



FEUP

**Sistema de Planeamento e Controlo de Execução para Veículo de
Superfície Não Tripulado**

Relatório Final

Equipa E

Orientador João Sousa

Líder da Equipa Carlos Pinto

Projeto desenvolvido do âmbito da unidade curricular
Sistemas de Engenharia - Automação e Instrumentação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Registo de Versões

Versão	Data	Descrição
0.1	21/12/2020	Criação do documento base segundo o template definido
0.2	21/12/2020	Definição da estrutura do documento
0.3	12/01/2021	Validação e Correção da estrutura do documento

Abreviaturas e Acrónimos

AIS	<i>Automatic Identification System</i>
GCS	<i>Ground Control Station</i>
GPS	<i>Global Positioning System</i>
IO	<i>Inputs and Outputs</i>
KPI	<i>Key Performance Indicator</i>
LoRa	<i>Long Range protocol</i>
LSTS	Laboratório de Sistemas e Tecnologia Subaquática
NMEA	<i>National Marine Electronics Association data format</i>
PWM	<i>Pulse Width Modulation</i>
SBS	<i>System Breakdown Structure</i>
UMAHS	<i>Unmanned Maritime Autonomous System for Hydrographic Studies</i>
USV	<i>Unmanned Surface Vehicle</i>
WBS	<i>Work Breakdown Structure</i>

Sumário Executivo

Serve o relatório para apresentar o projeto desenvolvido pela equipa E no âmbito da unidade curricular de Sistemas de Engenharia - Automação e Instrumentação: um Sistema de Planeamento e Controlo de Execução para Veículo de Superfície Não Tripulado. O projeto foi desenvolvido em conjunto com a Marinha Portuguesa, a quem a embarcação X-2601 pertence, e com auxílio do Laboratório de Sistemas e Tecnologia Subaquática (LSTS).

O foco do trabalho consistiu, numa primeira fase, na interiorização do papel do USV na frota da Marinha e das implicações disso para os requisitos do projeto. Esse enquadramento, traduz-se na secção 1 de apresentação do X-2601 e o desafio de o tornar autónomo e integrar na plataforma de gestão integrada (Secção 2) de outros veículos da Marinha, face ao já existente no mercado. Ainda nessa secção, faz-se um resumo da arquitetura de hardware e software do USV, antes de qualquer desenvolvimento da equipa.

Seguindo a linha do processo de integração, a secção 3 apresenta o software de base a usar (DUNE e Neptus, integrantes da LSTS toolchain), explicitando de que forma são úteis no desafio proposto, tanto a nível de controlo de baixo nível, como na monitorização e controlo de alto nível. O DUNE é o software de bordo e desempenha um papel importante devido às suas funcionalidades já implementadas no que diz respeito a controlo de veículos marítimos, bem como a sua modularidade e escalabilidade. O Neptus constitui a interface do utilizador, para que possa planejar e enviar missões, e apresenta facilidade de integração com a embarcação em questão e tem já sido usado por outros veículos da marinha, sendo esta interoperabilidade já inerente à sua utilização.

A secção 4 explicita o alvo de desenvolvimento: software do lado da GCS (usando Neptus) e do sistema a bordo (usando DUNE) usando um diagrama SBS. O mapeamento do Software no Hardware, bem como as ligações entre si (subsecção 4.2) ajuda a uma melhor compreensão das várias partes tanto como módulos individuais como parte de um sistema único que precisa de sensorização, atuação, comunicação e controlo. No fim, pretendeu-se dotar o utilizador de capacidades de comandar o veículo, planejar uma missão e monitorizar o USV. Tudo isto é possível pela troca de diversas mensagens, que seguem protocolos bem definidos, como NMEA e IMC (ver subsecção 4.4).

