

Sistema embarcado para transferência de dados de sinais vitais usando o padrão HL7(Health Level 7)

Gabriel Muniz Cerqueira
Instituto Federal da Bahia,
Salvador, Brasil
Rua Emídio dos Santos, S/N,
Barbalho
munhosga@gmail.com

Raimundo Jorge Abreu
de Jesus
Coorientador
Instituto Federal da Bahia,
Salvador, Brasil
Rua Emídio dos Santos, S/N,
Barbalho
rjorgebr@hotmail.com

Antonio Carlos Santos
Souza
Orientador
Instituto Federal da Bahia,
Salvador, Brasil
Rua Emídio dos Santos, S/N,
Barbalho
acsantossouza@gmail.com

Resumo

Aplicações pervasivas na assistência à saúde são importantes e eficazes, apesar de demandar uma infra-estrutura especializada de hardware e software ainda assim continuam na onipresença no cotidiano das pessoas. Nesse contexto coletar e processar os dados do paciente remotamente se torna uma oportunidade para o surgimento de novas técnicas de monitoramento remoto de saúde. Tomando como base o uso de tecnologia embarcada, o presente trabalho visa apresentar por meio de uma plataforma embarcada capaz de realizar o monitoramento de sinais vitais de aparelhos distintos a possibilidade de codificar esses dados em um padrão internacionalmente conhecido, protocolo HL7(Health Level 7) que é utilizado pelo setor da saúde para permitir a troca de informações de saúde através de mensagens, entre aplicações distintas. Nesse cenário o sistema viabiliza a transmissão de dados para uma aplicação servidora ou mesmo monitores multiparamétricos, oferecendo assim segurança e usabilidade, assim como a transmissão de dados clínicos que possibilitará um monitoramento que permita avaliar os sinais vitais.

Palavras-chave - embarcado, sinais vitais, HL7 (Health Level 7), dados clínicos, monitoramento, hardware, software

Abstract

Pervasive applications in health care are important and effective, although they require a specialized hardware and software infrastructure yet remain ubiquitous in people's daily lives. In this context, collecting and processing patient data remotely becomes an opportunity for the emergence of new remote health monitoring techniques. Based on

the use of embedded technology, this paper aims to present, through an embedded platform capable of monitoring vital signs of different devices, the possibility of coding these data in an internationally known standard, HL7 (Health Level 7) which is used by the health sector to allow the exchange of health information through messages between different applications. In this scenario, the system enables the transmission of data to a server application or even multiparameter monitors, thus offering security and usability, as well as the transmission of clinical data that will allow a monitoring that allows the evaluation of vital signs.

Keywords - embedded ,vital signs, HL7 (Health Level 7), clinical data, monitoring, hardware, software

1. INTRODUÇÃO

A crescente demanda por serviços médicos estão em um ritmo exponencial com o crescimento populacional. A computação e suas subáreas evoluem e a cada dia mais dispositivos estão interconectados.

De acordo com (Araujo, 2003) a computação ubíqua beneficia-se dos avanços da computação móvel e da computação pervasiva. A computação móvel é definida como a competência de transportarmos os dispositivos computacionais de forma fácil e prática, possibilitando a presença e a utilização dos mesmos em qualquer lugar . A computação pervasiva descreve que os dispositivos computacionais estão agregados ao ambiente de forma imperceptível para o usuário. Dessa forma, se induz que o sistema computacional tem a capacidade de obter informações e dados relativos ao ambiente no qual ele está inserido [2].

É importante que um sistema consiga configurar, alterar, colher informações e transmiti-las de acordo com o ambiente que ele faz parte. Nas palavras de (Araujo, 2003) o computador tem a capacidade de obter informações do ambiente no qual ele está embarcado e utilizá-lo para dinamicamente construir modelos computacionais, ou seja, controlar, configurar e ajustar a aplicação para melhor atender as necessidades do dispositivo ou usuário. Desta interação surge a capacidade de computadores agirem de forma 'inteligente' no ambiente no qual nos movemos, um ambiente povoado por sensores e serviços computacionais.

Com base nos avanços tecnológicos abordados e relacionados, diversos aplicativos móveis, softwares e outros já fazem parte do nosso dia a dia e podem auxiliar na obtenção de uma vida mais saudável ou a descobrir, em um curto período de tempo, sintomas de possíveis doenças ou informações mais apuradas sobre o indivíduo e seu estado atual. Já temos aplicações que monitoram os batimentos cardíacos em tempo real, que medem o índice glicêmico, a perda de gordura e até a qualidade do sono. Nesse contexto obter dados clínicos, acompanhamento de resultados, entre outras situações, que não sejam hospitalares é preciso e a tecnologia que ofereça um suporte adequado para que possa atuar nessas e em outras limitações. Dessa forma sistemas embarcados ou mesmo de tempo real podem auxiliar em soluções inteligentes para problemas dispendiosos.

O primeiro grande sistema embarcado que se tem notícia é o AGC (*Apollo Guidance Computer*). Ele foi desenvolvido para propiciar, através de interfaces eletrônicas, poder computacional para a navegação e controle de uma espaçonave. Ele era um microcomputador de 16 bits, 15 de dados e 1 de paridade, que operava a 1,024 MHz e que possuía 2048 words de memória RAM e 36864 words de memória de programa [4].

Segundo observações de (Ganssle, 2003) um sistema embarcado (ou sistema embutido) é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla.

Sistemas como PDAs são geralmente considerados sistemas embarcados, pela natureza do material informático (hardware) que os compõem, são mais flexíveis em termos lógicos (software), para realizar uma tarefa específica. Por questões como segurança e usabilidade, alguns inclusive possuem restrições para computação em tempo real. O software escrito para sistemas embarcados é muitas vezes chamado firmware, e armazenado em uma memória ROM ou memória flash ao invés de um disco rígido. Por vezes é executado com recursos computacionais limitados: sem teclado, sem tela e com pouca memória, em alguns casos não tem como alterar a natureza de sua funcionalidade, sendo necessário, uma nova reprogramação, algumas vezes somente na própria fábrica, são características de periféricos, aparelhos ou placas de expansão que enviam ou recebem informações do computador. Segundo (Laplante, 2018) na informática, o termo 'periférico' aplica-se a qualquer equipamento ou acessório que seja ligado à CPU (*Unidade Central de Processamento*), ou, em um sentido mais amplo, ao computador.

Por essa simplicidade de uso da ferramenta e aplicabilidade desses artefatos tecnológicos podemos direcioná-lo para área da saúde, monitoramento remoto e transferência de dados. A transferência de dados clínicos para monitores como exemplo os monitores multiparamétricos requer uma configuração específica, sendo assim configurações que possam ser adaptadas aos mais diversos ambientes são necessárias, visto o exemplo de monitores do setor de triagem até Unidades de Terapia Intensiva (UTI) de alta complexidade e Centros Cirúrgicos como, por exemplo, as tela de monitoramento multiparamétrico dos seguintes modelos T7 ou P12 que podem utilizar o protocolo *Health Level-7* ou *HL7* para a transmissão de dados dos sinais vitais. [9] O protocolo HL7 funciona como uma ligação entre diversas e distintas aplicações como sistemas de registro do paciente: **EMR** /

EHR através de um protocolo padrão de mensagens, com auxílio do protocolo *TCP/IP*.

Logo o uso de uma plataforma de baixo custo para decodificar e codificar o envio de dados específicos no padrão que possa ser interpretado clinicamente é uma solução viável. Uma arquitetura que aperfeiçoe a coleta de dados clínicos, já que hospitais e outros provedores de atenção e cuidados à saúde geralmente têm sistemas diferentes para diferentes aspectos dos serviços, oximetria de pulso arterial (OPA)¹, pressão arterial, temperatura e outros. O que nesse universo de possibilidades torna-se na maioria dos casos difícil se comunicarem uns com os outros, mesmo estando na mesma rede e possibilitar essa leitura e monitoramento de forma mais ampla ou mesmo padronizada desses dados que são coletados separadamente, além da dificuldade da transferência de dados clínicos essenciais para efetuar uma parecer do estado clínico do indivíduo.

Tendo em vista esse cenário, surge a possibilidade de estabelecer uma codificação padrão que possa englobar dados sensoriais de diferentes dispositivos e organizá-los no padrão do protocolo HL7 que, permite compartilhamento e recuperação de informações das informações de natureza clínica. O uso de um microcontrolador que no caso corrente foi escolhido o RaspBerry Pi 3 por ser uma ferramenta robusta e portátil o que torna viável para servir de proxy embarcado ou mesmo estabelecer o conceito de *middleware*² para que os dados a sejam trafegados, oferecendo uma rápida e prática prototipagem, permitindo assim uso do protocolo TCP/IP assim como a tecnologia bluetooth de baixa energia.

O trabalho está organizado em seções onde logo após esta introdução apresenta-se o referencial bibliográfico, posteriormente apresenta trabalhos correlatos, as soluções desenvolvidas, e por fim mostra os resultados obtidos, assim como as conclusões, referências e anexos.

2. REFERENCIAL TEÓRICO

Essa seção apresenta as principais referências usadas para compreensão do problema e desenvolvimento de uma solução que possa abranger a proposta. Ela está organizada da seguinte maneira: A subseção **2.1** traz um breve levantamento sobre o monitoramento dos sinais vitais, já a subseção **2.2** apresenta uma breve explanação sobre sistemas embarcados, **2.3** apresenta brevemente o conceito *middleware*, a subseção **2.4** aborda a ferramenta RaspBerry Pi, a subseção **2.5** apresenta conceitos referente ao protocolo *Health Level-7* ou *HL7* e sua importância para a área da saúde, **2.6** expõe os conceitos relativos ao protocolo *TCP/IP*, além do esclarecimento quanto ao socket e a tecnologia Bluetooth.

¹Oximetria de pulso arterial (OPA), fornece informações de relevância clínica sobre a saturação de oxigênio carregado pelas hemoglobinas presentes no sangue arterial e permite analisar a amplitude e a frequência de pulso, tanto na fase de repouso como de atividade.

²É o software de computador que fornece serviços para softwares aplicativos além daqueles disponíveis pelo sistema operacional.

2.1 Monitoramento de Sinais Vitais

Segundo (Murta, 2006) os indicadores do estado de saúde de um paciente são temperatura, pulso, respiração e pressão arterial, esses indicadores permanecem mais ou menos constantes e sua variação pode indicar alguma enfermidade. Os indicadores do funcionamento fisiológico básico, ou seja, do estado de equilíbrio térmico, circulatório e respiratório abrangendo assim pressão arterial; pulso, respiração e temperatura são de grande importância e caracterizam os sinais vitais. Ainda, o monitoramento pode gerar estatísticas que quando analisadas podem detectar alguns tipos de anomalias, em razão do grau de importância e por seus parâmetros serem regulados por órgãos vitais, revelando o funcionamento deles.

Nesse contexto (Murta, 2006) discorre que a variação dos valores pode indicar problemas relacionados com insuficiência ou excesso de consumo de oxigênio, depleção sanguínea, desequilíbrio eletrolítico, invasão bacteriana, entre outros. O desenvolvimento de aplicações que impactem o mínimo possível na vida dos pacientes é essencial e se torna um abordagem viável com as tecnologias desenvolvidas que transmitem e processam os sinais vitais à distância. Nessas circunstâncias o monitoramento à distância de pacientes permite a detecção precoce de alterações nas condições de saúde dos pacientes e viabiliza a intervenção mais precoce da terapia, com bons resultados para diversos tratamentos e potencial impacto positivo em desfechos clínicos apresentados pelos pacientes [26].

A comunicação wireless, ou mesmo via Bluetooth e as pequenas dimensões dos dispositivos eletrônicos permitem maior mobilidade e menores restrições ao paciente, os quais em conjunto com a computação vestível, viabilizam uma forma quase invisível para o monitoramento de pacientes. Um desafio na melhoria dos serviços prestados nestes ambientes está na disponibilização das informações geradas pelos equipamentos médicos sem prejuízo da mobilidade e sem implicar necessariamente em estresse devido ao aumento das fontes de informações.

