

## 2

### Descrição do Trabalho

Este capítulo discorrerá sobre os aspectos teóricos que fundamentaram a elaboração do presente trabalho, a forma como foi desenvolvido, e, por fim, as soluções de *hardware* encontradas.

Na seção “Conceitos Básicos” serão definidos os principais conceitos que influenciaram a construção do *framework* proposto.

Na seção “Desenvolvimento”, serão abordados os pontos principais da estrutura proposta, existentes independentemente da finalidade para a qual venha a se destinar. Tal abordagem é de extrema importância para o correto entendimento do capítulo seguinte – Estudo de Casos (militar e Monitoramento e controle de navegação), onde serão tratadas as derivações decorrentes da especificidade de cada segmento ilustrado.

Por fim, na seção “*Hardware*” serão explicitadas todas as características especificadas como de interesse para implementação do Sistema de Apoio ao Comando e Controle e a descrição da solução de *hardware* encontrada.

#### 2.1

##### Conceitos Básicos

A apresentação em detalhes dos conceitos que se seguirão é de grande relevância para o entendimento do escopo do trabalho. É necessário definir e discutir alguns termos recorrentemente empregados ao longo do texto. Serão tratadas, nesta seção, as definições de **Comando, Controle, Comando e Controle–C2, Sistemas de Comando e Controle e Ciclo de Boyd**.

Em que pese a origem militar dos termos, apenas a título de ilustração, não é difícil estabelecermos a possibilidade de emprego destes conceitos em estruturas gerenciais governamentais ou privadas, onde não mais a figura de um comandante militar mas a de um presidente, gerente ou função similar, tem igualmente a função de coordenar e controlar uma equipe de subordinados.

Para corroborar a validade de tal assertiva, podemos citar o exemplo da obra de Sun Tzu (544–496 A.C), considerado um dos maiores estrategistas militares de todos os tempos. Em seu famoso livro “A Arte da Guerra” o citado autor relata diversas manobras estratégicas militares que foram mi-

nuciosamente estudadas e publicadas por uma grande diversidade de autores contemporâneos com enfoques diversos. Apenas para citarmos alguns exemplos conhecidos temos em “Sun Tzu para o Sucesso” e “Sun Tzu: estratégia de vendas” ambos de Michaelson, Gerald A., obras que partem de princípios, originalmente empregados no âmbito militar, para tratar de assuntos como sucesso pessoal e estratégias de vendas.

Poderíamos citar aqui outros tantos autores que se baseiam igualmente em Sun Tzu para abordarem os mais variados segmentos do conhecimento humano, dada a tamanha diversidade existente na literatura. Entretanto, acredito ser bastante claro e já comprovado que variados conceitos e estratégias, oriundas do meio militar, não ficam apenas circunscritas à esse meio, sendo, portanto, permeáveis para diversas outras áreas.

Definiremos, a seguir, os termos:

**Comando:** “Autoridade decorrente de leis e regulamentos, atribuída a um militar para coordenar e controlar uma tropa, em qualquer nível, em razão de um posto e função.” (7)

Apesar da definição citar, explicitamente, a figura de um militar na posição de comando, neste trabalho extrapolamos o conceito para qualquer elemento que, decorrente de uma posição funcional, legalmente estatuída, tem o encargo de coordenar e controlar um determinado grupo, visando à consecução de alguma meta.

**Controle:** “O Controle é definido como o ato ou efeito de acompanhar a execução de qualquer empreendimento, de forma a não permitir que ele se desvie do propósito estabelecido, através da avaliação permanente e correlação das atividades desenvolvidas.” (7)

Neste caso, em que pese a referência utilizada (7), adotamos integralmente o conceito proposto, sem qualquer ressalvas.

**Comando e Controle:** “Comando e Controle (do inglês: Command and control), ou C2, pode ser definido como o processo de direção por pessoa ou autoridade legalmente ou legitimamente investida na utilização dos recursos colocados à disposição.” (1)

Cumpramos reforçar que a teoria do Comando e Controle–C2 foi totalmente enfocada nos conceitos trazidos pelo Cel John Boyd, abordados adiante.

**Sistemas de Comando e Controle:** “Um SisC2 é um conjunto de instalações, de equipamentos e de comunicações, regido por princípios,

