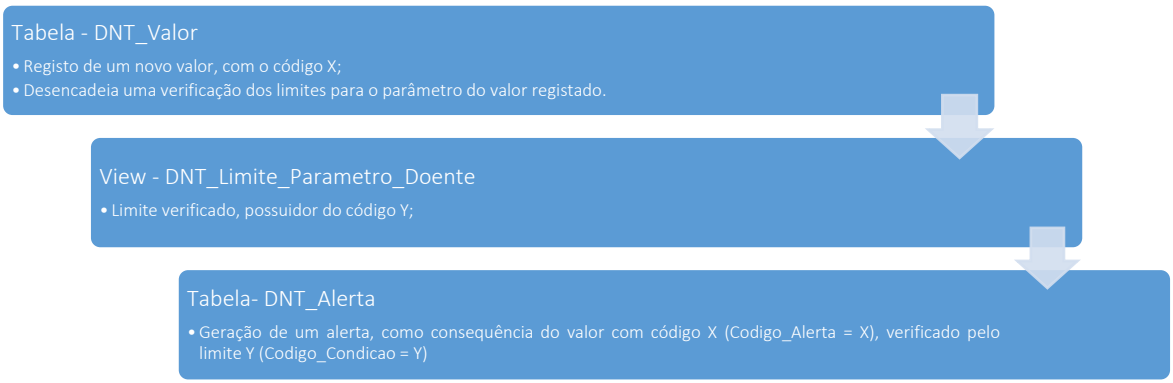
- DNT_Limite_Parametro (Limites dos parâmetros), esta tabela permite-nos, obter uma lista de limites associados aos valores recolhidos nos doentes, e em função desses limites enviar SMS's com informação sobre o estado do doente para o número do médico de serviço (criação de um alerta). Estes limites são definidos para um determinado período, onde, todos os valores que satisfação a condição composta pelo, operador e valor de referência durante o período especificado, dão origem a um alerta. Estes limites são verificados por cada valor recolhido ao doente.

- DNT_Limite_Parametro_Global (Limites globais dos parâmetros), esta tabela surge no contexto da anterior e permite listar os limites associados aos valores recolhidos nos doentes, sendo que estes são a um nível global: para todos os doentes.
  Desta forma podem ser criados alertas baseados em valores recolhidos aos doentes e despoletados por duas situações distintas: condições específicas ao doente (DNT_Limite_Parametro) e condições genéricas para todos os doentes em monitorização (DNT_Limite_Parametro_Global). Os limites locais aos doentes, quando idênticos (mesmo parâmetro, valor, condição e ativo) a um global, sobrepõem-se.
  A junção dos dois tipos de situações será feita através de uma vista (DNT_Limite_Parametro_Doente) que permite aceder às condições específicas do doente e às globais a todos os doentes (estas últimas são produzidas através do *crossproduct* da tabela DNT_Limite_Parametro_Global com a tabela DNT_Doente, excluíndo-se os casos em que a condição específica do doente, DNT_Limite_Parametro, se sobrepõe). A identificação da proveniência da condição limite (DNT_Limite_Parametro ou DNT_Limite_Parametro_Global) é

feita através de um campo booleano (Global) e do facto dos código dos limites específicos do doente serem negativos.

- DNT_Alerta (Alertas), esta tabela armazenará todos os alertas produzidos pelo sistema (componente fundamental deste projeto, por se tratar de um sistema de monitorização de doentes).
  Para além dos alertas gerados em função de um valor recolhido ao doente (criados com base nas duas tabelas anteriores) é, ainda, possível gerar-se um alerta em função de um evento lançado por um equipamento. Assim a tabela DNT_Alerta permite listar o histórico de todos estes tipos de alertas.
  A indicação do tipo de alerta (baseado em valores recolhidos ou eventos dos equipamentos) é feita através da ligação à tabela DNT_Tipo_Alerta (explicada já de seguida).
  O código que despoletou a geração do alerta (código da tabela EQP_Evento ou da DNT_Valor, tabela que regista os eventos e a tabela que regista os valores recolhidos, estas serão explicadas mais à frente) fica armazenado no campo Codigo_Alerta.
  Para os alertas baseados em valores recolhido ao doente é feito o registo (no campo Codigo_Condicao) do código da condição que o acionou (este será negativo para as condições provenientes da DNT_Limite_Parametro e positivo para as provenientes de DNT_Limite_Parametro_Global, mantendo-se o mecanismo construído na vista DNT_Limite_Parametro_Doente). Este campo (Codigo_Condicao) é preenchido com o código da tabela EQP_Tipo_Evento (tabela que identifica os tipos de eventos e será explicada mais à frente) para os alertas gerados por eventos.
  Por fim, para cada alerta será ainda registado o instante em que foi gerado e o instante em que foi enviada a SMS.
  O processo de geração de um alerta gerado a partir de valores recolhidos ao doente encontra-se exemplificado de seguida (o esquema para os alertas baseados em eventos será apresentado após a explicação das tabelas EQP_Evento, EQP_Tipo_Evento e associadas):

  - Alerta gerado por um valor do doente:



- DNT_Tipo_Alerta (Tipos de alerta), esta tabela permite obter a lista dos tipos de alertas (dependendo da sua proveniência: valores recolhidos aos doentes ou de eventos) que podem ser gerados pelo sistema. Para facilitar a manipulação dos alertas temos, ainda, aqui registada a tabela (EQP_Evento ou DNT_Valor) que deverá ser utilizada para consulta do Codigo_Alerta;

- DNT_Alerta_Por_Enviar (Alertas por enviar), esta tabela permite obter a lista dos alertas em espera para processamento e posterior envio (após o qual será atualizado o campo Instante_SMS da tabela DNT_Alerta);

- DNT_Mensagem (Mensagens), esta tabela permite listar as mensagens enviadas ou por enviar ao doente, com a indicação do instante de envio, estado de envio, estado da leitura e a respetiva mensagem. A consulta e atualização de parte desta tabela (campos Enviada, Lida e Instante) é realizada pelo serviço que comunica com os telemóveis e que também é responsável pelo envio dessas mensagens.

- DNT_Valor (Registos Clínicos), esta tabela permite-nos obter uma lista ordenada por instante dos valores recolhidos nos doentes, com indicação do parâmetro e respetiva qualidade do sinal (valor booleano) que recolheu o valor;

- DNT_Intervalo (Intervalos), esta tabela permite listar limites para a verificação percentual do tempo que cada parâmetro ocorreu entre o limite mínimo e máximo definidos, esta informação é importante no auxílio da análise dos exames clínicos efetuados pelos doentes e será apresentada no ecrã de monitorização dos doentes;

A outra tabela principal, é a tabela EQP_Equipamento (Equipamentos), que permite listar todos os equipamentos do sistema e respetiva indicação de estado ativo, descrição e tipo. Esta tabela relaciona-se com as seguintes:

- EQP_Tipo_Equipamento (Tipos de equipamentos), esta tabela permite-nos listar os tipos de equipamento utilizados na monitorização de doentes e em que cada tipo de equipamento possui um identificador único representado por um caracter;

- EQP_Parametro (Parâmetros), esta tabela permite-nos obter uma lista de parâmetros genéricos, associáveis a equipamentos independentemente do seu tipo. Cada parâmetro possui informação sobre:
    - Descrição;
    - Chave, indica se ao nível da configuração do equipamento o parâmetro é o identifica univocamente ou não (por exemplo, o endereço MAC identifica univocamente um equipamento);
    - Parametrizável, indica se o parâmetro é ou não configurável pelo utilizador não técnico (ex.: clínico), a indicação de não personalizável indica que apenas o administrador técnico o deve editar;
    - Valores por Omissão, valores válidos para o parâmetro. Caso haja mais de um serão representados através numa *string* separada por vírgulas (CSV);
    - Tipo, indica o tipo do componente ASP.NET que será usado na aplicação, para a apresentação dos seus valores por omissão (ex.: caso se tenha como valores por omissão uma lista de alternativas, separadas por virgulas no campo anterior, pode-se indicar como método de apresentação um System.Web.UI.WebControls.DropDownList, o sistema irá utilizar as alternativas para “alimentar” esse controle);
    - Unidade, indica a unidade, se aplicável, do valor assumido pelo parâmetro e será apresentada na interface gráfica.