A parte de validação experimental do trabalho utilizou uma abordagem bottom-up. Para isso, foram realizados testes unitários sucessivos. Os testes de sensorização incluíram teste com simulador de mensagens NMEA e hardware LT1000. A validação da atuação focou tanto o envio de valores em toda a gama especificada pelos módulos de lemes e motores, como na análise dos sinais de controlo PWM. Terminando os testes individuais, procedeu-se à realização de manobras de varrimento (útil para estudos hidrográficos), espiral e paragem para profundidades baixas, sendo esta última implementada de raiz pela equipa. As simulações correram no software Neptus, com um modelo de catamarã já existente e dimensões corrigidas para o X-2601. Para a simulação do sistema integrado, utilizaram-se lemes emulados por servos e LEDs que variam de cor consoante a direção e a velocidade. Os resultados permitiram analisar quais os requisitos efetivamente cumpridos, verificando-se que apenas a parte de comunicação ficou incompleta, justificando-se devidamente as dificuldades decorridas e soluções tomadas.

A terceira parte do relatório descreve os pontos relativos à gestão de projeto, nomeadamente equipa, estruturação do trabalho, divisão temporal de tarefas, orçamento, detalhes de execução e gestão de riscos.

Os anexos contêm os documentos entregues no decorrer da cadeira: Manual de Qualidade, Análise de Mercado, Lista de Requisitos e Conceito do Sistema.

Os vídeos das várias demonstrações, documentação específica de cada módulo desenvolvido e outros documentos podem ser consultados na página da equipa, em <https://sites.google.com/g.upporto.pt/seai-equipae>.

Índice

Lista de Tabelas	6
Lista de Figuras	7
I Sistema desenvolvido	8
1 Enquadramento do Projeto X-2601	8
1.1 Documentação Produzida	8
1.2 Organização do relatório	9
2 Descrição do Desafio	10
2.1 Pesquisa de mercado	10
2.2 Desafio proposto	10
2.3 Apresentação do veículo	10
2.4 Arquitetura de Software Original	11
2.4.1 <i>On-Board</i>	12
2.4.2 Software Instalado	12
2.5 Objetivos a cumprir	13
3 Estudo da Aplicação da LSTS Toolchain do Projeto	15
3.1 Papel do DUNE no sistema	15
3.1.1 Ficheiro de configuração	15
3.2 Papel do Neptus no sistema	16
3.2.1 Ficheiro de configuração	16
4 Conceito do Sistema	17
4.1 SBS	17
4.2 Arquitetura Funcional e Mapeamento em Dispositivos Computacionais	17
4.3 Casos de Uso do Sistema	19
4.4 Diagramas de Fluxo de Mensagens	19
5 Trabalho realizado	21
5.1 LT1000	21
5.1.1 NMEA	23
5.1.2 Tarefa LT100	24
5.1.3 Simulador de Mensagens NMEA	25
5.2 Atuadores	25
5.2.1 Tarefa SimpleHeadingAndSpeed	25
5.2.2 Tarefa UART_Comm	26
5.3 Comunicações	26
5.3.1 Wifi	26
5.3.2 LoRa	26
5.4 Controlo de Profundidade	27
II Resultados e Análise	28
6 Resultados	28
6.1 Resultados Simulador NMEA(panazzolo.com)	28
6.2 Atuadores	29
6.3 Simulação de Manobras	30

6.3.1	Plano de Varrimento	30
6.3.2	Plano de Trajetória Curva	32
6.3.3	Plano com variação de profundidade	34
6.4	Simulação de Todo o Sistema	36
6.5	Dificuldades e Desvios face ao plano inicial	37
7	Análise do Cumprimento dos Requisitos	40
7.0.1	Requisitos do Cliente	40
7.0.2	Requisitos Funcionais	40
7.0.3	Requisitos de Performance	41
7.0.4	Requisitos de Design	42
7.0.5	Requisitos Alocados	42
8	KPI	43
8.1	KPI's de equipa	44
8.2	KPI's do Projeto	44
III	Gestão de Projeto	46
9	Organização da equipa	46
9.1	Subequipes	47
10	WBS	47
10.1	Distribuição temporal de tarefas	48
11	Orçamento	50
12	Gestão de Riscos	51
13	Metodologia de Teste: Bottom-up Testing	51
14	Execução do Projeto	52
15	Interação com o cliente	53
16	Avaliação de equipa	53
Referências		55
Anexo A	Manual de Qualidade	56
Anexo B	Análise de Mercado	77
Anexo C	Lista de Requisitos	94
Anexo D	Conceito de Sistema	110