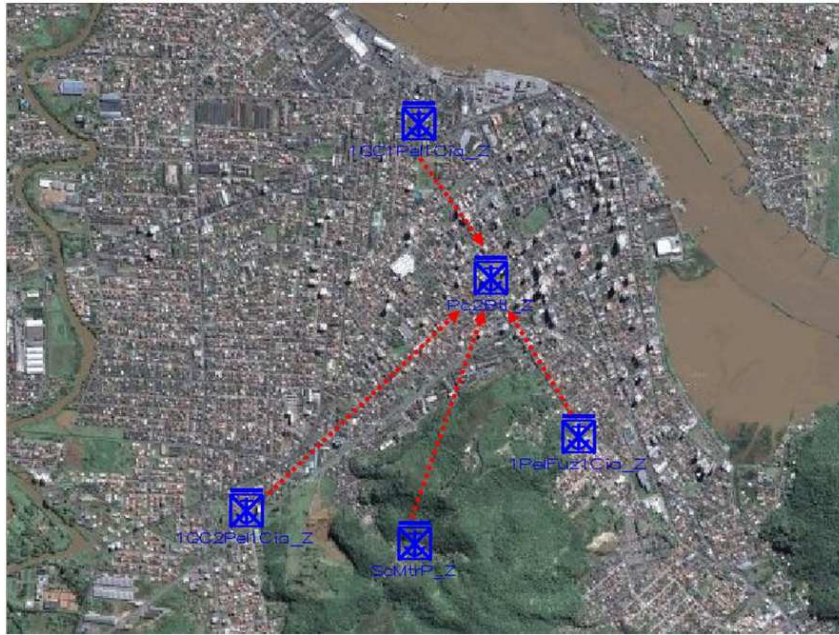


Figura 2.1: Tela de uma estação monitoradora

normas, processos e operado por pessoal especializado e utilizados em operações militares com o objetivo de contribuir para uma atuação eficiente das forças envolvidas em combate.”(7)

Cabe, mais uma vez, a extrapolação da definição, do âmbito militar, para qualquer outro ambiente que, por características afins, também possua a necessidade de implantar sistemas correlatos com a finalidade de melhor gerir um determinado recurso.

A Figura 2.1, esboça a ideia geral do Sistema de Apoio ao Comando e Controle na ocasião em que vários elementos distribuídos no terreno são monitorados por uma estação monitoradora – “PC2BTL\_Z”.

### Ciclo de *Boyd*:

Antes mesmo de definirmos o ciclo de *Boyd*, é interessante conhecermos, resumidamente, a história de seu criador e como se deu a sua formulação. Desta forma acredito ser mais fácil a compreensão do conceito.

Durante os anos 1950, John Boyd era muito conhecido na aviação de caça nos EUA. Seu apelido era “Forty-Second Boyd” ou “Boyd Quarenta Segundos”, porque ele desafiava outros pilotos em combates aéreos de 40 segundos contra o seu Sabre F-100, oferecendo US\$ 40 caso fosse derrotado.

Como instrutor da *Fighter Weapons School* (FWS), Boyd lutou contra pilotos da *USAF*, dos *Marines* e da *US Navy* e contra pilotos de outros países, que recebiam treinamento pelo *Defense Assistance Pact*. Mas nunca foi vencido.

Boyd começava os combates dando vantagem aos seus oponentes, que iniciavam em sua posição 6h<sup>1</sup>. Começado o combate, Boyd conseguia “frear” seu F-100 no ar, aplicando a manobra que ele batizou de “*flat-plating the bird*”, obrigando o oponente a ultrapassá-lo (*overshoot*), finalizando o desafiante com uma rajada certeira e com “*guns, guns, guns*” no rádio. O mito do “*Forty-Second Boyd*” irritou por muito tempo os pilotos de caça.

Boyd começou tarde na Guerra da Coréia, onde só conseguiu avariar um MiG-15 em combate. Mas ele era considerado o melhor no seu esquadrão e nos vôos durante o conflito, aprendeu a razão da vantagem do F-86 sobre o MiG-15.

Os pilotos americanos dominaram os céus da Coréia: a “*kill rate*” nos últimos meses da guerra chegou a 14:1 a favor do F-86 e no final 792 MiGs foram abatidos, para 78 jatos F-86 Sabres. A tática dos pilotos americanos na Coréia era basicamente a do P-38 na Segunda Guerra, ou seja, eles mergulhavam sobre as formações de caças inimigas e evitavam entrar em “*dogfight*” com os MiG.

John Boyd posteriormente debruçou-se sobre a questão dos Sabre versus MiG-15, pois ficou perplexo pelo fato de que no papel o MiG-15 era superior ao F-86. Então qual a razão do F-86 ter sido tão superior na prática? Boyd reconheceu que o “*canopy*” em bolha do Sabre dava uma melhor consciência situacional (SA – *Situation Awareness*) aos pilotos americanos e os controles totalmente hidráulicos lhes permitiam uma transição de manobras ofensivas para defensivas mais rápida que os pilotos soviéticos. Uma capacidade de observação melhor e uma agilidade maior foram as chaves para o sucesso dos pilotos de Sabre.(5)

A partir desta conclusão, Boyd formulou o ciclo da Decisão que hoje em dia também é conhecido pelo nome do seu criador – ciclo de *Boyd*, que consiste na sequência : Observação – Orientação – Decisão – Ação (OODA) obrigatória em um processo decisório.

Para expor tal teoria, John Boyd realizou uma série de palestras (“*A Discourse on winning and Losing*”), dentre as quais a mais importante foi “*Patterns of Conflict*”. Podemos resumir a essência de sua teoria da seguinte forma:

“Um conflito pode ser visto como uma competição no tempo de ciclos OODA. Cada parte, nesse conflito, é iniciada com a observação da situação, considerando-se ela mesma, o ambiente que a cerca e o seu inimigo. Ao observar, busca orientar-se, o que significa formar uma