Nos últimos anos estão sendo desenvolvidos sistemas que permitem o monitoramento de sinais vitais de pacientes, trazendo mais qualidade de vida e melhores resultados no tratamento de saúde [26]. Nessas circunstâncias as pesquisas dedicadas a este assunto indicam que teremos evoluções significativas tanto quanto a eficiência, efetividade da qualidade dos processos clínicos e assistenciais.

2.2 Microcontroladores

Microcontrolador é um pequeno computador (SoC)³ num único circuito integrado o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída [3]. A memória de programação pode ser RAM, NOR flash ou PROM a qual, muitas vezes, é incluída no chip.

³System-on-a-chip (SoC), System On Chip (SOC) ou, em português, sistema-em-um-chip, se refere a todos os componentes de um computador, ou qualquer outro sistema eletrônico, em um circuito integrado (chip).

Os microcontroladores surgiram como uma evolução natural dos circuitos digitais devido ao aumento da complexidade dos mesmos. Chega um ponto em que é mais simples, mais barato e mais compacto, substituir a lógica das portas digitais por um conjunto de processador e software. São concebidos para aplicações embarcadas, em contraste com os microprocessadores utilizados em computadores pessoais ou outras aplicações de uso geral, sendo um tipo especial de circuito integrado, pois vem com a possibilidade de ser programado para desempenhar tarefas específicas.

Alguns kits de desenvolvimento possuem diversos hardwares, para se usar com o microcontrolador, com display de LED, botões, luzes, acionador de motor, sensores e uma gigantesca infinidade [6]. De forma geral, um microcontrolador pode ser entendido como um computador simples construído em um único chip, que pode ser programado e integra processador (Unidade Lógica Aritmética - ULA), memória, portas de I/O, dispositivos de comunicação serial dentre outros [12]. Segue abaixo os componentes básicos do microcontrolador.

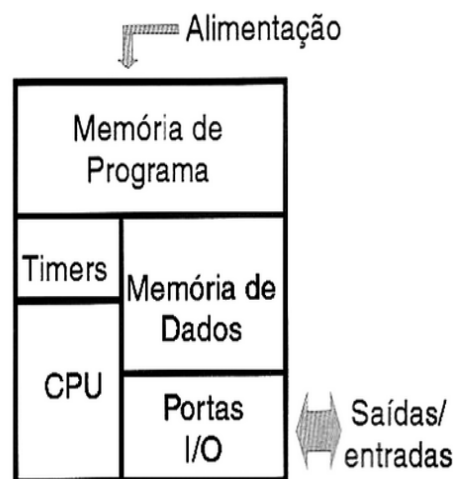


Figura 1: Componentes do microcontrolador.

O seu consumo de energia é relativamente baixo, na casa dos miliwatts e possui habilidade para entrar em modo de espera (Sleep ou Wait), aguardando por uma interrupção ou evento externo, como, por exemplo, o acionamento de uma tecla, ou um sinal que chega via uma interface de dados. Eles permitiram a evolução de equipamentos que há anos não evoluíam, como os motores a combustão, máquinas fotográficas, que migraram de processos químico/mecânico a circuitos com microcontroladores mais sensores digitais e memória. Existindo então algumas ferramentas que criam uma ponte de transferência de dados entre o aparelho utilizado e o microcontrolador [1A]. Logo, mesmo com poucas ideias e instruções em programação, foi possível criar diversas tecnologias, como aplicativos, sistemas operacionais, dentro do monitor, disco rígido, relógio de pulso, rádio relógio, máquinas de lavar, forno de micro-ondas, telefone, dentre outras coisas. Não há muita diferença da programação comum, de software, e o uso de microcontroladores pois você terá algumas modificações a nível de hardwares, instruções e opções de iram variar de acordo com o hardware em uso.

2.2.1 Sistemas embarcados

Um sistema embarcado (sistema embutido, microcontrolador ou controlador programável) é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado a um dispositivo ou sistema que ele controla. Embarcados são desenvolvidos para uma tarefa específica. Por questões como segurança e usabilidade, alguns inclusive possuem restrições para computação em tempo real. Segundo (Morais, 2012) são sistemas de processamento digital, cujo elemento central é um microcontrolador, que possuem seu firmware/software de controle armazenado em uma memória interna ou externa ao microcontrolador.

São dedicados a uma tarefa específica e interagem com o ambiente por meio de sensores e atuadores. Sendo assim, podem detectar variações paramétricas e ambientais e, após o processamento adequado destas informações, reagirem a essas perturbações, transmitindo-as para uma ferramenta que possa interpretar, mensurar e decodificar, nesse contexto podem ser utilizados em aviões, carros, telefones celulares, calculadoras, eletrodomésticos, equipamentos médicos, etc, ou seja, na maioria dos equipamentos eletrônicos atuais. Nesse contexto observe abaixo as características gerais do uso de sistemas embarcados.

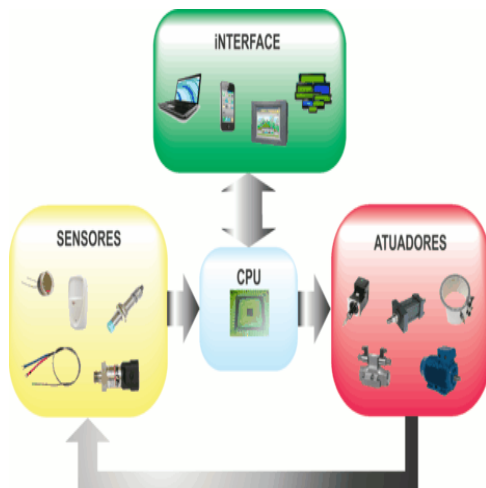


Figura 2: Desenvolvimento de Sistemas Embarcados.

2.2.2 ESP-WROOM32

O ESP32 é uma série de microcontroladores de baixo custo e baixo consumo de energia. Também é um sistema-em-um-chip com microcontrolador integrado, Wi-Fi e Bluetooth. A série ESP32 emprega um microprocessador Tensilica Xtensa LX6 com duas variações dual-core e single-core e inclui uma antena integrada, RF tipo balun, amplificador de potência, receptor de baixo ruído amplificado, filtros, gerenciamento de energia dos módulos [36]. Observe o módulo na figura abaixo.



Figura 3: Módulo ESP-WROOM32

O ESP32 reúne um apanhado de recursos que tornam seu uso em Internet das Coisas algo bem interessante, dado que agora a presença de mais periféricos permite a sua integração com mais dispositivos e componentes diversos. Dentre as interfaces de comunicação, ele possui suporte a SPI, UART e I2C (protocolos relativamente comuns), como também tem suporte a Infravermelho (IR) e SDIO (para interface com cartão de memória) [37].

Características do ESP-WROOM 32:

- Processador principal: LX6 32-bit Dual-core, operando 2-240 MHz.
- Processador secundário: ULP de 8 MHz.
- FLASH: 4 MB e RAM: 520 kB.
- GPIO: 34, com 3.3V e 12mA.
- ADC: 18, com resolução de 12-bit.
- DAC: 2, com resolução 8-bit.
- WiFi: 2,4 GHz, 802.11 b/g/n.
- Bluetooth: Bluetooth Low Energy v4.2 (BLE).
- Acelerador via hardware, encriptação, hash e afins.
- True Random Number Generator (TRGN).
- 4 Timers de 64-bit.
- 4 Watchdogs.
- 10 Sensores de Touch Capacitivo.
- 1 Sensor de temperatura interno e 1 efeito Hall.

2.2.3 Raspberry Pi

Apesar de não se enquadrar especificamente como um microcontrolador, o Raspberry Pi deve ser levado em consideração quando o assunto é automação e monitoramento pois o potencial do pequeno equipamento para prototipação é rápida e estável.

O Raspberry Pi (ou RPi) é um microcomputador, com seus componentes em uma única placa lógica. Há o processador, a memória RAM e a placa de vídeo impressos, e entradas USB, HDMI, áudio e vídeo composto, para câmera e telas LCD e uma GPIO, com pinos I/O de múltiplo propósito. A alimentação é feita através de uma porta microUSB, que permite usar fontes de energia de telefones celulares [14].

O Raspberry pi começou a ser desenvolvido em 2006 pela Fundação Raspberry Pi no Reino Unido como intuito de despertar o interesse pela computação através de um computador simplificado, flexível e que pudesse ser utilizado como uma ferramenta educacional [8]. Nesse âmbito, previa-se sua utilização apenas para fins educacionais, porém como o interesse pela placa foi tanta acabou se abrindo um infinito mundo de possibilidades, com uma gama imensa de projetos e aplicações, pode ser utilizado como um computador normal ou um computador para programação, seja em Python, C/C++, Java ou até mesmo Assembly.

O Raspberry Pi 3 é a terceira geração do famoso miniPC a placa é um computador barato, portátil e versátil, usado principalmente em projetos de programação, robótica e em iniciativas em geral com software e hardware livre.

Algumas utilidades do dia a dia incluem criação de servidor de arquivos pessoal e videogame retrô. O mesmo roda o sistema operacional Ubuntu, Raspbian e outras distribuições do Linux. Além disso, é compatível com o Windows 10 IoT, versão do software da Microsoft feita para automação doméstica e outras aplicações envolvendo Internet das Coisas. Abaixo se encontra um figura da Raspberry Pi 3.



Figura 4: Raspberry Pi.

2.3 Middleware

O Middleware ou mediador, no campo de computação distribuída, é um programa de computador que faz a mediação entre outros softwares. É utilizado para mover informações entre programas ocultando ou mesmo abstraindo do programador diferenças de protocolos de comunicação, plataformas e dependências do sistema operacional [27].

É geralmente constituído por módulos que proporcionam a sua integração com aplicações desenvolvidas em diversas linguagens de programação e interfaces de baixo nível que permitem a sua independência relativamente entre dispositivos. Com o objetivo de mascarar a heterogeneidade e fornecer um modelo de programação mais produtivo sendo composto por um conjunto de processos ou objetos em um grupo de computadores, que interagem entre si de forma a implementar comunicação e oferecer suporte para compartilhamento de recursos [28].

O Middleware é a designação genérica utilizada para referir os sistemas de software que se executam entre as aplicações e os sistemas operativos. O objetivo do middleware é facilitar o desenvolvimento de aplicações, tipicamente aplicações distribuídas, assim como facilitar a integração de sistemas legados ou desenvolvidos de forma não integrada [28].

2.4 Padrão HL7 - Health Level 7

O HL7 (Health Level 7) é um "padrão" utilizado pelo setor de saúde para permitir a troca de informações em saúde através de mensagens, entre aplicações, como LIS para EHR, HIS para RIS e equipamentos médicos, por exemplo.

É desenvolvido, gerenciado e mantido pela *Health Level Seven International* (HL7), que é uma organização sem fins lucrativos, certificada pela ANSI, com sede em Ann Arbor, EUA, fundada em 1987 pelo Dr. Ed Hammond, Duke University [9]. A primeira versão do HL7 surgiu em 1987 e não passou de uma apresentação de algo a ser implementado, uma proposta de protocolo.

Efetivamente não foi utilizada por alguma instituição médica. Seu intuito era apenas apresentar um protótipo de um padrão de intercomunicação, já segunda versão do HL7 surgiu um ano depois, em 1988. Esta já se apresenta como um protocolo a ser seguido pela comunidade de profissionais ligados à área de Informática para Medicina [31].

A partir do release 2.3, lançado em 1998, passa a ser adotada pelos fabricantes de equipamentos e sistemas hospitalares no mundo e até hoje, mesmo após os novos releases, continua como a versão mais utilizada pelos fabricantes.

Segundo a análise de Rades [22] o padrão HL7 resolve esse problema, fornecendo estrutura para a troca, integração, compartilhamento e recuperação de informações de registros eletrônicos em saúde. As interfaces HL7 proporcionam de forma criptografada e por meio seguro, a transferência de dados entre sistemas.