Lista de Tabelas

1	Requisitos funcionais	14
2	Requisitos do cliente.	40
3	Requisitos funcionais.	40
4	Requisitos de performance.	41
5	Requisitos de design.	42
6	Requisitos Alocados	42
7	KPI's de equipa	44
8	KPI's do Projeto	45
9	Orçamento estimado do Projeto	50
10	Riscos	51
11	Avaliações de equipa	53

Lista de Figuras

1	Catamarã X-2601	8
2	Competição Principal	10
3	Catamarã X-2601	11
4	Arquitetura Original do Sistema, antes dos desenvolvimentos	11
5	Subsistema On-board e seus componentes	12
6	Software OpenCPN	13
7	Interface Original da GCS	13
8	Interfaces oferecidas pelo Neptus	16
9	System Breakdown Structure	17
10	Estado final do sistema	18
11	Diagrama de Casos de Uso	19
12	Descrição das comunicações trocadas entre as partes do sistema	20
13	LT-1000	21
14	True Heading vs Magnetic Heading	22
15	Roll, Pitch, Yaw. Fonte: [1]	22
16	COG vs SOG. Fonte: [2]	23
17	NMEA 0183 - LT1000 <i>Fonte: Datasheet LT1000[3]</i>	24
18	Simulador Mensagens NMEA	25
19	Módulo LoRa6500Pro	27
20	Valores enviados pelo Simulador de NMEA	28
21	Resultado da leitura dos dados no DUNE	28
22	Sinais de entrada e saída do teste UART	29
23	Manobra de virar para a direita	30
24	Manobra de virar para a esquerda	30
25	Trajetória efetivamente executada (a laranja) em comparação com a trajetória original planeada (a branco).	31
26	Direção desejada ao longo do percurso (azul) em comparação com a direção (proa) ao longo do percurso (vermelho).	31
27	Distribuição temporal a executar cada subcomponente do plano (manobra).	32
28	Trajetória efetivamente executada (a laranja) em comparação com a trajetória original planeada (a branco).	33
29	Direção desejada ao longo do percurso (azul) em comparação com a direção (proa) ao longo do percurso (vermelho).	33
30	Distribuição temporal a executar cada subcomponente do plano (manobra).	34
31	Trajetória efetivamente executada (a laranja) em comparação com a trajetória original planeada (a branco).	35
32	Distribuição temporal a executar cada subcomponente do plano (manobra).	35
33	Batimetria ao longo do percurso	36
34	Configuração utilizada para teste de sistema completo.	36
35	Setup dos atuadores	38
36	LT1000 - estado à chegada do material	39
37	LT1000 - estado pós tentativa de reparação	39
38	Diagrama do Work Breakdown Structure	48
39	Diagrama de Gantt do desenvolvimento do projeto	49
40	Ilustração da metodologia Bottom-up no projeto	52
41	Diagrama do WBS com execução das tarefas	53

Parte I

Sistema desenvolvido

1 Enquadramento do Projeto X-2601

Integrando-se no âmbito da unidade curricular de Sistemas de Engenharia Automação e Instrumentação (SEAI), desenvolveu-se um Sistema de Planeamento e Controlo de Execução para Veículo de Superfície Não Tripulado. Consistiu na integração do veículo X-2601 da Marinha Portuguesa (Figura 1) com o software da toolchain do LSTS.