<sup>1</sup>Em sua retaguarda

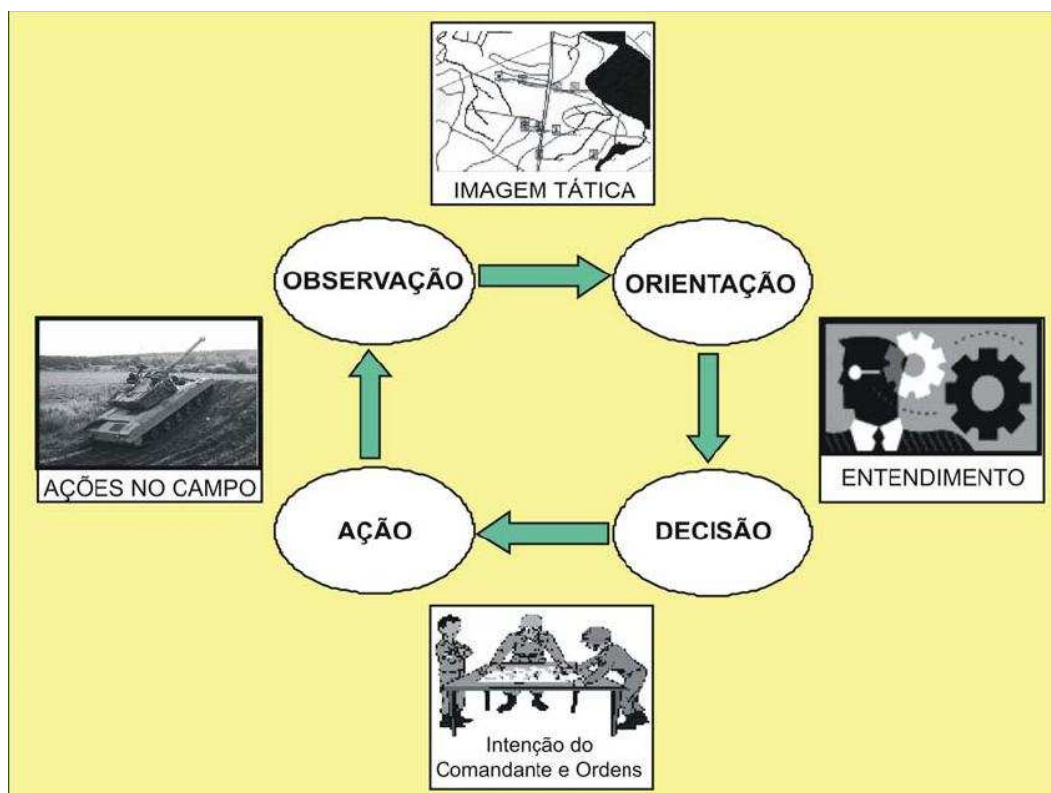
Figura 2.2: Ciclo de *Boyd*

imagem mental da situação. Ao se orientar, monta os elementos para sua tomada de decisão. Em seguida, põe em prática sua decisão, ou seja, executa as ações necessárias. A partir daí, assumindo que suas ações mudaram a situação, observa, novamente, reiniciando o ciclo. Tal processo é conhecido como ciclo de Boyd ou ciclo OODA. Se um dos lados em conflito puder, continuamente, executar tal processo mais rápido que seu oponente, ganhará uma enorme vantagem temporal, de forma que, quando o outro lado puser em prática sua decisão, tais ações parecerão inapropriadas, dadas as novas ações já realizadas pelo lado de ciclo mais veloz. A cada processo realizado, as ações do oponente mostrar-se-ão mais inapropriadas e seu ciclo tornar-se-á cada vez mais lento, até o momento em que suas ações não surtirão mais efeito.”(6)

O Ciclo de Boyd, ilustrado na Figura 2.2, é o ponto de partida para o estudo da teoria do C2. Este ciclo representa o próprio processo de comando e controle ao demonstrar que em combate as ações são desenvolvidas na sequência : Observação – Orientação – Decisão – Ação, de uma forma cíclica.(7)

## 2.2

### Desenvolvimento

A ideia que inspirou a construção do presente protótipo foi a introdução de recursos computacionais no complexo arcabouço de um sistema de C2, definido na Seção 2.1, visando auxiliar e acelerar os passos de observação e orientação de um decisor, em consonância com os preceitos criados por Boyd.

A proposta foi construir um *framework* que possibilitasse o monitoramento remoto de quaisquer elementos (desde uma pessoa até um veículo, trens, tratores, balsas e etc) a fim de exibir, em uma tela, em tempo interativo, informações como posição, velocidade, direção e outras de interesse, conforme o caso de uso a que se destinar. Além disso, foram criadas funcionalidades que proporcionassem informações adicionais que contribuíssem com a formação de uma imagem mental adequada, pelo decisor, a fim de que este se orientasse da melhor maneira possível antes de tomar alguma decisão.

Antes de iniciarmos a análise técnica sobre os passos necessários à construção do Sistema de Apoio ao Comando e Controle, esboçaremos, na próxima seção, as ferramentas de *software* utilizadas.

### 2.2.1

#### Ferramentas Utilizadas

O trabalho foi desenvolvido em versões para as plataformas Linux (Ubuntu 9.1) e MS Windows (Windows XP-SP3), o que permite boa flexibilidade de instalação para seus aplicativos.