O padrão ou mesmo protocolo HL7 é muitas vezes chamado de "padrão fora do padrão", e isso não é muito justo, mas reflete o fato de que quase todos os hospitais, clínicas, centros de imagem, laboratórios e unidades de saúde são "especiais" em termos de como implementam o HL7. A principal razão é porque não existe um formato ou um processo clínico padrão para interagir com pacientes, dados clínicos ou com as equipes envolvida sem sua atenção. A análise comercial descreve bem a usabilidade do protocolo, observe a figura abaixo suas características.

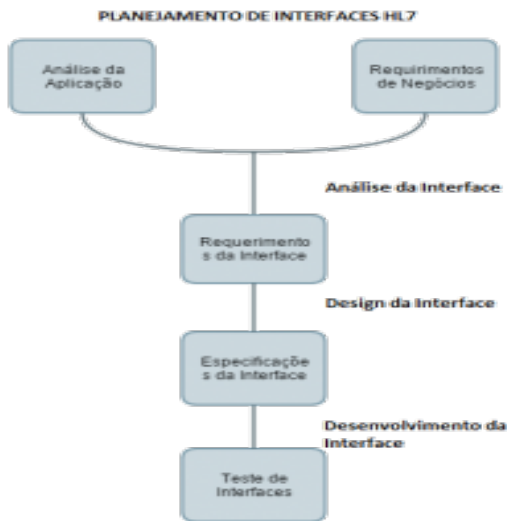


Figura 5: Análise Comercial HL7.

É visto que há um planejamento quanto ao padrão HL7, a identificação e revisão dos requisitos de negócio globais do projeto e o papel que as Interfaces HL7 irão desempenhar para a realização dos requisitos assim como as atividades integradas usando este padrão são de grande importância para alcançar os objetivos do projeto.

As especificações da interface HL7 descreve como os requisitos HL7 serão implementados nos componentes de software, assim como o próprio planejamento de testes, pois envolvem testes reais da interface, testes unitários e testes de integração do HL7, como é mostrada na figura abaixo.

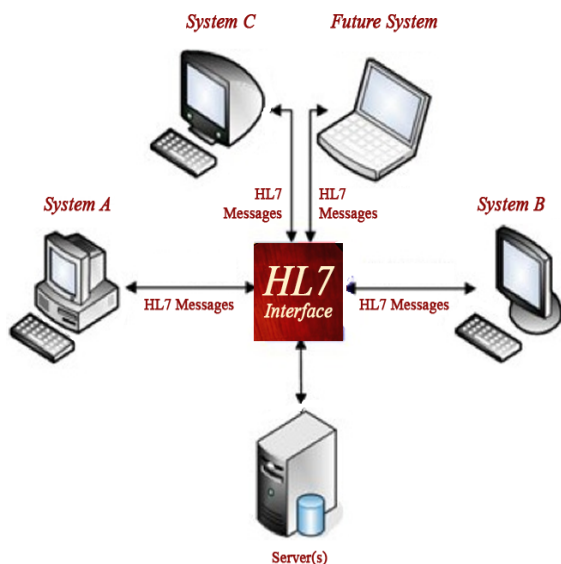


Figura 6: Interfaces HL7.

Com o uso de um motor de interface HL7, os provedores de saúde podem usufruir dos benefícios dos sistemas de informação existentes sem grandes reinvestimentos em novas tecnologias, reduzindo os custos e aumentando a vida e eficiências dos sistemas atuais [22].

É possível também conectar-se a sistemas externos a sua unidade de atenção à saúde, como prestadores de serviços terceirizados, como radiologia e laboratórios.

Na abordagem de (Rades, 2015) o uso de um motor de interface HL7, os provedores de saúde podem usufruir dos benefícios dos sistemas de informação existentes sem grandes investimentos em novas tecnologias, reduzindo os custos e aumentando a vida e eficiências dos sistemas atuais.

Nesse contexto essa transferência tradicional de arquivos pode ocorrer por um soquete TCP / IP, ou pode ser transmitido em tempo real como em lotes (batch) [9]. O *transmission control protocol/internet protocol* TCP/IP é um padrão de comunicação que reúne um conjunto de protocolos tais como tcp, ip, ftp (*file transfer protocol*), telnet, icmp, arp e nfs [24].

O TCP/IP foi desenvolvido em 1969 pelo U.S. Department of Defense Advanced Research Projects Agency, como um recurso para um projeto experimental chamado de ARPANET (Advanced Research Project Agency Network) para preencher a necessidade de comunicação entre um grande número de sistemas de computadores e várias organizações militares dispersas. O objetivo do projeto era disponibilizar links (vínculos) de comunicação com alta velocidade, utilizando redes de comutação de pacotes [23]. As informações que trafegam na rede necessitam do TCP/IP, por isso ele é utilizado como protocolo primário da rede na internet. Este protocolo foi dividido em "camadas" bem definidas, cada uma realizando sua parte na tarefa de comunicação (aplicação, transporte, rede, enlace/físico).

É possível estabelecer conexões a sistemas externos a sua unidade de atenção à saúde, como prestadores de serviços terceirizados, como radiologia, laboratórios, farmácias e outros, assim como o uso de serviços *Home Care*, os custos para troca de informações entre essas unidades são reduzidos porque o HL7 é uma interface padrão, utilizada pela indústria da saúde.

Ainda assim, devido à grande solidificação de sua gramática e grande aceitação entre as empresas e os desenvolvedores, o HL7 versão 2.X continua sendo amplamente utilizado e ainda encontra-se em pleno desenvolvimento.

Como a estrutura de uma mensagem HL7 não é algo totalmente fixo, existe liberdade durante a criação de mensagens, e criar uma estrutura que permita guardar a informação de uma mensagem de forma estruturada, sem perder a informação. Cada tipo de dado tem uma forma específica de codificação assim como as condições de repetição. Os campos de informação que compõe uma mensagem HL7 são agrupados segundo uma estrutura específica designada de segmento. Cada segmento individualmente inicia-se com um código de três caracteres que define a sua função. Assim, vários segmentos, divididos pelos respectivos separadores (normalmente <cr>), compõe uma mensagem HL7 [38].

Em paralelo à versão 2, surge a versão 3 do HL7, criada pela norma ISO/HL7 21731. Esta versão consiste de uma nova metodologia de interface, baseada nos princípios da orientação a objetos, compreendendo não só os tipos de dados de uma classe, mais a relação entre as mesmas.

2.5 Bluetooth

A tecnologia Bluetooth foi desenvolvida a partir do ano de 1998 através da parceria entre a Ericsson, Intel, Toshiba, Nokia e IBM com o objetivo de especificar um padrão mundial aberto para a conexão sem fio entre dispositivos de telecomunicações e de computação. Segundo (Araujo, 2004), a comunicação é onidirecional, suporta transmissões síncronas e assíncronas, aceita taxas de transferência de dados de até 1 Mbps e possui um alcance de 10 m.

O sistema utiliza uma frequência de rádio de onda curta (2.4 GHz) para criar uma comunicação entre aparelhos habilitados. Como seu alcance é curto e só permite a comunicação entre dispositivos próximos, seu consumo de energia é bem baixo [17].

2.5.1 Bluetooth BLE ou Smart

O BLE ou Bluetooth Smart é uma tecnologia de rede de área pessoal sem fios concebida e comercializada pela *Bluetooth Special Interest Group*. Em comparação com Bluetooth Clássico, o Bluetooth Smart destina-se a fornecer consumo de energia e custos consideravelmente reduzidos, procurando manter um alcance de comunicação similar [20]. Segue a figura que exemplifica a comunicação com os dispositivos.

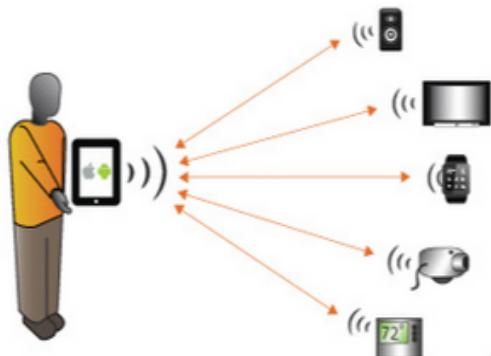


Figura 7: Comunicação de dispositivos com BLE.

Por sua característica principal ser a *economia de energia*, um dispositivo BLE permanece em modo IDLE (sleep) durante a maior parte do tempo. Foi desenvolvida para aplicações que apenas precisam enviar poucas informações, é vital para aplicações que precisam trocar apenas pequenas quantidades de dados periodicamente saindo do modo "sleep" apenas para realizar conexões que duram apenas milissegundos [20]. Isso ocorre pois ele foi projetado para aplicações que apenas precisam enviar poucas informações e esporadicamente.

Quando é realizada uma comparação entre as duas tecnologias fica claro que Bluetooth e Bluetooth Smart são utilizados para propostas diferentes, observe abaixo.

	Bluetooth Clássico	Bluetooth Low Energy (BLE)
Padrão de camada física	IEEE 802.15.1	GFSK
Frequências	2.4 GHz	2.4 GHz
Máxima Taxa de Bits (Mbps)	1 até 3	1
Distância Máxima (m)	10-100	50
Consumo de Energia	Alto	Muito Baixo
Vida Útil da Bateria	Dias	Meses a anos
Tamanho da Rede (dispositivos)	7	Indefinido

Figura 8: Comparação entre os Padrões.

	Bluetooth Clássico	Bluetooth Smart (BLE)
Preço	Alto	Muito Baixo
Consumo de Energia	Alto	Muito baixo
Aplicação	Transferências com grandes quantidades de dados	Transferências de pequenas quantidades de dados

Figura 9: Comparação de Tecnologias.

A existência de um dispositivo com essas características abre grandes possibilidades, sensores para informar pressão sanguínea ou o pulso de um indivíduo, comunicação máquina-a-máquina (M2M) e automatizar tarefas em uma residência doméstica (acender a luz quando alguém entrar no cômodo) ou no ambiente de trabalho torna-se imprescindível.

2.6 Qt Creator IDE

O *Qt Creator* é um poderoso ambiente de desenvolvimento integrado (IDE), que permite que o usuário possa desenvolver aplicativos e outros recursos para plataformas múltiplas, oferecendo muita facilidade e robustez para o seu código [40]. Permite que, através das bibliotecas QT GUI e plugins, você possa programar em diversas linguagens como *C*, *C++*, *Java*, *Ruby* e *Python*.

Além da possibilidade de códigos em diversas linguagens, o QtCreator também oferece ferramentas interessantes como o recurso de autocompletar, o que pode acelerar a velocidade em que você programa [40]. Haavard Nord e Eirik

Chambe-Eng (os desenvolvedores originais da Qt e o CEO e Presidente, respectivamente da Trolltech) começaram o desenvolvimento da "Qt" em 1991 [43].

As primeiras versões contavam apenas com duas edições: Qt/X11 for Unix e Qt/Windows for Windows. A versão para Windows só estava disponível em licença proprietária, significando que aplicações escritas para o Qt for X11 não poderiam ser portadas para Windows sem a compra de uma licença comercial. No fim de 2001, a Trolltech lançou a Qt 3.0, adicionando suporte para o Mac OS X. O suporte para o Mac OS X também estava disponível apenas com licença proprietária até Junho de 2003, quando a Trolltech lançou a Qt 3.2 com suporte ao Mac OS X disponível através de licença GPL, e em Junho de 2005, a Trolltech lançou a Qt 4.0 [43].

Ele também possui o QT Designs. É aqui que o nome "Creator" passa a fazer mais sentido. O QT Design permite você ir além do código e permite que você crie não apenas interfaces completamente visuais, interativas e chamativas, como também permite a criação de animações e outros recursos gráficos [40]. Essa ferramenta é fundamental para quem pretende criar aplicativos ou jogos para qualquer plataforma, principalmente para plataformas móveis, observe a figura com a página de apresentação da IDE, logo abaixo.