    - Identificador, caracter que permite identificar cada parâmetro univocamente.

- EQP_Parametro_Equipamento (Parâmetros dos equipamentos), esta tabela faz a associação entre os equipamentos e os parâmetros com os respetivos valores pré-definidos.

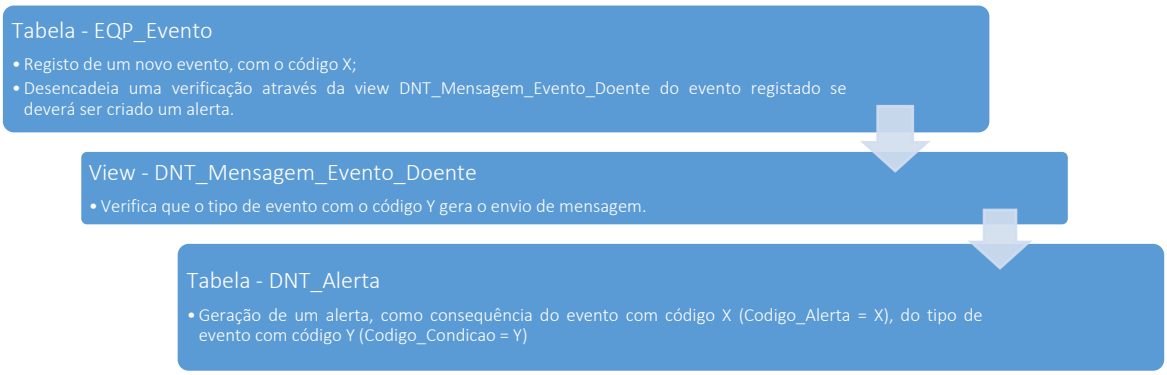
- EQP_Parametro_Doente_Global (Parâmetros de equipamentos associados ao doente), esta tabela tem, para cada doente, a definição dos respetivos valores para cada parâmetro.
  Devido à possibilidade de haver alterações de valores de parâmetros ao longo do tempo existe na tabela (campos Data_Inicio e Data_Fim) o registo do período de tempo em que o valor desse parâmetro foi utilizado, criando-se um historial dos valores utilizados.
  Baseando-se na informação dos parâmetros que identificam univocamente um equipamento (campo Chave da tabela EQP_Parametro) é possível determinar os equipamentos que estão a ser utilizados (associados) por cada doente.
  Quando se altera uma associação de um equipamento a um doente serão introduzidos os respetivos parâmetros nesta tabela, sendo inicializados com os valores existentes na EQP_Parametro_Equipamento.

- EQP_Tipo_Evento (Tipo de eventos), esta tabela permite listar os tipos de eventos que podem ser lançados pelos equipamentos utilizados pelos doentes. Para cada tipo de evento é possível definir, a título global, se a sua ocorrência implica ou não o envio de SMS para o número do médico de serviço (criação de um alerta).

- DNT_Mensagem_Evento (Mensagens de eventos), esta tabela permite, para um doente específico, indicar se deve ou não, ser enviada a mensagem no caso da ocorrência do tipo de evento, sobrepondo-se à informação da tabela anterior.
  A junção dos dois tipos de situações será feita através de uma vista (DNT_Mensagem_Evento_Doente) que permite aceder às condições específicas do doente e às globais a todos os doentes (estas últimas são produzidas através do *crossproduct* da tabela EQP_Tipo_Evento com a tabela DNT_Doente, excluindo-se os casos em que a condição específica do doente, DNT_Mensagem_Evento, se sobrepõe). A identificação da proveniência da condição (DNT_Mensagem_Evento ou EQP_Tipo_Evento) é feita através de um campo booleano (Global) e do facto do código dos limites específicos do doente serem negativos.

- EQP_Evento (Eventos), esta tabela permite obter o histórico de eventos gerados pelos equipamentos associados a doentes, com indicação do instante do registo deste evento e o seu tipo. Caso seja de enviar uma mensagem, será produzido um alerta, cuja criação segue o seguinte esquema:

- Alerta gerado por um evento produzido por um equipamento:



- EQP_Tipo_Evento_Parametro, esta tabela permite definir, para cada tipo de evento, quais os parâmetros do doente que não devem ser desenhados na apresentação gráfica, na aplicação (esta tabela poderá ser expandida no futuro para incluir outros tipos de situação, ex. alteração de cor ou tipo de traçado).

2.4 Automatismos

A necessidade da criação de alertas dependendo da informação introduzida na base de dados (valores recolhidos aos doentes e eventos gerados pelos equipamento) obrigou à criação de *Triggers*.

Quando um valor recolhido ao doente é introduzido na tabela DNT_Valor é despoletado um *trigger* (DntValor_GenerateAlerts_AftIns) que, conforme explicado anteriormente (após a verificação de uma condição excecional registada para o doente), introduz a informação na tabela DNT_Alerta. Este *trigger* depende de:

- Vista DNT_Limite_Parametro_Doente;
- Tabela DNT_Tipo_Alerta;
- Tabela DNT_Alerta;
- *Stored Procedure* telemold_DntLimiteParametro_CheckInterval.

Analogamente quando um evento é gerado (informação introduzida na tabela EQP_Evento) é despoletado um *trigger* (EqpEvento_GenerateAlerts_AftIns) que, conforme explicado anteriormente (após a verificação da necessidade de envio de mensagem para o doente), introduz a informação na tabela DNT_Alerta. Este *trigger* depende de:

- Vista DNT_Mensagem_Evento_Doente;
- Tabela DNT_Tipo_Alerta;
- Tabela DNT_Alerta.

Por último, quando a informação é adicionada à tabela DNT_Alerta é despoletado um *trigger* (DntAlerta_RegisterMessage_AftIns) que insere o código do novo alerta na tabela DNT_Alerta_Por_Enviar, para que este seja enviado por SMS. Este *trigger* depende de:

- Dnt_Alerta_Por_Enviar.

2.5 Vistas

No que diz respeito às vistas podemos agrupá-las por duas categorias:

- Vistas Internas – usadas para facilitar a consulta de informação em vários formatos e formas;
- Vistas de Integração – usadas para fornecer informação sobre determinadas entidades de forma sucinta e assertiva, de maneira a abstrair aplicações externas da complexidade do modelo de dados.

Foram criadas três vistas internas, que se explicam de seguida.

Como foi explicado anteriormente, a existência de condições limite que darão origem a alertas podem ser definidas a título individual, especificamente para o doente (DNT_Limite_Parametro), ou de forma global (DNT_Limite_Parametro_Global), aplicáveis a todos os doentes. A vista DNT_Limite_Parametro_Doente centraliza a informação dessas duas tabelas, disponibilizando para cada doente as condições que foram definidas a título global (produzidas através do *crossproduct* da tabela DNT_Limite_Parametro_Global com a tabela DNT_Doente, excluindo-se os casos em que a condição específica do doente, DNT_Limite_Parametro, se sobrepõe) e a título especifico do doente (DNT_Limite_Parametro). A identificação da proveniência da condição limite (DNT_Limite_Parametro ou DNT_Limite_Parametro_Global) é feita através de um campo booleano (Global) e do facto do código dos limites específicos do doente serem negativos. Esta vista está relacionada com:

- Relatório de Alertas, o relatório de alertas da aplicação Web consulta esta vista.
- *Stored Procedure* telemold_DntAlerta_ListByAlarmesDoente, esta *stored procedure* também consulta esta vista.