A escolha pelo desenvolvimento em uma plataforma Linux deveu-se, principalmente, aos seguintes fatores:

- **Baixo custo.** O Linux é baseado em software livre. Por isso, as empresas que distribuem o sistema não cobram pelo sistema em si, mas por serviços adicionais como publicação de manuais, suporte, treinamento, etc.
- **Segurança.** Qualquer instalação ou alteração do sistema, no Linux, requer a autorização do “super usuário”, que é uma espécie de administrador do sistema. Com isso, dificilmente um vírus ou programa malicioso será instalado a não ser que você autorize, inserindo sua senha e nome de usuário.
- **Comunidade de suporte ativa.** Fóruns, listas de discussão, wikipages, e-groups, sites... É grande a lista de recursos disponíveis na Web para resolver problemas relativos à Linux. De problemas básicos (como dificuldade para executar um tipo de arquivo) a dúvidas mais complexas,

é possível encontrar respostas em comunidades de usuários de software livre online.(8)

No entanto, o Linux apresenta a desvantagem de ser incompatível com alguns equipamentos devido ao fato destes não possuírem os “drivers”(proprietários) de instalação para esta plataforma. Este inconveniente acabou gerando a necessidade do desenvolvimento na plataforma MS Windows.

O banco de dados escolhido foi o PostgreSQL por possuir as características a seguir descritas:

- O PostgreSQL é um banco maduro, com um histórico de desenvolvimento iniciado em 1977;
- Recuperação Automática após crash de sistema (WAL);
- MVCC (controle de concorrência de multi-versão). Neste mecanismo, processos de leitura não bloqueiam processos de escrita e vice-versa, reduzindo drasticamente (às vezes, eliminando) a contenção entre transações concorrentes e paralisação parcial ou completa (deadlock);
- *Constraints / Foreign Keys*;
- *Backup On-line*;
- Tamanho ilimitado de registro: não impõe limites no tamanho de armazenamento dos tipos de dados. Com a arquitetura TOAST (*The Oversized-Attribute Storage Technique*), os únicos limites de armazenamento de tipos de dados são impostos pela configuração do *hardware*. Se um tipo de dado não se ajustar ao tamanho do bloco, atributos únicos podem ser automaticamente comprimidos ou movidos, ou ainda comprimidos fora do registro e armazenados independentemente, dentro de uma tabela relacionada. É possível armazenar registros longos, de qualquer tamanho e com ótima performance.
- O PostgreSQL oferece o mais baixo custo total de propriedade (TCO), reduzindo de forma significativa seus custos de administração, suporte e licenciamento e, ao mesmo tempo, fornecendo alta performance, confiabilidade e escalabilidade.(9)
- o PostgreSQL é um software livre com todas as vantagens que isso acarreta.

Na construção das aplicações específicas, para cada caso de uso tratado, foram utilizadas as ferramentas do TecGraf/PUC-Rio, desenvolvidas através de uma parceria com a PETROBRAS/CENPES a saber:

– IUP–*Portable User Interface*(10)

Utilizado na construção das interfaces gráficas, IUP é uma ferramenta portátil. Oferece configuração de API(*Application Programming Interface* ou *Interface de Programação de Aplicações*) em três linguagens de programação: C, Lua e LED. O seu propósito é permitir que um programa possa ser executado em diferentes sistemas operacionais sem que haja qualquer modificação.

– CD–*Canvas Draw*(12)

Biblioteca gráfica independente de plataforma. Foi implementada em várias plataformas usando as bibliotecas gráficas nativas: Microsoft Windows (GDI) e X-Windows (XLIB).

A biblioteca contém funções que suportam aplicações com imagens ou vetoriais, e a superfície de visualização pode ser uma janela ou uma superfície abstrata, como uma Imagem, Clipboard, Metafile, PS, ou qualquer outra.

– IM–*Image Representation, Storage, Processing and Capture*(13)

IM é uma ferramenta destinada para a manipulação de imagens digitais. É baseada em 4 conceitos: Representação de Imagens, Armazenamento, Processamento e Captura. A principal meta da biblioteca é prover uma API simples e a abstração de imagens para aplicações científicas.

Os formatos de arquivos mais comuns suportados são: TIFF, BMP, PNG, JPEG, GIF e AVI.

A representação de imagens incluem tipo de dados científico. Cerca de uma centena de operações de processamento de imagens estão disponíveis.

Por fim, a linguagem de programação selecionada foi Lua. Segue-se algumas das suas características principais:

– Linguagem de Programação Lua

Lua é uma linguagem de programação interpretada, imperativa, de script, procedural, pequena, reflexiva e leve, projetada para expandir aplicações em geral, por ser uma linguagem extensível (que une partes de um programa feitas em mais de uma linguagem), para prototipagem e para ser embarcada em software complexos, como jogos. Assemelha-se com Python, Ruby e Icon, entre outras.