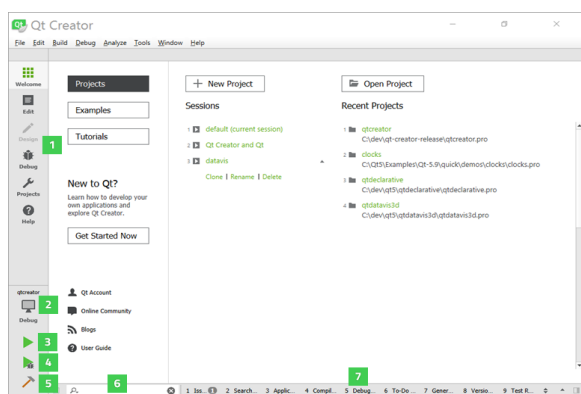


Figura 10: Home Page QTCreator.

Por ser multiplataforma, o QT Creator permite que o programador crie aplicações para Linux, Windows, Android e até mesmo para sistemas operacionais da Apple, tudo isso sempre precisando ficar alternando entre programas e compiladores [43].

O ambiente integrado do Qt Creator é sem dúvidas um dos mais importantes entre os atuais do mercado, não à toa que ele é usado por grandes empresas como Microsoft, AMD, Sky, Ubuntu, em diversas Smart TVs da LG e também faz parte da história de muitos aplicativos de sucesso [43].

Editor de código avançado

Qt Creator é focado em fornecer recursos que ajudem novos usuários Qt a iniciar o aprendizado e começar a desenvolver mais rápido e incrementem a produtividade dos desenvolvedores Qt experientes [44].

- Editor com suporte a C+, QML e ECMAScript;
- Ferramentas de navegação rápida pelo código;
- Realçamento de sintaxe e auto-complemento de código;

• Checagem estática de código e dicas de estilo enquanto digita;

- Suporte a refatoração de código;

Depurador visual

O depurador visual para C+ conhece a estrutura de muitas classes Qt, aumentando sua habilidade de exibir os dados do Qt claramente. Adicionalmente, o Qt Creator exibe os dados brutos do GDB de modo limpo e conciso [44].

- Interrompa a execução do programa;
- Ande através do programa linha por linha ou instrução por instrução;
- Defina pontos de parada;
- Examine o conteúdo da pilha de chamada, observadores e variáveis locais e globais;

3. TRABALHOS CORRELATOS

Nesta seção serão apresentados alguns trabalhos acadêmicos que abordam temas similares ao trabalho proposto, utilizam técnicas computacionais na solução dos problemas apresentados, utilizando uma ferramenta que podem alcançar áreas distintas de aplicabilidade. Foram analisados cinco projetos envolvendo o padrão HL7, tecnologias embarcadas e uso de aplicações terceiras.

O trabalho abordado por *Bernardo et al, 2015* os mesmos apresentam um sistema tecnológico atual que possa contribuir para a supervisão da qualidade de vida das pessoas, em especial idosos, permitindo que eventuais problemas possam ser detectados automaticamente, nesse contexto o uso das ferramentas propostas permeiam o gerenciamento de uma placa de desenvolvimento Intel Galileo que em caso de detecção de riscos como queda ou vazamentos de gás envia uma mensagem para o celular do responsável. Neste trabalho é apresentado as características dos sistemas propostos bem como suas vantagens e limitações. Para a proposta, a placa Intel Galileo acentuada limitação de recursos como processamento e memória RAM para aplicações em visão computacional, mas mostrou-se como uma solução bastante versátil, sendo possível encontrar soluções que não necessitem de alto poder de processamento e memória [29].

Ainda nos trabalho que apresentam a estrutura inicial deste projeto a técnica de controle para sistemas embarcados aplicado no monitoramento remoto de sinais vitais trabalhado por *Pereira et al, 2014*, demonstra o monitoramento remoto em tempo real dos sinais vitais, batimento cardíaco mais precisamente. Os microcontroladores são essenciais nesse projeto pois os dados coletados são comparados com os que estão parametrizados no microcontrolador,

assim como estes dados possuem um limiar mínimo e máximo estipulados no microcontrolador o que define o risco e envio de SMS, ou seja o microcontrolador funciona como interpretados de dados para atuar sinalizando uma interface ou mesmo um atuador como foi citado na *textbfigura 1*. Nesse contexto há a possibilidade de fazer a transmissão de dados diretamente utilizando o GPRS, ou seja, o médico manda os dados ao módulo do protótipo por SMS contendo a parametrização do paciente, assim o Arduino interpreta a mensagem recebida e parametriza esses dados no código.

Continuando no universo dos monitoramento correlacionados a saúde o projeto do sistema de monitoramento de pacientes em internação domiciliar, tem como foco o uso da tecnologia de computação pervasiva, através de sensores utilizados no ambiente domiciliar, o paciente pode ser monitorado a todo momento e em qualquer parte da casa.

Carvalho et al, 2011, dados fisiológicos similares a pressão arterial, frequência cardíaca, o que se caracteriza uma vertente pela qual esse projeto permeia, pois a frequência cardíaca pode ser um dos parâmetros a serem codificados e transmitidos via protocolo HL7. As atividades realizadas pelo paciente (se está caminhando, dormindo, comendo, etc.) e condições do ambiente (temperatura e umidade) podem ser obtidos continuamente. A similaridade com este projeto ainda se baseia pela leitura dos dados de sinais vitais, o que auxilia a fundamentação mas demonstra desvantagens pois abrange conceitos que diferem ao abordar plano de cuidados, o qual é composto de uma série de prescrições elaboradas pelo profissional de saúde, tais como quais medicações o paciente deve realizar e com qual periodicidade, quais medicamentos e quando devem ser tomados, recomendações de dieta e de exercícios físicos.

Os trabalhos seguintes já utilizam o protocolo, ou mesmo padrão HL7 no contexto da aplicação. O projeto de *Pereira, 2008*, referente a implantação de um parser HL7 para integração com aplicações terceiras, o foco do projeto correspondeu a desenvolver uma aplicação que permita a validação e interpretação de mensagens HL7 na versão 2.

Este projeto segue uma sequência plausível ao analisar algumas ferramentas existentes, as possíveis abordagens identificadas com o levantamento do estudo do HL7, assim como sua arquitetura e detalhes de implementação do projeto e de algumas ferramentas extra, ou seja, é um levantamento da aplicabilidade e as potenciais utilizações no padrão HL7 em software e aplicações. É extremamente abrangente o projeto, permeia várias tecnologias como : Java, Unit, XML(EXTensible Markup Language), JavaScript dentre outras. Diferencia-se deste projeto pois é mais específico minha abordagem, irei ler dados de dispositivos distintos e convergir para o padrão HL7 para que dessa forma possa ser aproveitado e apresentado posteriormente.

O trabalho de *Santos, 2014*, é uma monografia cujo analisa as principais características do protocolo HL7, focando em sua versão 3 na forma da implementação documental CDA(Clinical Document Architecture) ⁴. Aborda

⁴é um padrão de implementação do HL7 v3 que estabelece a troca de informações HL7 entre sistemas na forma de arquivos XML de compreensível leitura, para Documento Clínico.

em conjunto com o MongoDB, pretende-se demonstrar as vantagens da integração entre bancos não-relacionais e o padrão HL7, com o objetivo de otimizar a interoperabilidade e persistência de dados que representam resultados de exames clínicos, nesse contexto pode oferecer uma resposta ágil e flexível ao problema de consulta e armazenamento de resultados de exames clínicos, atendendo de forma otimizada às demandas da área de saúde quanto aos sistemas de informação.

Tendo em vista o uso do padrão e como foi aplicado a esse contexto o projeto mencionado foi direcionado em sua grande parte ao uso do protocolo e o banco de dados, o que sugere um leque de possibilidade para trabalhos futuros envolvendo uma aplicação completa, utilizando assim várias tecnologias em diferentes camadas do desenvolvimento do software, além de se apoiarem no suporte oferecido pelo padrão HL7.

4. SOLUÇÃO DESENVOLVIDA

Com base no que foi apresentado, a primeira implantação que é o cenário atual que a aplicação se encontra, está configurada com a ESP-32, um microcontrolador que desempenha um papel muito discreto, limitados em alguns serviços, os riscos pelo fato de hoje não comportar as necessidades que o projeto precisa e que são iminentes, a instabilidade na qual se encontra em ler os dados coletados via bluetooth e manter as trocas de informações continua muitas vezes é dificultoso, super aquece, além de ocasionar desligamento das funções por estresses no ambiente em questão o que hoje ainda encontramos detalhes para aperfeiçoamento, segue abaixo um figura do diagrama que exemplifica o cenário.

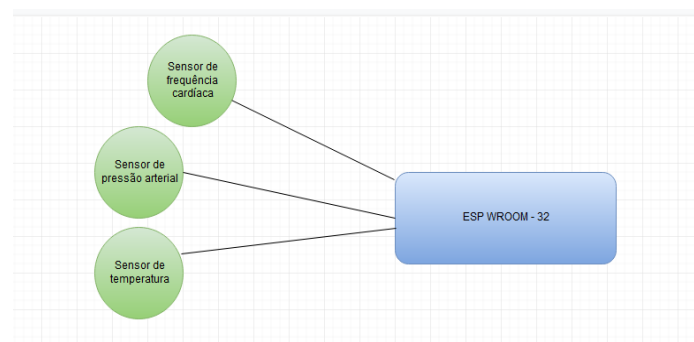


Figura 11: Cenário ESP-32.

Partindo dessa conjuntura, a modificação será fundamental para um microcontrolador que possa fazer um papel mais importante, que possa oferecer serviços mais elaborados que os serviços prestados e ainda assim continue sendo um processamento portátil e que consiga manter as características de uma infraestrutura já definida.

Sendo assim e a proposta atual que estamos elaborando com base no ambiente qual a aplicação se encontra se faz necessário o uso de um plataforma que seja de características embarcada para que se possa substituir ade-

quadamente, dispositivos como smartphone, tablet não são plausíveis, visando as possibilidades futuras que a aplicação promove.

A Raspberry Pi 3 por ser mais robusta que a ESP32 supre de forma essencial as necessidades pois seu poder de processamento, a performance, amplia o possibilidade de serviços, exemplo, escanear dispositivos BLE (Bluetooth Low Energy), conectar e desconectar, efetuar a leitura e escrita de dados, além de registrar notificações diretamente dos dispositivos, sem ter que passar por outro controlador para codificar ou decodificar, abstraindo assim a camada do SO para disponibilizar os serviços com o protocolo HL7 a outras aplicações.

Desta forma permite a leitura por meio de sensores para acompanhar o estado clínico do paciente, utilizando as tecnologias de protocolo de comunicação Health Level 7 (HL7) para a representação e a transferência dos dados clínicos nesse padrão. O que auxilia administrativamente o tráfego entre sistemas de informação em saúde, tais como em clínicas, consultórios, hospitais, sistemas de saúde pública, entre outras. Foi necessário a construção de um software que pudesse efetuar esse papel de proxy embarcado, ou mesmo centralizador, usando assim a RaspBerry-Pi 3 responsável por processar os dados e enviá-los via padrão HL7 e disponibilizar os dados clínicos para um uso multiplataforma, ou seja, tanto para web ou mobile, esses recursos em um sistema de informação possibilita integra todos os dados e processos de uma organização em um único sistema. Dessa forma um conjunto de negócios, empresas ou entidades podem funcionar como uma unidade interagindo assim com o mercado para fornecer software e serviços, sendo assim, abaixo uma figura do diagrama que exemplifica o cenário com a Raspberry PI 3.