Analogamente, a possibilidade de envio de mensagens produzidas pelos eventos podem ser definidas a título global (EQP_Tipo_Evento) aplicáveis a todos os doentes ou de forma individual, especificamente para o doente (DNT_Mensagem_Evento). A vista DNT_Mensagem_Evento_Doente centraliza a informação dessas duas tabelas, disponibilizando para cada doente a indicação de envio de mensagem definidas a título global (produzidas através do *crossproduct* da tabela EQP_Tipo_Evento com a tabela DNT_Doente, excluindo-se os casos em que a condição específica do doente, DNT_Mensagem_Evento, se sobrepõe) e a título especifico do doente (DNT_Mensagem_Evento). A identificação da proveniência da condição (DNT_Mensagem_Evento ou EQP_Tipo_Evento) é feita através de um campo booleano (Global) e do facto do código dos limites específicos do doente serem negativos. Esta vista está relacionada com:

- Relatório de Alertas, o relatório de alertas da aplicação web consulta esta vista.
- *Stored Procedure* telemold_DntAlerta_ListByEventosDoente, esta *stored procedure* também consulta esta vista.

A última vista interna, DNT_Valor_Parametro, foi criada para agrupar os parâmetros de actividade registados nos diversos instantes de um determinado doente, existindo a indicação de imprecisão de sinal caso num determinado instante os valores registados não sejam fidedignos. Esta vista é acedida pela *stored procedure* telemold_DntValor_TabularList, que por sua vez é consumida pelo relatório de valores da aplicação web.

Portanto o que esta vista permite é pegar na informação que se encontra "listada na horizontal" e dispô-la verticalmente, criando aquilo a que se pode chamar de uma *pivot table*:

Ao invés de termos:

Parametro	Instante	Valor
COUNT	01-01-2010 09:00	1000
FC	01-01-2010 10:00	100
SPO2	01-01-2010 10:00	80

Passamos a ter:

Instante	FC	COUNT
01-01-2010 09:00	NULL	100
01-01-2010 10:00	80	100

A informação sobre os doentes existentes e respetivos telemóveis associados (através do parâmetro que contém o MAC Adress do telemóvel) está disponibilizada na vista EQP_Doente. Esta vista integra com a aplicação da plux e depende das vistas (que também integram com a aplicação da plux):

- BIOPLUX;
- LOG;
- MOBILES;
- NONIN.

A informação sobre os parâmetros dos diversos equipamentos associados aos doentes está disponível através da vista EQP_Parametro_Doente.

Quando um sensor não é utilizado pelo doente, seja o acelerómetro (BIOPLUX) ou o oximetro (NONIN) o doente deve possuir um sensor *dummy* com valores a zero que permitem ao Plux Socket Server constatar que o sensor está desligado e otimizar nesse sentido a aplicação para os sensores em uso (este work-flow foi imposto pela Plux). Para resolver o problema era necessário existir um equipamento *dummy* que se pudesse associar a todos os doentes nesta condição, no entanto a possibilidade de associação de um equipamento em simultâneo a mais de um doente iria tornar o modelo de dados inconsistente. Por este motivo, surgem estas vistas que "geram" esse equipamento (virtual) e integram com a vista EQP_Parametro_Doente. Foram criadas duas vistas (IGT_Parametro_Doente_BIOPLUX e IGT_EQP_Parametro_Doente_NONIN) para os dois tipos de equipamentos.

A vista BIOPLUX permite obter os parâmetros dos acelerómetros associados a doentes. Esta vista depende para isso da vista EQP_Doente e da vista IGT_EQP_Parametro_Doente_BIOPLUX, é aqui que as vistas explicadas anteriormente entram em ação, a vista BIOPLUX retorna todos os parâmetros dos acelerómetros associados a doentes, no entanto para os doentes que não possuam acelerómetro são gerados parâmetros na mesma para esses doentes, mas todos com valores '0', através da vista IGT_EQP_Parametro_Doente_BIOPLUX.

A vista NONIN permite obter os parâmetros dos oxímetros associados a doentes, na mesma lógica da vista anterior, para os doentes sem oxímetros irão ser gerados parâmetros com valores '0' através da vista IGT_EQP_Parametro_Doente_NONIN.

A vista DATA é idêntica à tabela DNT_Valor a menos de um campo, o "Codigo", que é um *identity* mantido pela própria base de dados.

A vista EVENTS é idêntica à tabela EQP_Tipo_Evento a menos do campo "Mensagem", que apenas é relevante para a criação dos alertas.

A vista LOG, relacionada com EQP_Evento, permite obter os eventos gerados, com indicação do telemóvel que os originou e o respetivo instante.

A vista MEASUREMENTS permite obter o código e descrição dos parâmetros monitorizados aos doentes (baseada na DNT_Parametro).

A vista MOBILES permite obter os parâmetros dos telemóveis, nomeadamente o período de amostragem associados a doentes.

2.6 Serviços

Embora a o SQL Server disponibilize um vasto conjunto de serviços, neste projeto tirou-se partido apenas dos serviços de notificação. Dentro da categoria dos serviços de notificação existem duas possibilidades ao nível do protocolo de comunicação com aplicações ou base de dados, que se resumem ao uso de:

- Serviços personalizados (*tailor-made)* e portanto feitos à imagem dos requisitos anexos ao problema a resolver e sendo este o caso é necessário definir o tipo de mensagens a serem enviadas, o seu formato, as *queues* que iram fazer a sua gestão e respetivo protocolo a usar, *etc…*, é um processo trabalhoso que só se justifica em cenários deveras mais complexos do que o nosso.
- *Service Broker* ou Facilitador de Serviços, fornece um serviço pré-configurado que apenas requer por parte do utilizador subscrição e renovação de notificações.

No nosso caso, adotamos a 2ª opção, para a integração com a aplicações externas à base de dados mas que necessitam de acesso à sua informação ou cujo o seu funcionamento dependa de alterações de estado da base de dados.

Em particular o envio de SMS (aplicação explicada em capítulo próprio) será realizado por uma aplicação externa que subscreve este serviço.

Atenção/Nota: Sempre que seja feito um *restore/attach* da base de dados num SGBD SQL Server é necessário correr o script abaixo para a activação do *Service Broker*:

Código T-SQL

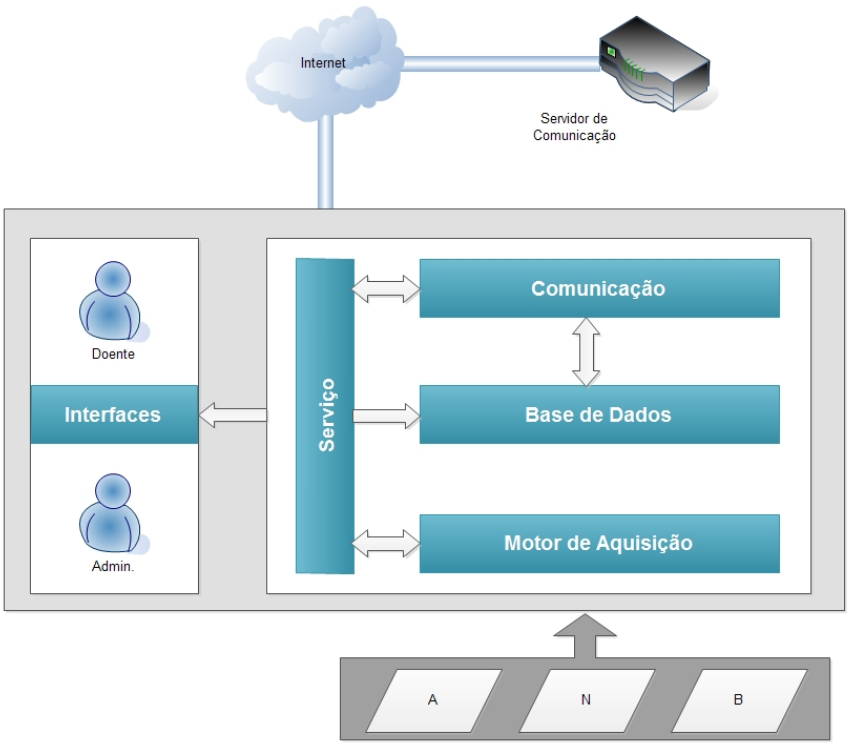
```
USE Telemold; GO;
ALTER DATABASE Telemold SET ENABLE_BROKER WITH ROLLBACK IMMEDIATE; GO;
GRANT SUBSCRIBE QUERY NOTIFICATIONS TO telemold; GO;
ALTER AUTHORIZATION ON DATABASE::[Telemold] TO telemold; GO;
```

3 Aplicação Móvel

3.1 Descrição

- Aplicação Android responsável pela aquisição de parâmetros relativos ao doente, processamento e envio dos mesmos para um servidor remoto

3.2 Arquitetura



A aplicação móvel está subdividida em vários módulos, responsáveis pela interface, comunicação, aquisição e processamento da informação.

3.3 Interface

- Este módulo contém todas a interfaces relativas ao utilizador comum e administrador. A interface para o administrador é apresentada após 7 cliques no logotipo da aplicação, e requer um username e password para ser acedida.

3.4 Serviço

- Responsável pela camada lógica do sistema. Faz a interação entre a interface e os restantes módulos. Este módulo faz também a comunicação entre os módulos de Comunicação, Base de Dados e Motor de Aquisição.

3.5 Motor de Aquisição

- O motor de aquisição é módulo do sistema que é responsável pela aquisição de parâmetros. Este comunica com os vários dispositivos disponíveis, processa a informação recebida, e envia para o módulo Serviço para ser guardado.

3.5.1 Base

- A base do motor é composta por classes que representam genéricamente dispositivos e parâmetros, e também classes que tratam do processamento e amostragem dos parâmetros adquiridos pelos dispositivos assim como a gestão da bateria e comunicação com os mesmos.

3.5.2 Dispositivos

3.5.2.1 *Acelerómetro (A)*

- Aquisição de valores de aceleração
- Cálculo de COUNTs

$$CPM = \sum_{i=0}^{EDseg \times 10} |\Delta g| \times \frac{60seg \times 10}{EDseg^2 \times 0.01664g \cdot seg^{-1} \cdot count^{-1}}$$

- Cálculo de METS

3.5.2.2 *Oxímetro (N)*

- Aquisição de valores de frequência cardíaca e percentagem de oxigénio

3.5.2.3 *Outros*

- Como configurar um novo dispositivo
    - Exemplo: Bitalino (B)

3.6 Base de dados

- Base de dados local
- Parâmetros de actividade
- Logs

3.7 Comunicação

- Envio de parâmetros
- Envio de logs
- Receção de configurações
- Receção de alertas

4 Aplicação Web

4.1 Descrição

Na conceção da aplicação Web para o sistema TelemOLD pretendeu-se tornar o acesso à informação existente na base de dados o mais intuitivo possível do ponto de vista clínico. A aplicação será essencialmente utilizada por clínicos onde, pela sua formação base, privilegiam a visão do doente como um todo.

A função primordial do sistema será a de possibilitar de forma imediata a consulta/alteração relativa aos doentes, nas suas múltiplas vertentes: informação administrativa, clínica, diagnósticos, prescrições de oxigenoterapia e de ventilação, alertas, mensagens enviadas, equipamentos associados e respetivos parâmetros e dados monitorizados. Acessoriamente deverá ter um robusto sistema de relatórios que permita a geração e impressão de mapas estatísticos e/ou de informação essencial para o processo de diagnóstico e tratamento dos doentes. Por último, deverá permitir a o registo/alteração da informação de configuração necessária para se realizar a monitorização clínica do doente (ex.: criação e parametrização de novos equipamentos, gestão de utilizadores, etc.).

4.2 Arquitectura

A aplicação web foi concebida e dividida em cinco projetos distintos acoplados na mesma solução do Visual Studio, que são:

- Web Application Telemold;
- Cast.Web;
- Cast.Net;
- DotNetNukeLibrary;
- ServerSocketTelemOLD.Database

O projeto “Web Application Telemold” é o projeto principal que contém as diversas páginas do website.

O projeto “Cast.Web” é uma biblioteca da empresa Cast, que contém um conjunto de classes que serviram de base para o desenvolvimento da aplicação web.

O projeto “Cast.Net” é uma biblioteca da empresa Cast, referente ao envio de SMS’s. Foi desenvolvido primordialmente para estar associado à aplicação de envio de SMS’s que se encontra noutra solução e que é explicada em capítulo próprio. No entanto, foi incluído na presente solução por se ter desenvolvido uma página web que permite enviar SMS’s.

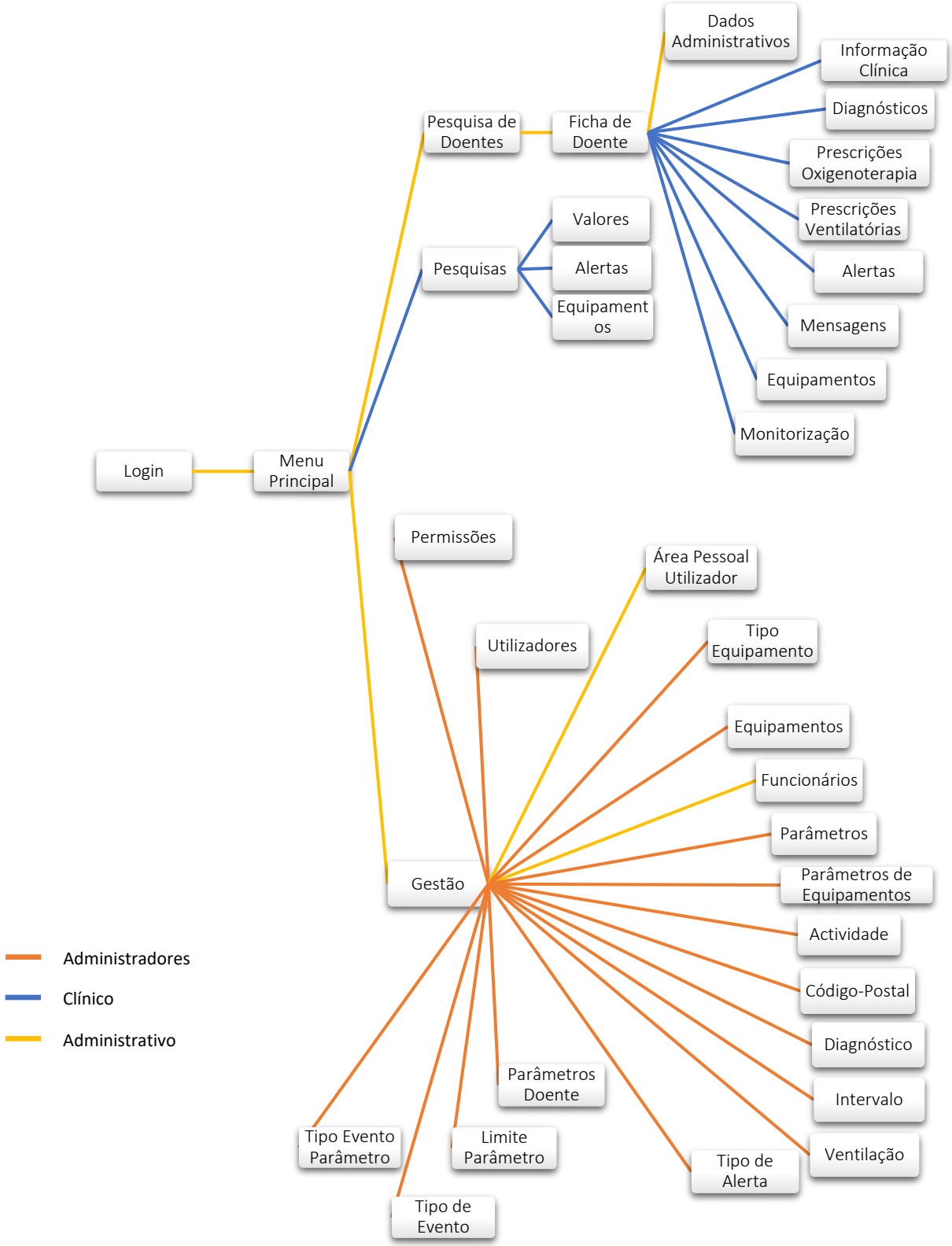
O projeto “DotNetNukeLibrary” é uma *framework opensource* que permite o desenvolvimento de aplicações web, com inúmeras funcionalidades. No entanto, a sua utilização restringiu-se apenas às classes de acesso a dados.