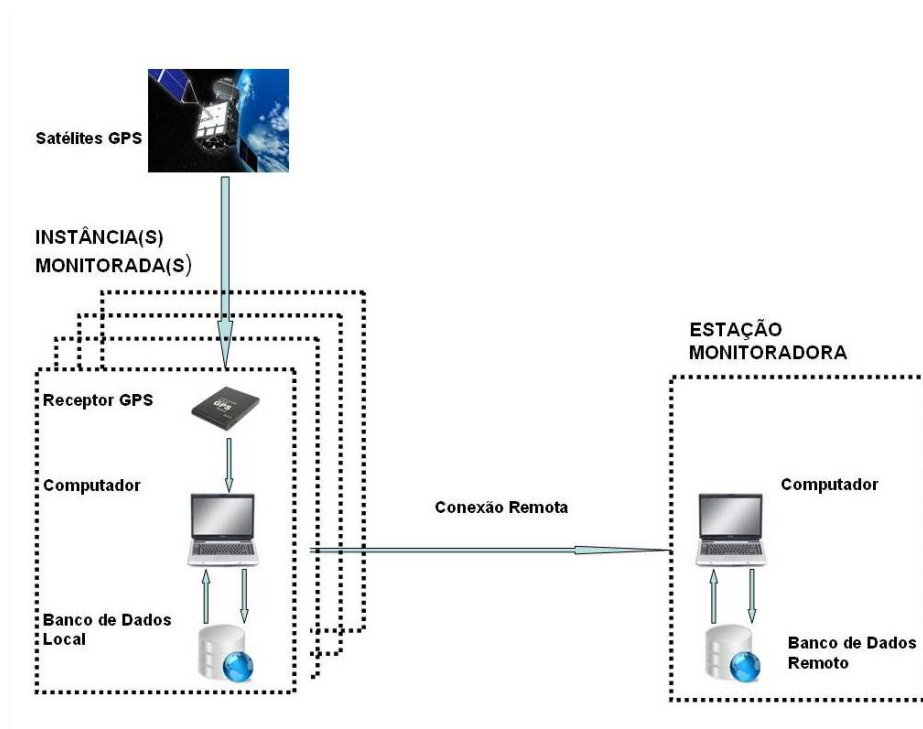


Figura 2.3: Arquitetura do protótipo do Sistema de Apoio ao Comando e Controle

Lua foi criada por um time de desenvolvedores do Tecgraf/PUC-Rio, a princípio, para ser usada em um projeto da Petrobras. Devido à sua eficiência, clareza e facilidade de aprendizado, passou a ser usada em diversas áreas, como no desenvolvimento de jogos (a LucasArts, por exemplo, usou a linguagem no jogo “Escape from Monkey Island”), controle de robôs, processamento de texto, etc. Também é frequentemente usada como uma linguagem de propósito geral.

Lua combina programação procedural com poderosas construções para descrição de dados, baseadas em tabelas associativas e semântica extensível. É tipada dinamicamente, interpretada a partir de bytecodes, e tem gerenciamento automático de memória com coleta de lixo. Essas características fazem de Lua uma linguagem ideal para configuração, automação (scripting) e prototipagem rápida.<sup>(14)</sup>

### 2.2.2

#### Visão Geral da Arquitetura

Para facilitar o entendimento do *framework* desenvolvido é importante que se mostre, em um primeiro momento, a sua arquitetura para, a partir deste ponto, discorrermos sobre cada passo em particular. A Figura 2.3 demonstra uma visão geral da estrutura proposta.

A ideia foi escrever um *script* capaz de realizar o *parsing* (análise sintática da *string* padrão NMEA – *National Marine Electronics Association*) recebida dos Satélites que compõem o sistema GPS, através de um receptor de sinal GPS e de um PC (*Personal Computer*), carregados por uma instância a ser monitorada.

Dado esse passo, o próximo foi inserir em um banco de dados local e em um banco de dados remoto (fisicamente posicionado, preferencialmente, na estação monitoradora) as informações de interesse advindas do *parsing* do sinal recebido do GPS, anteriormente.

Cumpra observar que a inserção no banco de dados remoto se dá via conexão remota, por um modem 3G, ou qualquer outra forma de conexão remota (wi-fi, rádio, etc), estabelecida entre o PC da instância monitorada e o da estação monitoradora, vide a Figura 2.3.

Através de uma aplicação específica desenvolvida, há a leitura das informações do banco de dados e, desta forma, o posicionamento da instância é obtido, georeferenciado em uma carta, tanto pela instância monitorada quanto pela estação monitoradora. No capítulo 3 abordaremos em detalhes as aplicações construídas para atenderem aos propósitos do *framework*.

A partir deste ponto, o tratamento dado ao conjunto de informações (dados diversos inseridos no banco de dados e posicionamento georeferenciado da instância) torna-se específico de acordo com os interesses da atividade que se utiliza do sistema. As funcionalidades implementadas nas aplicações também devem ser específicas de cada segmento.

### 2.2.3

#### Instância Monitorada

Na Figura 2.3, podemos verificar a existência de dois blocos distintos, o primeiro denominado Instância Monitorada e o segundo Estação Monitoradora. Trataremos, nesta seção, dos passos consignados para a construção do primeiro bloco, ilustrado na Figura 2.4.

O presente bloco tem, como entrada, uma sentença com o padrão de formato de dados NMEA 0183, captada através do GPS *receiver* e enviada ao computador da instância monitorada.

A NMEA se auto-intitula a força unificadora por trás da indústria de eletrônica marinha, buscando aglutinar diversos aspectos ligados ao setor. O padrão NMEA 0183 define, dentre outros, características elétricas do sinal,



Figura 2.4: Instância Monitorada

protocolo da transmissão de dados, sincronismo e formatos específicos de sentenças de transmissão.

Todas as sentenças de NMEA consistem em até 82 caracteres em ASCII. Cada sentença começa com o símbolo \$ e termina com um *carriage return* e o consequente começo de nova linha. Todos os campos de dados são delimitados por vírgulas (,) e têm comprimento variável. Os campos nulos também são delimitados por vírgulas (,) mas não contêm nenhuma informação. O primeiro campo de dados é um campo de endereço, ao passo que o último campo de dados é um controle de soma (*checksum*).