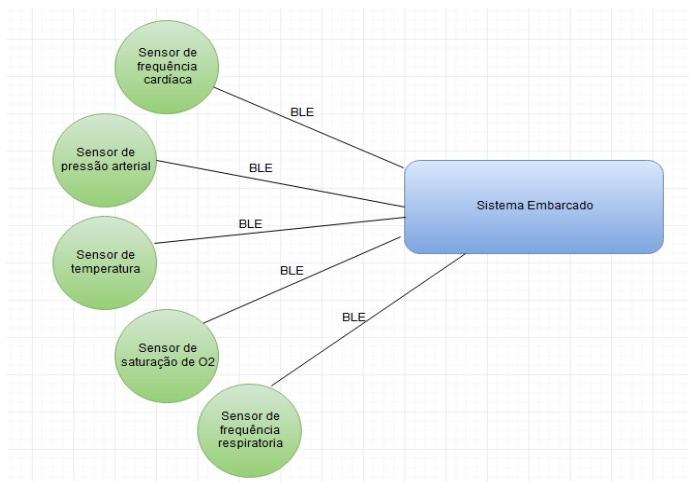


Figura 12: Cenário com o Raspberry PI 3.

A ideia é que possa oferecer mais do que uma simples ferramenta que envolve sensores para simples medições e sim uma ferramenta que possa mensurar o estado que fuja do padrão de homeostase⁵ do indivíduo e possa monitorá-lo em

⁵Homeostasia ou homeostase é a condição de relativa esta-

sua residência, auxiliando assim serviços específicos na área de saúde e até mesmo auxílio no atendimento *Home Care*⁶.

O primeiro passo para a construção do sistema foi estabelecer o escopo da aplicação define os pontos a serem monitorados e abordados, ou seja, frequência cardíaca, a saturação de (O₂), a capnografia, a pressão arterial, a temperatura e a frequência respiratória do paciente, segue uma rápida explanação sobre alguns desses parâmetros que serão utilizados para medir os dados deste projeto.

Saturação de O₂ - Oxímetro de Pulso de Dedo Portátil é o equipamento mais comum para tal medição e pode ser o acessório ideal para medir a saturação de oxigênio no sangue e verifica os batimentos cardíacos, além de ser muito útil na prática de atividades físicas. A saturação de oxigênio é definida como a proporção de oxí-hemoglobina com a concentração total de hemoglobina presente no sangue (isto é oxí-hemoglobina + hemoglobina reduzida) [16]. A hemoglobina é uma proteína que contém ferro, estando presente nos glóbulos vermelhos do sangue permitindo o transporte do oxigênio do pulmão para todo o corpo.

Logo o transporte do oxigênio que é feito pelas moléculas de hemoglobina é limitado em no máximo quatro moléculas de oxigênio por cada molécula de hemoglobina

Capnografia - Os capnógrafos analisam e registram a pressão parcial de CO₂ durante o ciclo respiratório por um sensor aplicado nas vias aéreas do paciente ou pela aspiração de uma amostra de ar nas vias aéreas processada por um sensor. A capnografia é utilizada como parâmetro indicativo de acidose respiratória incipiente e como ferramenta no auxílio ao desmame do respirador. Existe a possibilidade do uso do Monitor Multiparâmetro com Capnografia T5, que foi desenvolvido com alta tecnologia, a fim de promover uma visualização abrangente de perfil gráfico e uma medida específica da condição ventilatória de cada paciente [25].

Pressão arterial - existem várias maneira e equipamentos para medir a pressão arterial, o oxímetro monitor de oxigênio e Frequência Cardíaca Modelo CONTEC, eu mesclam a utilização de uma ferramenta com mais de uma funcionalidade acoplada a mesma.

Frequência cardíaca - A medição da frequência cardíaca óptica (OHR) baseia-se na fotopletismografia (PPG). Simplificando, ela usa luzes de LED e um detector de luz para medir as mudanças no tamanho dos vasos sanguíneos e transforma essas medidas em leituras da frequência cardíaca.

Frequência cardíaca óptica baseada no pulso

Meça a sua frequência cardíaca opticamente através da pele do seu pulso com facilidade e sem a necessidade de usar uma cinta peitoral separada enquanto estiver se exercitando.

Frequência cardíaca óptica baseada no braço

bilidade da qual o organismo necessita para realizar suas funções adequadamente para o equilíbrio do corpo.

⁶Também conhecido como internação domiciliar, é a prestação de assistência médica realizada na casa do paciente.

Monitore seus batimentos de forma conveniente e confortável em esportes em que a colocação do sensor pode atrapalhar seu movimento e desempenho ou afetar a qualidade do sinal da frequência cardíaca óptica.

Temperatura - A medição de temperatura é um processo pelo qual se busca medir o grau de agitação térmica das moléculas de uma determinada substância. Esse tipo de medição é muito importante para o controle de processos em diversos setores da indústria. É também o que ajuda a garantir a qualidade final de produtos das mais diversas áreas, desde alimentos até medicamentos.

Mas, é claro, para obter medições mais precisas, é necessário dispor de instrumentos certificados e que atendam aos mais altos padrões de qualidade. abaixo cito algumas ferramentas corriqueiras para medir a temperatura [39].

Termômetro portátil

O termômetro portátil é o tipo de instrumento utilizado para medir temperaturas em situações que não necessitam de medidas constantes, apenas eventuais e esporádicas [39]. Os termômetros digitais são instrumentos amplamente utilizados em indústrias, hospitais, armazéns, processos fabris etc. Há também o termômetro digital por contato, que utilizam pontas sensoras, geralmente intercambiáveis e com modelos diferentes de sensores para cada aplicação.

Termômetro de ambiente

Como o nome já indica, o termômetro de ambiente, também chamado de termômetro meteorológico, é utilizado para medir temperaturas de um determinado local. Tanto nos modelos analógicos como nos digitais, ele permite determinar a temperatura máxima e mínima de um ambiente (que pode ser desde uma sala até a parte interna de um freezer ou depósito, por exemplo)[39].

Termo-higrômetro

Assim como os termômetros de ambiente, o termo-higrômetro é um aparelho que serve para medir a temperatura de um determinado local. O diferencial é que esse instrumento também permite medir a temperatura externa e a umidade relativa do ar. Alguns modelos possuem até relógio e podem apresentar outras funções de medição, como decibelímetro (de decibéis) e luxímetro (de luminosidade)[39].

4.1 Requisitos

Esta seção trata de algo que não é uma funcionalidade, mas que precisa ser realizado para que o software atenda seu propósito, mas são de suma importância para o desenvolvimento do sistema, ou seja os requisitos não-funcionais que têm um papel de suma importância durante o desenvolvimento de um sistema, podendo ser usados como critérios de seleção na escolha de alternativas de projeto, estilo arquitetural e forma de implementação.

Nesse sentido temos:

- *Escalabilidade*

Deve ser possível aumentar ou diminuir o número de componentes sem causar impactos ao sistema, a leitura dos sensores permite que possam ser pareados com a tecnologia Bluetooth BLE que tente se conectar ao dispositivo.

- *Heterogeneidade*

O sistema deve ser capaz de lidar com componentes que distinguem nos seguintes aspectos: medidor de frequência cardíaca ou respiratória, a saturação de O₂, a pressão arterial, temperatura, a possibilidade de estabelecer um canal de comunicação via tecnologia bluetooth permite essa heterogeneidade.

- *Portabilidade*

Pela possibilidade que o Qt Creator oferece a aplicação é a sua capacidade de ser compilado, e executado em diferentes arquiteturas (seja de hardware ou de software). O termo pode ser usado também para se referir a re-escrita de um código fonte para uma outra linguagem de computador.

- *Baixo processamento*

Essa perspectiva da ferramenta aumenta a gama de possibilidades de dispositivos nos quais pode rodar, justamente por oferecer via Bluetooth BLE, ou seja de baixa energia (Low Energy), mas sem excluir dispositivos que tenham uma alta capacidade de processamento.

4.2 Ferramentas

Essa subseção detalha as ferramentas, tanto de hardware que foram utilizados para o projeto quanto a nível de software, utilizadas para a implementação e implantação.

- *Hardware*



Figura 13: Principais componentes a nível de hardware.

ESP-32 - Divisor de águas no mundo dos Embarcados e IOT. Dotado de um poderoso microcontrolador de 32

bits com WiFi integrado, tornou-se famoso por vir presente em toda sorte de placa, com destaque para uma placa barata e de fácil acesso.

Raspberry PI 3 - É um microcomputador completo com dimensões, aproximadas, de um cartão. A versão PI 3 conta com processador Broadcom Quad Core BCM2837 de 64 bits, 1G de RAM, além de adaptadores Wi-fi e Bluetooth já integrados.

SSD Card - São pequenos cartões de memória não voláteis desenvolvidos pela SD Association que são usados principalmente em dispositivos eletrônicos portáteis como celulares, câmeras e GPS, micro controladores para sistemas embarcado e afins, para fornecer ou aumentar o armazenamento desses dispositivos, o que no nosso projeto serviu para suportar o Raspbian versão 4.19, na última release com data de 10 de julho de 2019, tamanho de 1945 MB.

- Software

Raspbian - É um dos sistemas operacionais disponíveis para a instalação no Raspberry PI 3. Ele é baseado no Debian e foi utilizado como servidor MQTT. Raspbian é uma variante do Debian baseada no ARM hard-float, sendo um porte da arquitetura Wheezy, otimizada para o conjunto de instruções ARMv6 do hardware do Raspberry Pi. Raspbian é uma palavra-valise ou siglificação, composição de Raspberry Pi e Debian. Utilizado como base para o Raspberry PI 3 para que os teste do proxy embarcado rodasse num SO que fosse minimamente robusto, com a tecnologia do Bluetooth LE que já é nativa das configurações do Raspberry PI 3, dessa forma facilitando nossa comunicação com os dispositivos externos para leitura das informações. A imagem Buster Raspbian com desktop e software que é recomendada baseada no Debian Buster, contida no arquivo ZIP, tem mais de 4 GB, o que significa que esses arquivos usam recursos que não são suportados por ferramentas mais antigas de descompactação em algumas plataformas[41].

Qt Creator - Além de ser muito usado para desenvolvimento de aplicações Desktop, o Qt vem crescendo bastante na área de sistemas embarcados atendendo a indústria automobilística, médica, automação e IoT (Internet das Coisas) entre outras. O Qt nos disponibiliza todo um ecossistema podendo ser usado de maneira Open Source. Para desenvolvimento em Qt, é utilizado uma IDE(Integrated Development Environment) chamada Qt Creator [40]. Por meio dela conseguimos desenvolver uma interface gráfica satisfatória para atender de forma visual o contexto do projeto, ficou responsável por transmitir as informações lidas recebidas do servidor de desenvolvimento que estabelecemos para teste do tráfego dos dados da aplicação que vinham já no formato HL7.

Microsoft Visual Studio - Sendo um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software dedicado ao .NET Framework e às linguagens C, C++, C Sharp, entre outras. Serviu para construção do servidor, o mesmo foi conectado ao MySql para usar as informações do banco de dados para os testes no período do desenvolvimento.

MySQL - É um sistema de gerenciamento de banco

de dados, que utiliza a linguagem SQL como interface. Banco de dados da Oracle Corporation, foi usado para alimentar a aplicação web, mobile e armazenar os dados da plataforma que foram passados para o teste no decorrer do desenvolvimento.

4.3 Metodologia

Para atender aos objetivos do desse projeto e às demandas necessárias, às atividades de desenvolvimento do módulo eletrônico embarcado, junto ao servidor de comunicação com os concentradores para transmissão de dados dos sinais vitais e com o uso da linguagem compilada C++, oferecendo as mesmas vantagens de acesso à hardware, flexibilidade e performance. Oferece encapsulamento e controle de acesso a dados privados de uma classe, checagem forte de tipos e portabilidade, nesse contexto é coerente efetuar as seguintes operações : scanning, advertising, notification, read e write. Segue a imagem da estrutura de páginas da aplicação.