Por último, o projeto “ServerSocketTelemOLD.Database” corresponde a uma representação de uma nova camada de acesso aos dados desenvolvida com a tecnologia *EntityFramework* e que foi utilizada na aplicação web, assim como no servidor de comunicação.

4.3 Componentes/Tecnologias

- DNN, CAST, D3js, EF

4.4 Mapa da Aplicação



Como é possível verificar pelo mapa ilustrado anteriormente, a aplicação web divide-se em três principais zonas: doentes, pesquisas e gestão.

4.5 Monitorização

4.5.1 Introdução

Na ficha de doente está disponível a funcionalidade de monitorização que permite monitorizar os valores adquiridos pelos equipamentos ao longo do tempo.

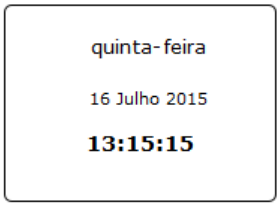
Os requisitos para esta funcionalidade centram-se na visualização, através de um gráfico de linhas, da relação tempo-valor para cada parâmetro a ser avaliado no doente; na configuração do respetivo gráfico, podendo-se mudar o período avaliado, os parâmetros visíveis, e o dinamismo do mesmo; e na visualização de estatísticas relativas aos valores apresentados no gráfico.

Para a implementação do gráfico foi utilizada a biblioteca de JavaScript D3.js. Esta biblioteca orientada aos dados, permite criar diferentes representações gráficas usando HTML, SVG e CSS, de acordo com os mais recentes *standards* para a *web*. O facto de utilizar SVG para a representação gráfica, que permite que os objetos sejam criados no DOM, e depois manipulados ao longo do tempo, em detrimento do *canvas* do HTML5, que obriga a um desenho de todos os objetos sempre que há uma atualização, levou a que as bibliotecas JavaScript que trabalham com SVG fossem avaliadas. Neste capítulo, o D3.js foi o que à data de implementação pareceu mais adequado ao problema, sendo também consideradas sua documentação, exemplos e comunidade de suporte. Outras bibliotecas avaliadas à data não reuniam melhores condições que o D3.js para a resolução do problema, como o Raphael.js, e por isso foram preteridas.

4.5.2 Descrição

4.5.2.1 *Componentes*

A monitorização é composta por um gráfico de linhas, onde cada linha corresponde a um parâmetro; por uma legenda, onde estão indicados os parâmetros; por um quadro de estatísticas, onde são mostradas estatísticas relativas aos valores apresentados; por uma zona de opções, que contém as opções de zoom, período temporal e controlo da visualização de componentes internos do gráfico; e por um relógio que mostra a hora atual no servidor.

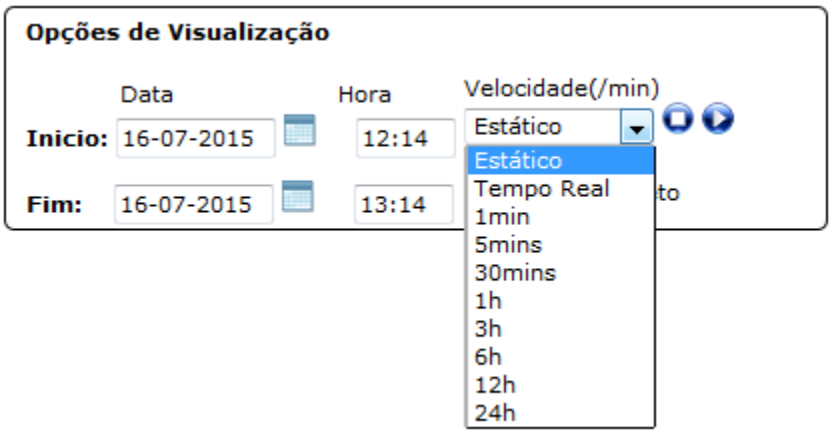


4.5.2.2 *Modos*

Existem vários modos de visualização do gráfico, quer temporais, quer ao nível do aspeto visual.

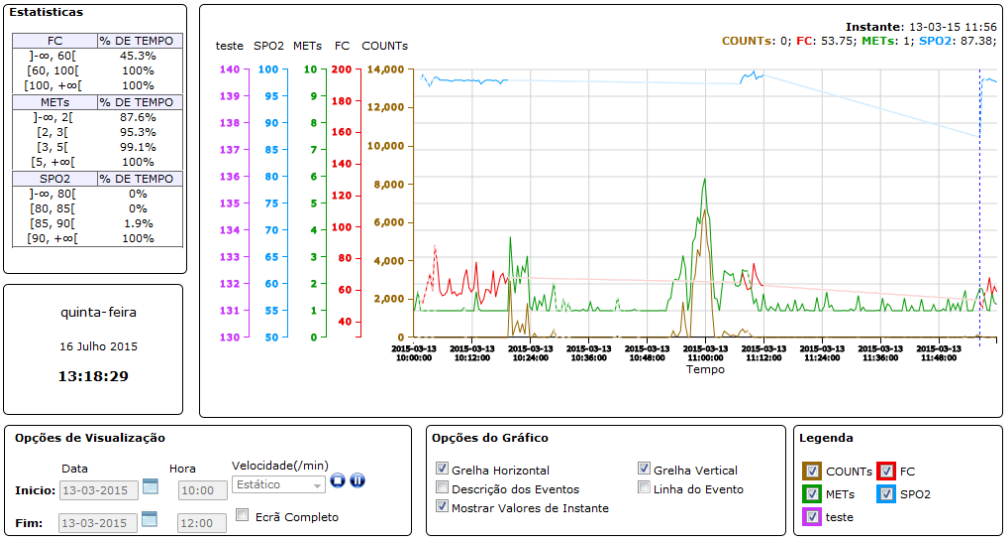
Dentro dos modos temporais, existe o modo estático, que tal como o nome indica, não tem qualquer animação, e refere-se ao período de tempo escolhido. Existem depois os modos dinâmicos, que têm uma velocidade associada para o período escolhido, existindo neste grupo um modo especial

denominado de “Tempo Real”. Este modo tem uma atualização a cada segundo, e mostra os valores relativos aos últimos 5 minutos, com um atraso programado de 2 minutos (necessário devido ao algoritmo de cálculo dos METS).