O último campo de dados de uma mensagem NMEA 0183 termina com um asterisco (\*) delimitador e um valor de controle de soma (*checksum*). Este último é exclusivo de 8-bit ou de todos os caracteres na mensagem, incluindo as vírgulas entre os campos, mas sem incluir o \$ e os delimitadores com asterisco \*. O resultado, hexadecimal, é convertido a dois caracteres ASCII (0-9, A-F). O caracter mais significativo aparece primeiro.

Existem três tipos de campos de endereço: aprovado, de busca, e proprietário. No presente trabalho, os dados julgados de interesse foram extraídos do primeiro (aprovado). Endereços aprovados têm a estrutura \$TTFFF, onde:

- TT = identificador de transmissão
- FFF = formatador da sentença

Exemplo: \$GPGGA

Abaixo, incluímos alguns exemplos de formatos aprovados da sentença NMEA:

- ALM = dados do almanaque GPS
- DTM = datum de referência
- GBS = detecção de falha de algum satélite GNSS
- GGA = solução de posição determinada por GPS
- GNS = solução de posição determinada por sistemas de posicionamento por satélites
- GRS = resíduos em distância de observações oriundas de sistemas de posicionamento por satélites.
- GSA = DOP e satélites ativos
- GST = estatística de erro das observações de pseudo-distância
- GSV = satélites “visíveis”
- HDG = orientação, declinação e variação da declinação.
- HDT = orientação, verdadeira Heading, true
- RMC = trânsito de dados GPS mínimo recomendado
- VTG = rumo percorrido
- XDR = medições oriundas de transdutores
- ZDA = tempo e data (15)

Os formatos selecionados das sentenças NMEA foram o RMC, GGA e o GSA por possuírem todas as informações necessárias à construção das aplicações desenvolvidas no *framework*. A partir daí, através de um *script*, chamado de “gps.lua”, é realizado o *parsing* onde os dados de interesse são extraídos e armazenados em uma tabela Lua.

Segue-se alguns exemplos de *strings* dos formatos RMC, GGA e GSA com os respectivos *parsings*:

– **\$GPRMC**

*String* recebida:

\$GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E\*68

*Parsing*:

225446 = Time of fix 22:54:46 UTC

A = Navigation receiver warning A = OK, V = warning

4916.45,N = Latitude 49 deg. 16.45 min North

12311.12,W = Longitude 123 deg. 11.12 min West

000.5 = Speed over ground, Knots

054.7 = Course Made Good, True

191194 = Date of fix 19 November 1994

020.3,E = Magnetic variation 20.3 deg East

\*68 = mandatory checksum

#### – \$GPGGA

*String* recebida:

\$-GGA,081014.14,7345.23,N,11103.15,W,1,2,2.5,10.5,M,3.3,M,3.4,0011

*Parsing:*

081014.14 = UTC of position

7345.23 = latitude of position

N = N or S

11103.15 = Longitude of position

W = E or W

1 = GPS Quality indicator (0=no fix, 1=GPS fix, 2=Dif. GPS fix)

2 = number of satellites in use

2.5 = horizontal dilution of precision

10.5 = Antenna altitude above mean-sea-level

M = units of antenna altitude, meters

3.3 = Geoidal separation

M = units of geoidal separation, meters

3.4 = Age of Differential GPS data (seconds)

0011 = Differential reference station ID

#### – \$GPGSA

*String* recebida:

\$GPGSA,A,3,19,28,14,18,27,22,31,39,,,,,1.7,1.0,1.3\*35

*Parsing:*

1 = Mode: M=Manual, forced to operate in 2D or 3D A=Automatic, 3D/2D

2 = Mode: 1=Fix not available 2=2D 3=3D

- 3-14 = IDs of SVs used in position fix (null for unused fields)
- 15 = PDOP (positional-3D, dilution of precision)
- 16 = HDOP (horizontal dilution of precision)
- 17 = VDOP (vertical dilution of precision)(16)

Convém ressaltar que, em relação à latitude e à longitude, antes da inserção dos dados na tabela, estes são convertidos do padrão de graus decimais para o padrão de coordenadas UTM (*Universal Transverse Mercator*) através de funções específicas. Tal tratamento visa facilitar o passo do georeferenciamento, realizado pelas aplicações específicas.

A taxa de inserção de dados no banco, como descrito anteriormente, está condicionada à velocidade corrente da instância monitorada, de acordo com as faixas ilustradas na tabela 2.1. A implementação do filtro é justificada pela ideia de otimizar a alocação de espaço no banco de dados sem que haja perda na qualidade das informações. Assim, por exemplo, se uma instância monitorada estiver parada (  $veloc. = 0$  ) não seria necessário enviar para o banco de dados, a uma taxa elevada, o seu posicionamento pois, se assim o fosse, estaríamos duplicando informações no banco de dados desnecessariamente. No entanto, se há um deslocamento com uma velocidade alta, é razoável que se atualize a informação sobre o posicionamento com uma maior rapidez a fim de evitar grandes *gaps* sobre este dado.