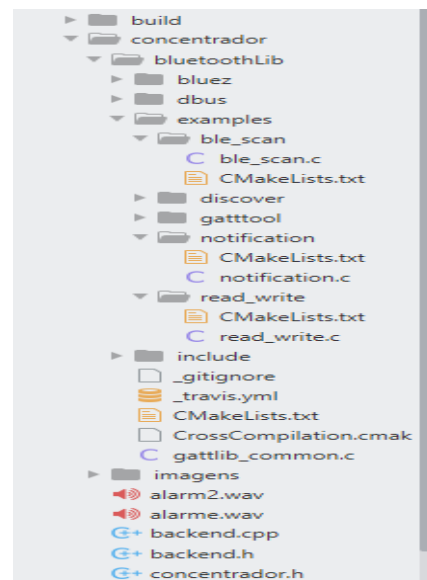


Figura 14: Divisão dos arquivos para o projeto.

Conectividade

O módulo BLE pode operar de duas maneiras distintas: modo *scanning* ou *advertising*. Nesse contexto nas proximidades é descobrir um dispositivo para pareamento, similar ao que ocorre com dispositivos na função Fast Pair, um processo sem problemas para parear seus dispositivos Bluetooth em dispositivos Androids e afins. O modo scanning nada mais é do que a leitura de advertising emitidos por outros *devices*(*dispositivos*), este procedimento é chamado de device discovery(Descoberta de Dispositivos). Quando está em modo *scanning*, este pode ser um *scan* do tipo passivo ou ativo[42]. A principal diferença entre eles é que o *scan* ativo pode enviar uma requisição de informação adicional para o *advertiser*, o, mencionado anteriormente, *Scan Response Request*, já o *scan* passivo somente faz a leitura dos pacotes[42].

Nesse contexto os dados a serem trabalhados, serão trafegados no padrão HL7, a notificação facilitará o aprazamento da administração de possíveis medicamentos, possibilitar o agendamento de visita do profissional, informa ciência do cuidador para o servidor e registro da administração realizada, esse proxy embarcado será a via de transmissão de dados de sinais vitais e das notificações de administração de medicamentos, conforme planejado na figura abaixo.

	Concentrador
Sinal Vital (pressão arterial, SaO2, temperatura, frequência cardíaca)	Notificação e Aprazamento da Coleta Automática através dos sensores
Sinal de Emergência e Queda do paciente	Monitora os sensores, enviam o sinal para o servidor e recebe ciência da Interface WEB
Administração de Medicamentos	Notificação e aprazamento da administração de medicamentos, informa ciência do cuidador para o servidor e registro da administração realizada

Figura 15: Funcionalidade do Centralizador.

A comunicação entre o centralizador e o servidor de comunicação com os concentradores utiliza o Serviço Geral de Pacotes por Rádio (GPRS) com criptografia AES-256.

Configurar o Bluetooth Low Energy

Como fui por osmose, esqueci de testar se nessa nova versão do sistema operacional (Jessie) é necessário esse procedimento, mas de qualquer modo, servirá para versões anteriores do RPi.

Após instalação foi feita a compilação do Bluez. A pilha BlueZ foi desenvolvida inicialmente por Max Krasnyansky na Qualcomm e em 2001 (3 de maio) eles decidiram lançá-la sob a GPL. O anúncio foi enviado para a lista de discussão dos desenvolvedores do Axis Bluetooth e este se tornou o início da pilha oficial do Linux Bluetooth. A tecnologia sem fio Bluetooth é uma especificação mundial para uma solução de rádio de baixo custo e fator de forma pequeno que fornece links entre computadores móveis, telefones celulares, outros dispositivos portáteis portáteis e conectividade com a Internet.

5. IMPLANTAÇÃO

Tivemos duas vertentes este projeto, uma da qual se baseia no cenário da ESP-32 que foi estabelecido como protótipo e teste do experimento no qual já foi citado nas seções anteriores, sendo assim conseguimos montar a estrutura para atender as necessidades iniciais. O código se encontra na linguagem C++, foi utilizado o socket que nada mais é que um ponto final de um fluxo de comunicação entre processos através de uma rede de computadores.

Como o diferencial do HL7 e as potencialidades que ele nos permitiu executar, usando o exemplo da troca de mensagens que segue na imagem abaixo.

```

HL7® V2.X (ER7)

MSH|^~\&|APPELVIZO|LOCAL|EINVO|APPELVIZO|LOCAL|REMOTO|201810011300||ORU^R01|0001|T|2.5.1
PID||ID||LAST^NAME^SECOND|19700121|M||RUAA^CIDADE^UF^CEP||PHONE|||CNS
PV1||E|ER|||CRM^LAST^FIRST
ORC|MI|12345|LAB|||D12345^LAST^NAME
OBX|1|TX||Clinical History: HISTORICO CLINICO|||F
OBX|2|TX||Finding: ACHADOS|||F
OBX|3|TX||Conclusion: CONCLUSOES|||F
NTE|1|L|NOTAS E COMENTARIOS
  
```

Figura 16: Formato HL7 na troca de mensagens.

Sendo assim para encapsulamento do protocolo HL7 junto com a estrutura do centralizador que permite essa utilização, o trafegado o pacote se encontra via Socket, num servidor nosso com IP estabelecido para o teste.

Nesse contexto o compartilhamento de dados clínicos entre centros médicos, o que promove um gama de possibilidades com a geração desses dados, via uso para um Data Warehouse, uma base de análises clínicas para auxílio do paciente e anamneses que podem ser mais precisas. Nesse sentido, sua personalização e coleta elaborada se torna fundamental, visto que as informações recolhidas variam de acordo com a especialização. A exemplo da anamnese que é um tipo de análise feita pelo médico ao paciente, onde o mesmo é submetido a diversas perguntas que auxiliarão o médico a descrever no documento o diagnóstico.

Aliada a outros exames, como físicos, por exemplo, ela é parte essencial dos exames clínicos, por isso, é um documento constantemente utilizado em empresas de saúde e em clínicas de saúde ocupacional. Esses exames ajudam a identificar doenças, bem como traçar os motivos de suas causas e o tratamento para tal. Mas, ainda que o avanço da tecnologia tenha melhorado muitas questões médicas, a anamnese parece ter ficado esquecida por muitas empresas de *softwares* que “engessam” no modelo e dificultam o trabalho do dia a dia dos profissionais.

Consiste que uma padronização por meio do HL7 seria uma solução viável para as trocas de informações gerenciadas entre os dados fornecidos pelo paciente assim como deixar as informações sobre o mesmo atualizada na empresa que gere a saúde e bem estar do mesmo. Visando essas utilizações aproveitamos o experimento para dar seguimento a parte Web que também está sendo alimentada pela plataforma, ou seja, o concentrador, auxiliado no monitoramento web do paciente usando servidor de comunicação com os centralizadores para monitorar os sinais vitais, o que a solução permite atendimento a domicílio do paciente, envio de sinal de emergência, sendo que essa interface web promove uma interação dos profissionais de saúde, com os pacientes.

Sinal Vital (pressão arterial, SaO2, temperatura, frequência cardíaca)	Inserir Prescrição e apresentar valores coletados
Sinal de Emergência e Queda do paciente	Receber notificação do servidor e informar ciência
Administração de Medicamentos	Inserir Prescrição, apresentar ciência do cuidador sobre a notificação e apresentar registro de administração

Figura 17: Características para Plataforma Web.

6. RESULTADOS OBTIDOS

Nessa etapa do projeto já foi possível por meio conectar a ESP32 com um equipamento o que funcionou perfeitamente, pode-se fazer um teste com o manguito para trafegar o dado avaliado.

Colocamos um servidor que fornece uma porta, sendo o IP não é dinâmico, somente para requisição estabelecida para o uso da leitura dos dados configurados no protocolo HL7 que já possuímos das leituras anteriores e teste prototipados para dar seguimento no experimento, dedicado a prover dados para a aplicação dentro de uma rede LAN ou WAN. Usando um socket para comunicação, podemos estabelecer a read(leitura), fechamento e abertura de conexão, aliado a ESP32 serviu de base para os teste efetuados, poderíamos usar um monitor multiparamétrico no melhor caso, sendo de extrema importância ao longo do desenvolvimento do projeto, como podemos observar logo abaixo.

```

587     }
588     break;
589 }
590
591 case MPL_DESTROY:
592 {
593     PostQuitMessage(0);
594     shutdown(ServerSocket, SD_BOTH);
595     closesocket(ServerSocket);
596     WSACleanup();
597     return 0;
598 }
599 break;
600
601 case MPL_SOCKET:
602 {
603     if (!) {
604         switch (WSAGETSELECTEVENT(IParam))
605         {
606             case FD_READ:
607                 {
608                     ...
609                 }
610                 break;
611             case FD_CLOSE:
612                 {
613                     ...
614                 }
615                 break;
616             case FD_ACCEPT:
617                 {
618                     ...
619                 }
620                 break;
621             else if (IParam == ServerSocket_s2) {
622                 switch (WSAGETSELECTEVENT(IParam)) { case FD_READ: { break; } }
623             }
624         }
625     }
626     return DefWindowProc(hwnd, msg, wParam, lParam);
627 }
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667

```

Figura 18: Condições com o uso do socket.

```

365 kprintf(str, hEditOut);
366 if (pptr->Fig_panic == 1) kprintf("PACIENTE EM PANICO!!!!\n", hEditOut);
367 if (pptr->Fig_fall == 1) kprintf("PACIENTE EM QUEDA!!!!\n", hEditOut);
368
369 sprintf(data, "%d-%02d-%02d", vamo, vmes, vdia);
370 sprintf(hora, "%02d-%02d-%02d", vhora, vmin, vseg);
371
372 GetLocalTime(&localtimes);
373 sprintf(qry, "%d-%02d-%02d-%02d-%02d-%02d", localtimes.wYear, localtimes.wMonth, localtimes.wDay, localtimes.wHour, localtimes.wMinute, localtimes.wSecond);
374 strcpy(dataIncluse, qry);
375
376 szoutData.id2 = pptr->id2;
377 szoutData.hora = localtimes.wSecond + localtimes.wMinute * 60 + localtimes.wHour * 3600;
378 szoutData.vData = localtimes.wYear << 16 | localtimes.wMonth << 8 | localtimes.wDay;
379 if (pptr->idamy[0] == 0x03) { // HL7 = msg
380     char msgh1[1024] = {"\r\nSH|^~|^S|northern|patient monitor|||20120101124019|ORU^R01^ORU_01|2124|P|2.6|v\r\n"};
381     "PID|||||12345|gabriel^muniz|^|00000000||v\r\n";
382     "PV1|||||^v\r\n";
383     "OBX|1|||20120101124019|v\r\n";
384     "OBX|1|N|147842^MDC_ECG_HEART_RATE^MDC|1.7.4.147842|60|264864^MDC_DIH_BEAT_PER_MIN^MDC|||||F|||||v\r\n";
385     "OBX|2|N|150456^MDC_PULS_OXIM_SAT_O2^MDC|1.3.1.150456|90|202688^MDC_DIH_PERCENT^MDC|||||F|||||v\r\n";
386     "OBX|3|N|149530^MDC_PULS_OXIM_PULS_RATE^MDC|1.3.1.149530|60|264864^MDC_DIH_BEAT_PER_MIN^MDC|||||F|||||v\r\n";
387     "OBX|4|N|151578^MDC_THOR_RESP_RATE^MDC|1.7.1.151578|20|254928^MDC_DIH_RESP_PER_MIN^MDC|||||F|||||v\r\n";
388     "OBX|5|N|150344^MDC_TEMP^MDC|1.2.1.150344|36.1|268192^MDC_DIH_DEGC^MDC|||||F|||||v\r\n";
389     "OBX|6|N|150344^MDC_TEMP^MDC|1.2.2.150344|-100|268192^MDC_DIH_DEGC^MDC|||||F|||||v\r\n";
390     "ORU^R01^ORU_01|||ORU^R01^ORU_01|||P|2.6|v\r\n";
391     "PID|||||^v\r\n";
392     "PV1|||||^v\r\n";
393     "OBX|||||^v\r\n";
394     "OBX|15|T|^MDC|1.1.1.150037.1|||^H-PL-SP|||F|||||v\r\n";
395     kprintf("!!PAQOTE 7!!", hEditOut);
396     strcpy(ph17.texto, msgh1);
397     int outDataLength = send(Socket[n][0], (char*)&ph17, sizeof(ph17), 0);
398     p2->szoutDataLen7 = ph17;
399 }
400 else {
401     int outDataLength = send(Socket[n][0], (char*)&szoutData, sizeof(szoutData), 0);
402     p2->szoutData = szoutData;
403 }
404 _asm { nop }
405 packinvalid;
406 }
407 }
408 }

```

Figura 19: Código servidor com HL7.