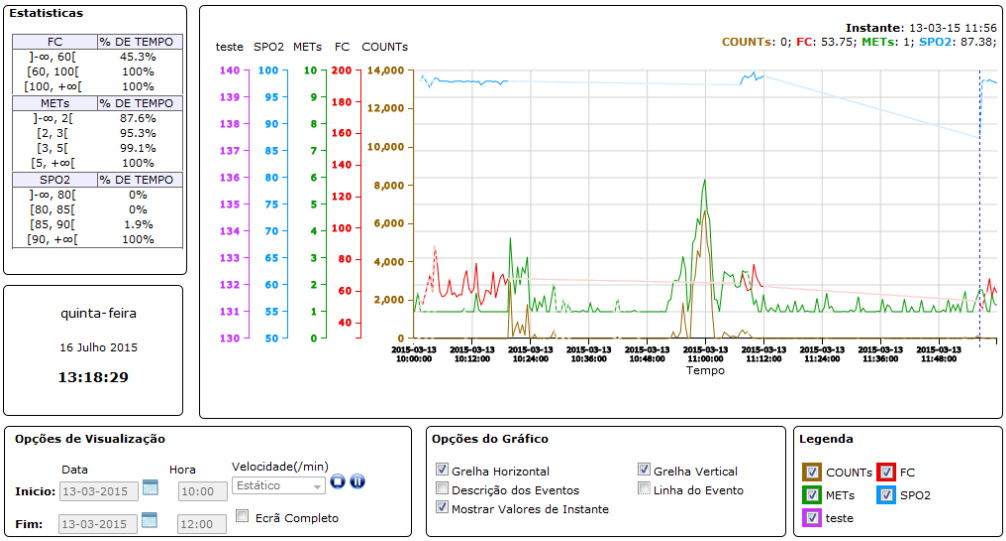


Ao nível do aspeto, existe o modo de ecrã normal que é o selecionado por omissão, e o modo de ecrã completo, que ocupa todo o espaço disponível no ecrã, ajustando os componentes da página e removendo os que não são estritamente necessários, tais como o relógio e as estatísticas.

Modo Normal:



Modo Ecrã Completo:



4.5.3 Entidades

Foram criadas 3 entidades específicas para a utilização do gráfico em D3.js, que representavam toda a informação necessária a ser visualizada. Estas entidades são representadas no formato JSON, e estão descritas nos seguintes pontos.

4.5.3.1 *Parameter*

Esta entidade representa um parâmetro visualizado no gráfico, e é composta por nome do parâmetro, cor a ser apresentada, valores mínimos e máximos a mostrar no eixo, e os conjuntos de valores relativos à aquisição de dados e eventos.

4.5.3.2 *Record*

A entidade *Record* representa um valor do parâmetro num determinado instante. Este valor pode ter sido adquirido com ou sem fiabilidade, cujo valor deverá ser também representado no gráfico.

4.5.3.3 *Log*

Os eventos relevantes para a monitorização, são representados pela entidade *Log*, e são compostos pela sua data e descrição.

4.5.4 Funções Principais

As funções principais para o funcionamento do gráfico e dos diferentes modos encontram se no ficheiro “D3ChartMain.js” e contêm as seguintes funções:

4.5.4.1 *DrawGraphic*

Esta função é responsável pela criação do gráfico, e de todos os seus componentes, tais como eixos e linhas. Dentro deste método é possivel identificar alguns pontos importantes:

4.5.4.1.1 Amostragem

No DrawGraphic caso o número de dados para cada "linha" do gráfico seja duas vezes superior á largura do gráfico, é efetuada então a teoria da amostragem.

A teoria é aplicada por razões de desempenho do gráfico e tem como objectivo a redução de pontos a desenhar alterando o minimo possivel a estruturas das linhas e do gráfico.

Para isso é efectuada uma recolha dos pontos especifica para determinados intervalos de forma uniforme.

Depois de efetuada, as linhas que tinham ultrapassado o limite irão ter no máximo, duas vezes a largura do gráfico em pontos.

4.5.4.1.2 Eixos do X e Y

- Criação dos eixos

```
    //X AXIS
    xAxis = d3.svg.axis()
        .scale(x)
        .orient("bottom")

.tickValues(arrayHelp).tickFormat(d3.time.format(d3FormatT
ime));

//Y AXIS
createYAxis(data);

function createYAxis(array) {
    allAxis = [];

    for (var index in listY) {
        if (index < Object.size(array)) {
            var yAxis = d3.svg.axis()
                .scale(listY[index])
                .orient("left");
            allAxis.push(yAxis);
        }
    }
}
```

- Colocação no gráfico

```
        //Append the Axis
        xAxisNodes = layer1.append("g")
            .attr("id", "xAxisId")
            .attr("class", "x axis")
            .attr("transform", "translate(0," + height + ")")
            .attr("stroke", "black")
            .attr("stroke-width", strokeWidthNormal)
            .style("font-size", xAxisFontSize)
            .call(xAxis)
        .selectAll(".tick text")
        .call(wrap, 100);


    //Append all the yAxis
    for (var index = 0; index <
    Object.size(listCheckboxAddAux) ; index++) {

                layer3.append("rect")
                    .attr("id", "yAxisIdBox" + index)
                    .attr("class", "yAxisClassBox")
                    .attr("x", 0)
                    .attr("y", yAxisClassBoxY)
                    .attr("width", axisMarginLength) // PEDRO
    -> tem-se de mudar
                    .attr("height", yAxisClassBoxHeight)
                    .style("fill", "white")
                    .attr("transform", "translate(" +
    sumMargin + " ," + -height + ")");  //"Distance" from the
    other Y's Axis;

    }
```

4.5.4.1.3 Criação das linhas

- Criação das variaveis das linhas

```
    for (var index = 0; index < Object.size(listY) ;
index++) {
        var valueline = d3.svg.line()
            .x(function (d) { return x(d.datetime); })
            .y(function (d) { return
listY[index](d.value); });

        valueLines.push(valueline);
                }
```

- Colocar no gráfico

```
  layer1.append("path")      // Add the valueline path.
                .attr("class", "line")
                .attr("id", "line" + data[index].label)
                .style("stroke", data[index].color)     //
line color
                .style("stroke-width", strokeWidthNormal)
                .style("fill", "none")
                .attr("d",
valueLines[index](data[index].records));
```

4.5.4.2 *UpdateXAxisDomain*

Nesta função, o conteúdo do gráfico é ajustado aos novos valores do X. Esta função é utilizada nos modos dinâmicos, tais como o "Tempo Real" e outras velocidades. Atualiza as linhas e legendas do eixo do X

4.5.4.3 *UpdateYAxis*

Quando o utilizador remove ou adiciona um parâmetro da lista dos parâmetros visíveis, esta função é chamada e tem como objetivo esconder ou mostrar o respetivo eixo e ajustar as dimensões do gráfico.