Tabela 2.1: Filtro para inserção de dados no banco de dados

Velocidade (Km/h)	Taxa de Inserção no Banco de Dados (s)
$veloc. < 10$	25
$30 > veloc. \leq 10$	20
$60 > veloc. \leq 30$	15
$100 > veloc. \leq 60$	10
$100 < veloc.$	5

Finalmente, a tabela contendo os dados de interesse, oriundos do *GPS Receiver*, é inserida em um banco de dados( *postgreSQL* ) local e, logo após, em um banco de dados remoto, valendo-se de uma conexão de rede, procedimento explanado a seguir.

## 2.2.4

### Inserção no Banco de Dados

A operação de inserção realizada no banco de dados utiliza uma *Application Programming Interface – API* construída para atender as necessidades do sistema implementado para a simulação de operações militares do Centro de

Jogos Didáticos – CJD, do Corpo de Fuzileiros Navais da Marinha do Brasil. Este projeto é desenvolvido em parceria com o TecGraf/PUC–Rio desde 1997.

A operação realizada no módulo “gps.lua” consiste, basicamente, em intercalar a atribuição de uma variável de *Database Source Name* – *DSN*, entre local e remoto quando se deseja popular o banco da instância local e o da estação monitoradora.

Todas as instruções para a criação das conexões com os bancos e demais procedimentos em *Structured Query Language* – *SQL* estão imbutidas na referida API.(17)

## 2.2.5

### Aplicações Específicas

Neste momento não nos prenderemos a detalhes acerca da construção das aplicações que manipulam os dados inseridos nos bancos de dados. No entanto, é importante ressaltar que, de acordo com o previsto na Figura 2.3, a mesma aplicação instalada no computador da instância monitorada deverá estar igualmente disponível para a estação monitoradora.

A construção das aplicações será vista, com uma abordagem pormenorizada, nos capítulos que tratam sobre os casos de uso para o *framework* (Militar e Monitoramento e controle de navegação).

## 2.3

### Hardware

Nesta seção trataremos das especificações desejáveis para os equipamentos componentes do Sistema de Apoio ao C2, detalhando cada um dos componentes encontrados no mercado que, na ocasião em que se desenrolou a busca, preenchiam os requisitos levantados em melhores condições.

### 2.3.1

#### Especificações de Hardware

Tendo em vista que o Sistema de Apoio ao C2 tem, por formulação, os objetivos elencados no capítulo 1, seção 1.2, um dos requisitos fundamentais ao seu êxito seria ter, como componentes de *hardware*, para a instância monitorada, mormente, dispositivos que aglutinassem as seguintes características:

**Leveza:** uma das possibilidades de emprego prevista é o monitoramento de pessoas em deslocamento a pé. Desta forma, quanto mais leve os dispositivos componentes do bloco ilustrado na Figura 2.4, menor será o desgaste relativo ao transporte dos itens.

**Portabilidade:** uma vez que haverá deslocamentos com os dispositivos, não seria viável, por exemplo, a utilização de equipamentos excessivamente pesados ou que não dispusessem de fontes de alimentação portáteis, dependendo, desta forma, de fontes externas.

**Autonomia:** considerando-se que, principalmente em deslocamentos, dependendo do caso de uso, nem sempre será possível contar com fontes externas, torna-se bastante interessante a obtenção de equipamentos que aliem baixo consumo à baterias recarregáveis e com boa capacidade de armazenamento de energia.

**Resistência à intempéries:** levando-se a efeito a possibilidade de deslocamentos dos dispositivos, sem que estes estejam abrigados, é conveniente que os *hardwares* sejam resistentes a chuva e a temperaturas elevadas.

**Capacidade computacional e de armazenamento de dados:** por imposição da arquitetura do sistema, há a necessidade de uma aplicação executada pela instância monitorada. Assim sendo, existem as demandas de armazenamento e leitura de imagens, instalação de um banco de dados bem como o processamento exigido pelas diversas funcionalidades desenvolvidas em conformidade com o caso de uso.

### 2.3.2

#### Soluções de Hardware encontradas

Levantadas as necessidades, iniciamos uma pesquisa cuidadosa para a obtenção dos equipamentos que mais se aproximassem das características descritas. Cabe ressaltar que os *hardwares* encontrados permitem configurar a instância monitorada tanto no ambiente *Windows* quanto no *Linux*. Os resultados obtidos serão vistos a seguir.

##### – *GPS Receiver–I-gotU*

Como notamos na tabela A.1, todos os requisitos previstos em 2.3.1 são plenamente satisfeitos. Convém ressaltar que como o *IgotU* possui memória interna de 64Mb, há um *backup* de todas as posições demandadas pela instância monitorada, até o limite de 262.000. Com esta configuração, o Sistema de Apoio ao C2 conta com três *backups* de posições: memória interna do IgotU, banco de dados local e banco de dados remoto.

##### – *Personal Computer – fit-PC2*

O *fit-PC2* foi uma das gratas surpresas da pesquisa. As impressionantes características de dimensões e peso reduzidos somadas à boa capacidade





Figura 2.5: GPS Receiver



Figura 2.6: fit-PC2



Figura 2.7: fit-PC2

computacional, baixo consumo (96% menor que um *desktop* convencional (19)), possibilidade de conexão via rede *wireless*, facilitaram sobremaneira a implementação do Sistema de Apoio ao C2. É oportuno observar, no entanto, que o equipamento deve ser protegido de eventos como a exposição à água ou quedas bruscas.