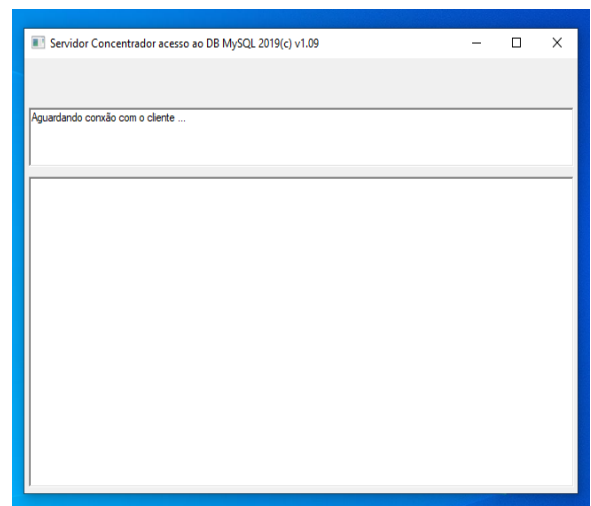


Figura 20: Servidor aguardando requisição.

Após os experimento da primeira etapa serem concretizados, já foi possível vislumbrar os passos para a segunda etapa experimento no qual focamos mais na estrutura com o uso da Raspberry Pi 3, o controlador com a interface do QT Creator, o que possibilitou usar uma biblioteca para o uso do Bluetooth Low Energy, a necessidade foi integrar de forma simplificada a biblioteca com o a criação da aplicação no QT que simula a prototipagem do centralizador do nosso projeto que já foi elaborado no primeiro experimento.

Foi feita uma modificação para atender o mesmo uso na Raspberry Pi 3 para que caracteriza-se como cliente e conseguisse exibir os dados lidos que foram colhidos na prototipagem na primeira etapa do projeto, sendo assim tiveram mudanças cruciais para que funcionasse a leitura dos dados, essas modificações não impactaram o foco principal da segunda etapa do desenvolvimento da aplicação planejada. Ainda com a primeira etapa conseguimos prover dados para

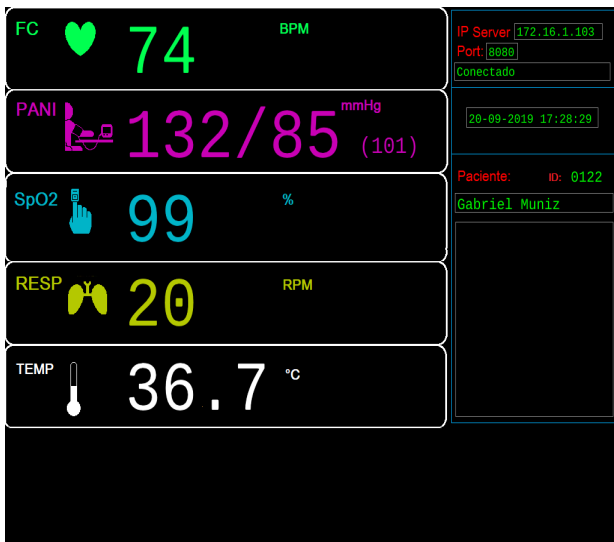
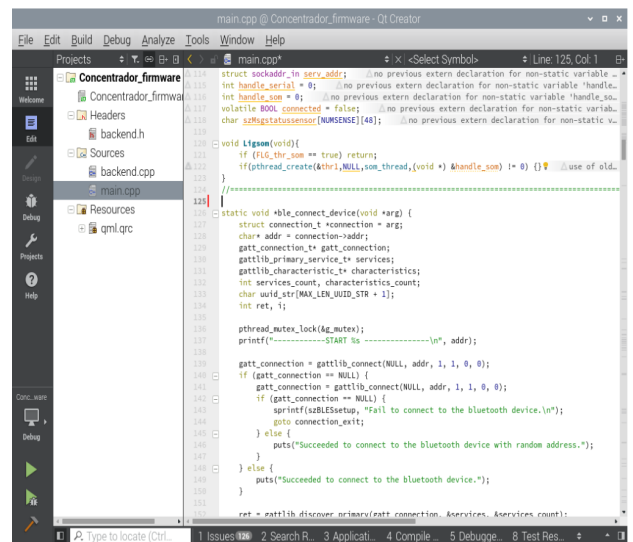


Figura 24: Valores exibidos no centralizador.

O alicerce da primeira etapa, a qual o que foi testado e estabelecido já surtiu resultados passamos para a segunda etapa do projeto, o qual se torna a mais demorada para aprendizado da tecnologia Bluetooth LE e as suas características quanto às modificações para aplicabilidade na Raspberry Pi 3, o que se encontra no servidor criado para os testes preliminares permanece, e o que será modificado é o que coordena para fazer as novas tarefas, ou seja, receber os sinais, montar esses mesmo dados no padrão HL7, salvar essas informações na base de dados, assim como receber os dados já organizados para exibição dos mesmo como mostramos nos exemplos anteriores.

Na segunda etapa, foi utilizar a tecnologia Bluetooth Low Energy com a estrutura já estabelecida do firmware do concentrador desenvolvido, anteriormente conseguimos fazer a comunicação usando a ESP32 para mandar os dados, com a finalidade do mesmo mandar os dados para o servidor que consequentemente salva no banco, para que esses dados sejam compartilhados a plataforma mobile e web do projeto. Então tínhamos a comunicação: sensores de sinais vitais, passando para a ESP32 que por sua vez se conectava ao centralizador para que esses dados fossem para o banco de dados na estrutura do MySQL, que por sua vez iria alimentar outras plataformas. Removendo a ESP32 o centralizador teria que fazer as tarefas de ler os dados via tecnologia Bluetooth LE, e transmitir para o banco de dados na arquitetura do HL7, assim como receber os dados para exibição no formato amigável para o teste que elaboramos, assim como um possível monitor multiparamétrico.

Usando uma biblioteca GATT(Generic Attribute Profile) o que define a maneira como dois dispositivos Bluetooth Low Energy transferem dados para frente e para trás usando conceitos chamados Serviços e Características[34]. Um periférico BLE pode ser conectado a um dispositivo central modificamos um pouco o comportamento da ferramenta para comunicar-se com os sensores para leitura dos sinais vitais desejados. Junto com a biblioteca do **BlueZ, versão 5** que também foi utilizada na aplicação, um biblioteca que é nativa dos pacotes do raspbian, que foi o SO que usamos na Raspberry PI 3. Observe na figura abaixo.



A aplicabilidade no universo da saúde é o auxílio domiciliar (home care), devido ao grau de importância destinado a esse tipo de cuidado no atendimento domiciliar personalizado, que não seja tão complexo quanto os programas de Internação domiciliar e prestação de serviço oferecidos hoje. O cuidado de uma monitoria precisa e que seja um procedimento minimamente invasivo ao paciente se faz necessário nesse contexto o uso dessa técnica que o projeto visa e demonstra a plausível eficácia quanto ao monitoramento e tráfego de dados referentes ao sinal vital.

As características do sistema são escalabilidade e facilidade de implantação, pois possibilita a adição de novos elementos que possam enviar os sinais vitais, sem a necessidade de parar o sistema. Nota-se que não somente com o uso do protocolo HL7, que foi o estabelecido como padrão para o desenvolvimento e avaliação dos dados, também pode-se perceber pela arquitetura montada que podemos estabelecer um padrão proprietário de comunicação caso não utilize o HL7, as características do dispositivo possibilita esse uso, modificando sutilmente o codificador e decodificador já se encontra a possibilidade de uso com distinto do protocolo de tráfego das informações.

Ao longo do desenvolvimento do projeto tive todo o suporte para ser engajado ao mesmo, desde da parte de hardware como o desenvolvimento da parte mobile e Web o que facilitou minha interação e compreensão do que já se encontrava em andamento. Meu orientador, assim com, meu coorientador que foi extremamente solícito a me ajudar e ensinar as tecnologias que tive a oportunidade de trabalhar, sendo que é uma área que tenho um apreço muito grande e que pretendo dar seguimento e incorporar os aprendizados a minha carreira.

7.1 Trabalhos futuros

- Possibilidade de usar padrão HL7 na versão 3 que se encontra em XML, foi fortemente influenciado pelo governo e pelos usuários de informações médicas, é baseado em uma metodologia formal (o HDF) e em princípios orientados a objetos, não sendo compatível nativamente com o HL7 V2 e ainda assim com essas modificações a plataforma Raspberry Pi 3 conseguirá ser útil;

- Estudo mais aprofundado sobre a temática a fim de que se possa levantar mais técnicas para o aperfeiçoamento da transferência de dados no padrão HL7;

- Melhoria dos componentes de hardware a fim de criar uma unidade mais robusta para a transferência de outros dados mais rebuscados;

Testes preliminares mostraram ser viável a transmissão dos sinais. O uso da plataforma de prototipagem rápida ajuda a testar a técnica, e destacar a importância dos dispositivos de baixo custo para transmitir dados sem fio. Outros passos estão em desenvolvimento, tais como a codificação do software para monitoramento web facilitando as informações para o paciente e facilitação da análise por um especialista.

Este trabalho, como citados ao longo do projeto, se propõe a demonstrar que a área de tecnologias para auxílio

da saúde é uma crescente e o uso delas na área de saúde é relevante ao meio, e a proposta mostra-se importante, pois foca na maior independência do paciente. Assim como coopera para clínicas e hospitais que precisam monitorar corriqueiramente dados de sinais vitais para mensurar o quadro clínico dos seus pacientes.

8. REFERÊNCIAS

[1] Leal, G., Flores G., Rosa M. e Giglio R. Tutorial de utilização do padrão HL7 FHIR para troca de mensagens de sistemas em Java. Disponível em: <https://medium.com/@floresgabriela/tutorial-de-utiliza%C3%A7%C3%A3o-do-padr%C3%A3o-hl7-fhir-para-troca-de-mensagens-de-sistemas-em-java-335907ca71c6>. Acessado em: 15 de Junho de 2019.

[2] ARAÚJO, R. B. (2003). Computação Ubíqua: Princípios, Tecnologias e Desafios. XXI Simpósio Brasileiro de Redes de Computadores. (Org.). 1 ed. Natal – RN: SBRC. p.45 – 115, 2003. Disponível em: <https://www.tiespecialistas.com.br/definicao-e-principios-da-computacao-ubiqua/>. Acessado em: 22 de Março de 2019.

[3] Microcontrolador. Disponível em: <https://pt.wikipedia.org/wiki/Microcontrolador> Acesso em: 21 de julho de 2019.

[4] Pereira, F. Apollo Guidance, Navigation and Control Hardware Overview Microcontroladores RL78 Guia Básico – Renesas, p. 17 e 18. Disponível em: <https://www.embarcados.com.br/agc-primeiro-grande-sistema-embarcado/>. Segundo link disponível: https://en.wikipedia.org/wiki/Apollo_Guidance_Computer. Acessado em: 22 de Março de 2019.

[5] Ganssle, J. G., Taylor, Francis. Navigation and Control Hardware Overview. p.1-60,2003. Disponível em: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20090016290_2009014409.pdf Acessado em: 14 de Fevereiro de 2019.