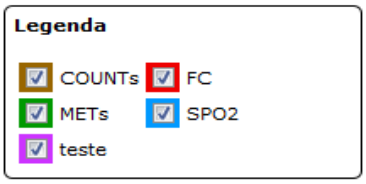
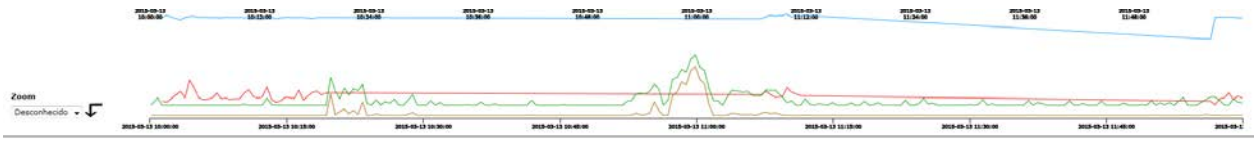


Imagem 5 : Lista de parâmetros

4.5.4.4 *Brushed*

Durante o modo de ecrã completo, é disponibilizado ao utilizador uma zona para se fazer zoom. Ao usar essa opção, esta função é chamada e faz o respetivo ajuste no gráfico.



4.5.4.5 *Mousemove*

A função mousemove é responsável pela atualização e visualização dos valores do instante no gráfico selecionados pelo movimento do rato ou toque do dedo no gráfico em ecrãs tácteis.

4.5.5 Funções Secundárias

As funções secundárias e/ou auxiliares encontram se no ficheiro "MainAuxiliarFunctions.js" e contêm as seguintes funções:

4.5.5.1 *UpdateValueLines*

Função responsável pela atualização das linhas dos parâmetros conforme os valores do eixo do X e dos dados atuais.

4.5.5.2 *UpdateLogs*

Função responsável pela atualização das linhas dos eventos conforme os valores do eixo do X e dos dados atuais.

4.5.5.3 *UpdateDashLines*

Função responsável pela atualização das linhas a tracejado conforme os valores do eixo do X e dos dados atuais.

4.5.5.4 *UpdateTransparentLine*

Função responsável pela atualização das linhas “transparentes” conforme os valores do eixo do X e dos dados atuais.

4.5.6 Configurações

As configurações relativas à monitorização encontram-se no ficheiro GráficoVariaveis.js.

Dentro desse ficheiro é possivel alterar as variáveis relativas ás dimensões do gráfico:

```
//Canvas Margins,Width and Height
var margin = { top: 60, right: 0, bottom: 60, left: 0 },
    width = 800 - margin.left - margin.right,
    height = 390 - margin.top - margin.bottom,
    widthXDomain = width; //Used for the X Axis
var svgHeight = height + margin.top + margin.bottom;
var svgWidth = width + margin.left + margin.right;

var currentDefaultWidth = width;
```

Definir formatos e funções relativos a datas:

```
//Canvas Margins,Width and Height
var parseDate = d3.time.format("%Y-%m-%d %H:%M:%S").parse;
var formatTime = d3.time.format("%e %B");
var bisectDate = d3.bisector(function (d) { return d.datetime; }).left; //Usado no metodo Mousemove()
```

E ainda outras variáveis usadas no processamento dos dados recebidos para serem mostrados no gráfico.

4.5.7 Comunicação com o servidor

Exemplo de comunicação com o servidor:

```
var $allCheckbox = $('.allCheckbox :checkbox');

$('.allCheckbox :checkbox').each(function () {
    listCheckbox += $(this).closest("div").attr("id") + ",";
});
var listArray = listCheckbox.substring(0, listCheckbox.length - 1);

var pageUrl = '<%= ResolveUrl("~/FichaDoente.aspx/GetJSonData") %>';
var parameter = { "doente": doente, "parametros": listArray };
$.ajax({
    type: 'POST',
    url: pageUrl,
    data: JSON.stringify(parameter),
    contentType: 'application/json; charset=utf-8',
    dataType: 'json',
    success: function (data) {
        DrawGraphic(data.d, linhaEventoCheckbox, descricaoEventoCheckbox, verticalGridLine,
horizontalGridLine, instantValue);
    },
    error: function (data, success, error) {
        alert("Error : " + error);
    }
});
```

5 Serviço de Comunicação

5.1 Descrição

- Serviço de comunicação entre a aplicação móvel e a base de dados

5.2 Arquitetura

- Interpretador
- Interação com a Base de dados
- Sockets

5.3 Protocolo de Comunicação

5.3.1 Operações

- Comunicação inicial: cliente-servidor

  É uma mensagem de iniciação de conversação entre o cliente e o servidor, que tem como principal objectivo alertar o servidor que um cliente cujo equipamento possui o MAC Address enviado na mensagem se pretende conectar.

- Comunicação inicial: servidor-cliente
- Mensagem de resposta do servidor ao cliente enviada após recepção da mensagem de iniciação de conversação entre o cliente e o servidor. A mensagem serve de identificação do doente que se está a conectar e o servidor envia para o cliente alguns parâmetros necessários para o processamento e aquisição dos sinais.
-
- Comunicação periódica: cliente-servidor
- Logs: cliente-servidor
- Alertas: servidor-cliente

- Feedback da comunicação

5.3.2 Sintaxe

5.3.2.1 *Comunicação inicial: cliente-servidor*

Estrutura:

—	—	—	—
1	2	3	4

1. **\ (char)** - inicio da mensagem
2. **(long)** – hora da comunicação
3. **(6 hex)** – mac address do dispositivo que pretende conectar com o servidor
4. **\ (char)** – fim da mensagem

Exemplo:

\ 1417162942424 0019B9FBE258 \

- Tipo de Mensagem "\ ... \": inicio de comunicação cliente-servidor
- Mensagem enviada no dia 28-11-2014 às 8h20

Mac Address "00:19:B9:FB:E2:58"

5.3.2.2 *Comunicação inicial: servidor-cliente*

Cada dispositivo configurado para o cliente irá originar uma mensagem diferente.

Estrutura:

—	—	—	—	—	—	—	—
1	2	3	4	5	6	7	8

1. **/ (char)** - inicio da mensagem
2. **(long)** – hora da comunicação
3. **(int)** – código do doente que se está a ligar
4. **(int)** – período de comunicação entre cliente e servidor (em segundos)
5. **(char)** – equipamento
6. **(6 hex)** – mac address do equipamento
7. **(char float)[]** – pares de parâmetro e valor referentes à configuração do equipamento
8. **/ (char)** – fim da mensagem

Exemplo:

/ 1417162942424 457 30 N 0088144D4CFB h30 o30 /

- Tipo de Mensagem "/ ... /": inicio de comunicação servidor-cliente
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- 30 – período de comunicação entre cliente e servidor (30 segundos)
- N - Nonin
- Mac Address "00:88:14:4D:4C:FB"
- h30 – HR com período de amostragem de 30seg
- o30 – SPO2 com período de amostragem de 30seg

5.3.2.3 *Comunicação periódica: cliente-servidor*

Estrutura:

— — — — —

1 2 3 4 5

1. **# (char)** - inicio da mensagem
2. **(long)** – data de registo dos parâmetros de actividade
3. **(int)** – código do doente
4. **(char[2] float)[]** – pares de parâmetro e valor referentes aos indicadores de actividade que os equipamentos registam. A primeira letra representa o dispositivo, e a segunda o parâmetro. Se a segunda letra estiver em letra minúscula serve de indicação ao servidor que o valor é pouco fiável. Por outro lado, caso seja uma letra maiúscula o valor é fiável.
5. **# (char)** – fim da mensagem

Exemplo:

1417162942424 457 Nh75 No98 C0 M1

- Tipo de Mensagem "# ... #": comunicação periódica cliente-servidor
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- Nh75 – HR com valor de 75 (pouco fiável)
- No30 – SPO2 com valor de 98 (pouco fiável)
- C0 – 0 Counts (fiável)
- M1 – 1 METS (fiável)

5.3.2.4 *Logs: cliente-servidor*

Estrutura:

— — — — — —

1 2 3 4 5 6

1. **? (char)** - inicio da mensagem
2. **(long)** – hora do log
3. **(int)** – código do doente
4. **(int)** – id do evento
5. **(char)** – equipamento
6. **? (char)** – fim da mensagem

Exemplo:

? 1417162942424 457 N 25 ?