– **Modem – Claro–HUAWEI**



Figura 2.8: Modem Claro 3G

As dimensões e o peso do modem são desprezíveis. Convém ressaltar que poderiam ser utilizados outros dispositivos para conexão com a rede, como, por exemplo, uma conexão via rede *wireless*. Neste caso não seria necessária a utilização do Modem 3G.

A opção por utilizar o Modem 3G foi em função de ser esta tecnologia a única que permitiria a implementação do *framework* de forma tempestiva.

– **Bateria – Tekkeon**



Figura 2.9: Bateria Tekkeon

A bateria adotada, além de cumprir satisfatoriamente as especificações levantadas na seção 2.3.1, possui ainda a possibilidade de recarga por meio de painel solar. Tal possibilidade não foi testada no presente trabalho em função de não ser escopo, neste momento, refinamentos de *hardware*, sendo esta, no entanto, uma tarefa incluída como trabalhos futuros.

### – Monitor – MIMO



Figura 2.10: Monitor utilizado para a instância monitorada – MIMO

O monitor utilizado atende perfeitamente ao *framework*. Devido ao fato de possuir a característica *touchscreen*, não demanda periféricos adicionais como *mouse* ou teclado. Entretanto, a inexistência, até o momento, de *Driver* para a plataforma Linux, o deixa restrito ao sistema operacional *Windows*.

### – Display de LCD

Foi desenvolvido pelo TecGraf/PUC-Rio um protótipo de tela de LCD a fim de disponibilizar informações de texto para a instância monitorada. O display possui dimensões e peso bastante reduzidos sendo alimentado através de uma porta USB. Dados como posição, velocidade e direção podem ser facilmente exibidos através deste dispositivo.

Um aspecto relevante é a possibilidade de utilização do display tanto no ambiente Windows quanto no Linux. O display pode ser visto na Figura 2.11.

### – Ítems acessórios

Como o fit-PC2 possui entrada para microfone, áudio e diversas portas USB, estabelecemos um canal de comunicações, por áudio e/ou video, entre a estação monitoradora e a instância monitorada, através do Skype, MSN ou Ekiga – aplicações de *software* que permitem chamadas de voz e video através da internet.

Desta forma foi empregado, como acessórios, uma *web camera* e um *headset*.

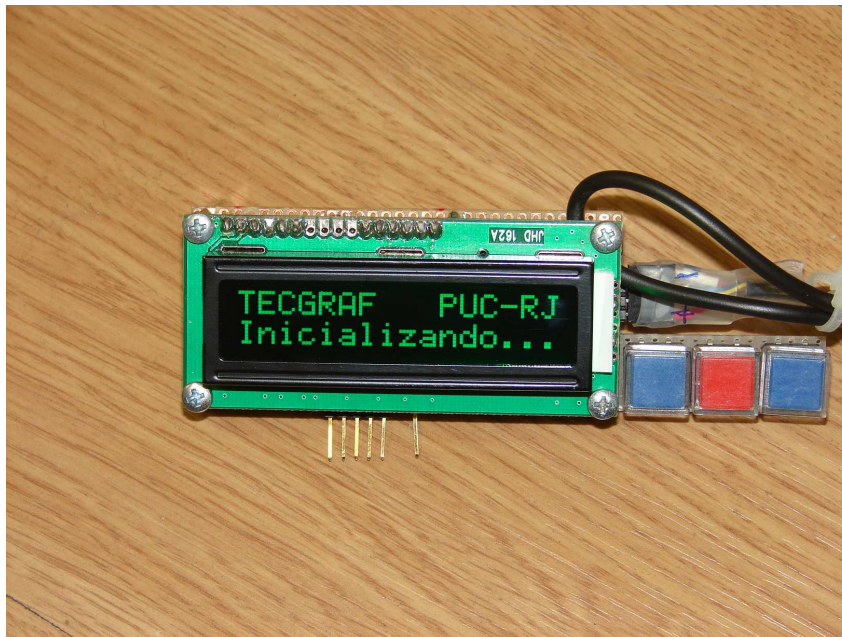


Figura 2.11: Display de LCD desenvolvido pelo TecGraf/PUC-Rio

### 2.3.3

#### Custo Estimado

Um aspecto relevante para a viabilidade da implementação efetiva do *framework* é, sem dúvida, o custo dos *Hardwares* envolvidos. Desta forma, ainda que este ponto não tenha sido considerado como fator principal na escolha dos equipamentos, é importante tratarmos sobre esta questão. A tabela 2.2 explicita os valores dos principais componentes do Sistema de Apoio ao C2 (o câmbio utilizado para a conversão de USD para Real foi de 1,70). O custo do Modem 3G Claro não foi incluído na tabela tendo em vista que o aparelho é disponibilizado gratuitamente pela operadora. O custo envolvido neste caso é de uma mensalidade que tem seu valor variando em função de aspectos como velocidade da conexão e quantidade de dados baixados.

Tabela 2.2: Custo dos principais componentes do Sistema de Apoio ao C2

fit-PC2	USD 398,00	R\$ 676,60
Monitor MIMO	USD 179,99	R\$ 305,98
Bateria Tekkeon	USD 139,99	R\$ 237,98
GPS IgotU	USD 95,99	R\$ 163,18
Total	USD 813,97	R\$ 1383,74