[6] Microcontroladores - O que são, Para Que Servem e Onde São Usados. Disponível em: <https://www.eletronicaprogressiva.net/2014/08/Microcontroladores-O-que-sao-Para-que-servem-Where-sao-usados.html>. Acesso em 18 de julho de 2019.

[7] Laplante, Philip A. Dictionary of Computer Science, Engineering and Technology. p.1-366. Disponível em: https://books.google.com.br/books?id=U1M3c1UwCfEC&pg=PA366&redir_esc=y#v=onepage&q&f=false. Acessado em: 16 de Abril de 2019.

[8] R. BATISTELLO. Automação residencial utilizando raspberry pi e android. p. 1-86, 2014. Disponível em: <http://www.webartigos.com/storage/app/uploads/public/588/4ce/5f4/5884ce5f4b96e498733571.pdf>. Acessado em: 16 de Junho de 2019.

[9] Health Level-7 ou HL7. Disponível em:

- <http://interopera.esy.es/entendendo-o-papel-das-interfaces-hl7-nas-informacoes-de-saude/>. Acessado em: 14 de Junho de 2019.
- [10] Gonçalves, P. R. P. MONITORIZAÇÃO REMOTA DE PACIENTES EM AMBULATÓRIO - Dissertação apresentada à Universidade Fernando Pessoa, para obtenção do grau de Mestre em Computação Móvel. Porto Alegre. p. 1-86, 2009.
- [11] Sinais Vitais Def - RESCUER. Disponível em: <http://aprendis.gim.med.up.pt/index.php/MonitorizacaoDeSinaisVitais>. Acessado em: 12 de Março 2019.
- [12] R. S. PENIDO, Edilus de Carvalho Castro; TRINDADE. Microcontroladores. e-Tec Brasil, 2013. Acesso em 18 de Julho de 2019.
- [13] ESP32 BLUETOOTH LOW ENERGY . Disponível em: <https://www.fernandok.com/2018/08/esp32-bluetooth-low-energy.html>. Acessado em : 27 de Agosto de 2019.
- [14] O que é o Raspberry Pi?. Disponível em: <https://tecnoblog.net/282739/o-que-e-o-raspberry-pi/> . Acessado em : 15 de Junho de 2019.
- [15] MURTA, G. F.; RUSSI, J. N. Procedimentos Básicos de Enfermagem no Cuidar. São Caetano do Sul: Difusão Editora, p.15-30, 2006.
- [16] Oxímetro de Pulso – O que é saturação de oxigênio (SpO2)?. Disponível em: <https://dimave.com.br/blog/oximetro-de-pulso-spo2/>. Acessado em : 17 de Agosto de 2019.
- [17] MORAIS, BRUNO MAIA - Serviços de Atenção domiciliar baseada em Redes de Sensores Sem fio e Sistemas Embarcados. Universidade Federal da Paraíba, João Pessoa, p.1-104, 2012. Disponível em: <https://repositorio.ufpb.br/jspui/handle/tede/6099>. Acessado em: 20 de Maio de 2019.
- [18] Souza, A. R. de R. Mesquita, F. C. C. UFPel. Monitoramento de Sinais Vitais REVISTA LUPS, VOL. 2, NO. 1, MARO 2013. Disponível em: http://ubiq.inf.ufpel.edu.br/arrsouza/lib/exe/fetch.php?media=artigo_monitoramento_de_sinais_vitais_rev_4_.pdf. Acessado em: 20 de Maio de 2019.
- [19] Araujo, R. de B. Computação ubíqua: princípios , tecnologias e desafios.XXI Simpósio Brasileiro de Redes de Computadores. Universidade Federal de S. Carlos (UFS-Car) – São Carlos - SP. p. 1-70, 2004. Disponível em: <http://professordiovani.com.br/rw/monografia-araujo.pdf>. Acessado em: 17 de Junho de 2019.
- [20] INTRODUÇÃO A BLUETOOTH SMART (BLE). Disponível em: <https://www.embarcados.com.br/bluetooth-smart-ble/>. Acessado em: 21 de Junho de 2019.
- [21] Bluetooth Low Energy - BLE. Disponível em: <https://www.embarcados.com.br/bluetooth-low-energy/> . Acessado em: 21 de Junho de 2019.
- [22] Rades, P. Entendendo o papel das interfaces HL7 na troca de informações de saúde. Disponível em: <http://interopera.esy.es/entendendo-o-papel-das-interfaces-hl7-nas-informacoes-de-saude/>. Analista de Sistemas, Gestor Hospitalar, Especialista em Informática em Saúde e em Redes de Dados Médicos. Link suporte disponível em: <http://interopera.esy.es/empresa/staff/>. Acessado em: 15 de Junho de 2019.
- [23] Davies J.G. Lee, T. F. Microsoft Windows Server 2003 TCP/IP Protocols and Services. ISBN 0-7356-1291-9. Disponível em: <https://pt.wikipedia.org/wiki/TCP/IP>. Acessado em: 27 de Junho de 2019.
- [24][Portuguese] Sockets em linguagem C. Disponível em: <https://www.exploit-db.com/papers/13634>. Acessado em: 28 de Junho de 2019. TRANSMISSION CONTROL PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION. Link suporte disponível em: <http://www.faqs.org/rfcs/rfc793.html>. Acessado em: 28 de Junho de 2019.
- [25] Monitor multiparametro com capnografia. Disponível em: <http://www.prolife.com.br/monitor-multiparametro-capnografia>. Acessado em: 14 de Agosto de 2019.
- [26] ANJOS, I. S. NOBRE, JR. V. A. BARBOSA, E. L. Monitoramento à distância de temperatura em pacientes neutropênicos com neoplasia hematológica internados no Hospital das Clínicas da Universidade Federal de Minas Gerais (UFMG) Projeto de Mestrado. Faculdade de Medicina da UFMG. Belo Horizonte, 2015. Disponível em: http://aprendis.gim.med.up.pt/index.php/Monitoriza%C3%A7%C3%A3o_de_Sinais_Vitais. Acessado em: 09 de Julho de 2019.
- [26] Carrara Dr. Dirceu, Dra. Ariane F. Machado Avelar, Denise M. Kusahara e Profa. Dra. Mavilde L. G. Pedreira, Câmara Técnica do Coren SP, - Oximetria de pulso arterial - Artigo publicado em: São Paulo, 22 dezembro de 2009. Disponível em: <https://portal.coren-sp.gov.br/sites/default/files/oximetria%2022-12.pdf> . Acessado em : 08 de Julho de 2019.
- [27] Middleware. Disponível em: <https://portogente.com.br/portopedia/74181-middleware> . Acessado em: 09 de Julho de 2019.
- [28] ■What is Middleware?■. Middleware.org. Defining Technology. 2008. Consultado em 11 de agosto de 2013. Arquivado do original em 29 de junho de 2012. Disponível em: <https://pt.wikipedia.org/wiki/Middleware>. Acessado em: 9 de Julho de 2019.
- [29] Bernardo, Alexandre Moretti. PROPOSTA DE SISTEMA EMBARCADO PARA AUXÍLIO E MONITORAMENTO DO IDOSO USP-ESCOLA DE ENGENHARIA DE SÃO CARLOS. p. 1-171, 2015.

- [30] Pereira, P. F. de O., Roque, S. Técnica de Controle para Sistemas Embarcados aplicado no Monitoramento Remoto de Sinais vitais. p. 1-3, 2014. Disponível em: <https://www6.univali.br/seer/index.php/acotb/article/viewFile/5353/2809>. Link Suporte Disponível em: <http://www.eati.info/eati/2013/assets/anais/artigo317.pdf>. Acessado em: 11 de Julho de 2019.
- [31] Sergio T, Carvalho, Erthal M, Mareli D, Sztajnberg A, Copetti A, Loques O. Monitoramento Remoto de Pacientes em Ambiente Domiciliar. p.1-8, 2011.
- [32] Pereira, Bruno Ricardo da Silva. Implementação de um parser HL7 para integração do ALERTR© com aplicações terceiras. p. 1-134, 2008. Disponível em: <https://repositorio-aberto.up.pt/bitstream/10216/59893/1/000129616.pdf> . Acessado em: 20 Junho de 2019.
- [33] Geraldo, Gabriel Crelier dos Santos. Implementação do Segmento Laboratorial do Protocolo HL7 Usando o MongoDB. p. 1-61, 2014. Disponível em: <http://bsi.uniriotec.br/tcc/textos/201412Crelier.pdf>. Acessado em: 20 Junho de 2019.
- [34] GATT p.1-30, 2014. Disponível em: <https://www.google.com/search?client=firefox-b-d&q=GATT+%28Generic+Attribute+Profile%29> . Acessado em 30 de Agosto de 2019.
- [35] Pereira, Fábio. Microcontroladores RL78 Guia Básico. p.1-50, 2013. Joinville - SC. Disponível em: http://www.perse.com.br/novoprojetoperse/BSU_Data/Books/N1367506931413/Amostra.pdf . Acessado em: 20 de Abril de 2019.
- [36] ESP32. Disponível em: <https://pt.wikipedia.org/wiki/ESP32> . Acessado em 22 de Julho de 2019.
- [37] ESP32 – Um grande aliado para o Maker IoT. Disponível em: <https://www.filipeflop.com/blog/esp32-um-grande-aliado-para-o-maker-iot/> . Acessado em 22 de Julho de 2019.
- [38] HL7. Disponível em: <http://aprendis.gim.med.up.pt/index.php/HL7> . Acessado em 23 de Julho de 2019.
- [39] Tipos de medidores de temperatura. Disponível em: <http://blog.instrusul.com.br/tipos-de-medidores-de-temperatura/>. Acessado em 18 de Agosto de 2019.
- [40] Introdução ao Qt Creator. Disponível em: <http://ninjadolinux.com.br/introducao-ao-qt-creator/>. Acessado em 19 de Agosto de 2019.
- [41] Raspberrypi Org. Disponível em: <https://www.raspberrypi.org/>. Acessado em 18 de Agosto de 2019.
- [42] Iwami, B. S. Advertising em Bluetooth Low Energy. p.1-66, 2018. São Carlos- SC. Disponível em: www.tcc.sc.usp.br/tce-06082018-164129/publico/Iwami_Bruno_tcc.pdf. Acessado em 18 de Agosto de 2019.
- [43] Qt Creator. Disponível em: <https://pt.wikipedia.org/wiki/Qt>. Acessado em 22 de Agosto de 2019.
- [44] Qt Creator/pt. Disponível em: https://wiki.qt.io/Qt_Creator/pt. Acessado em 21 de Agosto de 2019.

HL7® V3 (XML)

```
<POLB_IN224200 ITSVersion="XML_1.0" xmlns="urn:h17-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id root="2.16.840.1.113883.19.1122.7" extension="CNTRL-3456"/>
  <creationTime value="200202150930-0400"/>
  <!-- The version of the datatypes/RIM/vocabulary used is that of May 2006 -->
  <versionCode code="2006-05"/>
  <!-- interaction id= Observation Event Complete, w/o Receiver Responsibilities -->
  <interactionId root="2.16.840.1.113883.1.6" extension="POLB_IN224200"/>
  <processingCode code="P"/>
  <processingModeCode nullFlavor="OTH"/>
  <acceptAckCode code="ER"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="GHH LAB" root="2.16.840.1.113883.19.1122.1"/>
      <asLocatedEntity classCode="LOCE">
        <location classCode="PLC" determinerCode="INSTANCE">
          <id root="2.16.840.1.113883.19.1122.2" extension="ELAB-3"/>
        </location>
      </asLocatedEntity>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id root="2.16.840.1.113883.19.1122.1" extension="GHH OE"/>
      <asLocatedEntity classCode="LOCE">
        <location classCode="PLC" determinerCode="INSTANCE">
          <id root="2.16.840.1.113883.19.1122.2" extension="BLDG24"/>
        </location>
      </asLocatedEntity>
    </device>
  </sender>
  <!-- Trigger Event Control Act & Domain Content -->
</POLB_IN224200>
```

HL7 versão 3 com XML