- Tipo de Mensagem "? ... ?": logs cliente-servidor
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- N - Nonin
- 25 – id do evento

5.3.2.5 *Alertas: servidor-cliente*

Estrutura:

— — — — — —

1 2 3 4 5 6

1. **! (char)** - inicio da mensagem
2. **(long)** – hora do alerta
3. **(int)** – código do doente
4. **(int)** – número de caracteres da mensagem de alerta
5. **(char[])** – mensagem de alerta
6. **! (char)** – fim da mensagem

Exemplo:

! 1417162942424 457 20 Mensagem do Servidor !

- Tipo de Mensagem "! ... !": alerta servidor-cliente
- Mensagem enviada no dia 28-11-2014 às 8h20
- 457 – código do doente
- Mensagem do Servidor – mensagem de alerta

5.3.2.6 *Feedback da Comunicação*

Para cada tipo de mensagem enviada poderá existir um feedback do recetor, como por exemplo uma mensagem de *acknowledge*. Essa mensagem será composta por o caracter inicial e final de cada tipo de mensagem enunciada anteriormente, a hora da mensagem original, e um valor inteiro que representa a resposta.

Estrutura:

—	—	—	—
1	2	3	4

1. **(char)** - inicio da mensagem
2. **(long)** – hora da mensagem original
3. **(int)** – código do feedback
4. **(char)** – fim da mensagem

Exemplo:

1417162942424 0

- Acknowledge para a comunicação periódica
- Mensagem enviada no dia 28-11-2014 às 8h20
- 0 – Sucesso
- > 0 – Erros / Eventos inesperados

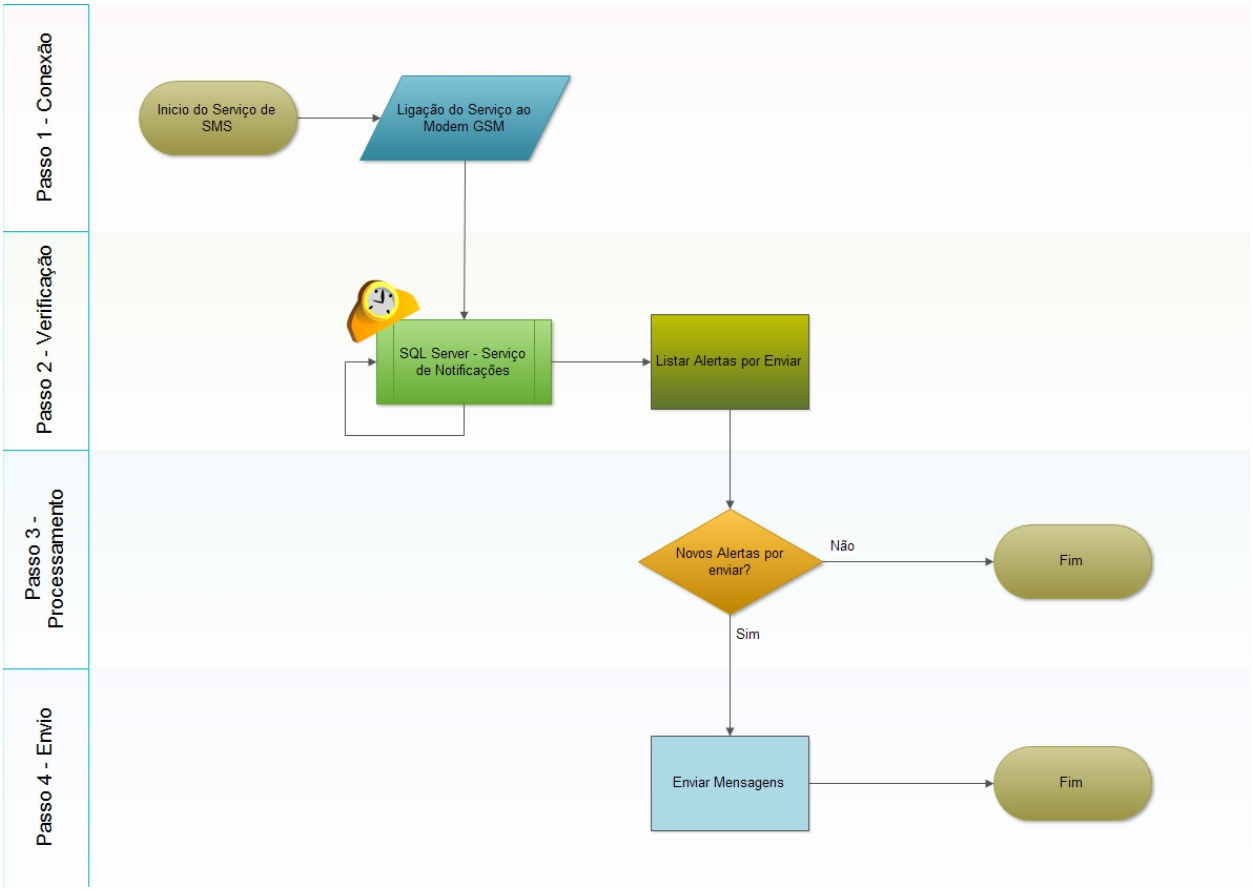
5.4 Configurações

- Porta

7 Serviço de SMS

7.1 Descrição

Um dos componentes do sistema TelemOLD é o sistema de registo e envio de alertas via SMS. Nesse sentido, o serviço de SMS é o componente passivo do sistema de alertas, cuja função, consiste essencialmente em, verificar alertas por enviar (Tabela DNT_Alerta_Por_Enviar) e efetuar o seu processamento e envio através de um modem GSM, para o número do médico de serviço. Desta forma, apenas é possível configurar um contacto telefónico que irá receber todos os alertas que surgirão.



7.2 Verificação de Alertas por Enviar

A verificação dos alertas por enviar utiliza um sistema de notificações incorporado no SQL Server que permite em tempo real a leitura e processamento de novos alertas, através de um *trigger* que é lançado quando à modificações na(s) tabela(s) que estamos a monitorizar.

7.3 Processamento dos Alertas

O processamento dos alertas é feito através de um serviço Windows.

7.4 Envio dos Alertas

O envio é feito pelo serviço Windows através do uso da componente da Cast (**Cast.Net**), para estabelecimento da comunicação com o *modem* GSM. Para um correcto funcionamento do serviço de

envio de SMS é necessário que o modem GSM esteja conectado ao computador onde se encontra o SGBD, uma vez que os alertas são gerados através de um sistema de notificações automaticamente criadas via *Sql Server Notification Services*, o que faz com que o serviço de SMS tenha de ser executado nessa mesma máquina de modo a que haja conectividade entre os *Notification Services* do Sql Server e o serviço de SMS.

7.5 Instalação

- Requisitos
- Instalação da aplicação
- Configuração

8 Trabalho Futuro

8.1 Sistema de Classificação

- Classificação do estado do doente: Dormir, Repouso, Movimento

8.2 Aplicação Web

- Migração do ORM do DNN para a EF

8.3 Serviços de SMS

- Configurações na Base de Dados
- Possibilidade de ter mais que um médico de serviço

8.4 Aplicação Móvel

- Sistema de feedback para o Doente (pergunta / resposta)
    - Perguntas programadas
    - Perguntas baseadas nos valores adquiridos

9 Anexos

- METS Compendium
- Paper COUNTs vs METS
- Específicações Nonin
- Fórmula METS