A unidade curricular de SEAI enquadra-se no último ano curricular do Mestrado Integrado em Engenharia Electrotécnica e de Computadores, procurando proporcionar um contacto mais realista com o universo dos projetos complexo de engenharia.

O projeto a nós atribuído com a Marinha enquadra-se no foco da Célula de Experimentação Operacional de Veículos Não Tripulados (CEOV). Esta célula da Marinha vai ao encontro com a estratégia de investir nos avanços tecnológicos e com a integração de técnicas mais vanguardistas, promovendo uma melhor performance no terreno.

O papel do Laboratório de Sistemas e Tecnologia Subaquática (LSTS) deu-nos a base para a realização do projeto, já que é um laboratório especializado no design, desenvolvimento de veículos de superfície, subaquáticos e aéreos, interligados entre si. O software que desenvolvem foi a base para a interligação do X-2601 na rede de veículos da Marinha.



Figura 1: Catamarã X-2601

1.1 Documentação Produzida

De forma a fazermos escolhas fundamentadas no decorrer do ciclo de vida do projeto, foram elaborados vários documentos. Numa fase inicial, desenvolveram-se

- Manual de Qualidade- estabelece normas e princípios a adotar pela equipa, fomentando as boas práticas de trabalho e um desenvolvimento organizado do projeto proposto;
- Estudo de Mercado- permitiu adquirir uma perspetiva em relação ao mercado dos veículos não tripulados de superfície e das soluções já implementadas;
- Definição de Requisitos- formulação das necessidades do cliente e dos requisitos técnicos que daí advém;
- Conceito do Sistema- estruturação dos componentes integrantes da solução, ilustrando o fluxo de informação entre cada um dos blocos;

- Relatório intermédio- documenta os avanços num ponto intermédio do sistema, com um planeamento mais direto;
- Relatório final- presente documento com a apresentação da solução desenvolvida e aspetos da gestão do projeto, nomeadamente gestão de dificuldades.

1.2 Organização do relatório

O documento está dividido em 3 partes e 16 secções. A primeira parte faz o enquadramento do sistema e descreve o antes e depois da implementação. A segunda parte descreve os testes realizados, as simulações e os resultados obtidos. Ainda se descreve sucintamente as dificuldades enfrentadas e soluções procuradas e de que forma é que os requisitos foram cumpridos. Por fim, descreve-se a gestão do projeto, nomeadamente os assuntos ligados à equipa, riscos, metodologias e estruturação do trabalho.

2 Descrição do Desafio

Ao longo das últimas décadas, os veículos marítimos têm sido uma ferramenta crucial no que toca à protecção e patrulhamento da área costeira nacional. O recurso a veículos autónomos permite assim aumentar a longevidade e precisão das embarcações, projetar missões ininterruptas mais longas e corresponder a situações mais complexas e exigentes, como o caso de situações de catástrofe e emergência. Em relação aos veículos tripulados, os USV são mais dispendiosos no que toca à fase inicial de desenvolvimento do veículo e procedimentos de manutenção. No entanto, esses gastos podem ser contrabalançados pela ausência de gastos com tripulação e aumento de eficiência, nomeadamente combustíveis no caso de um controlador bem projetado.

2.1 Pesquisa de mercado

O mercado dos USV's toca várias áreas, tais como fornecedores de matéria-prima, processadores, fabricantes de produtos finais e organizações regulatórias na cadeia de abastecimento. O lado da procura inclui utilizadores finais, sendo organizações militares ou navegadores individuais. A oferta baseia-se essencialmente no desenvolvimento da embarcação e do software para a tornar autónoma. Como o projeto em questão se foca na automatização do USV e na integração numa rede de veículos, o mercado potencial é constituído por fabricantes deste tipo de veículos ou qualquer embarcação tradicional já existente numa perspetiva de conversão.

Analizando os tipos de aplicação de USV, salienta-se de defesa/patrulhamento da costa, pesquisa científica para estudo da qualidade da água e mapeamento oceanográfico, monitorização ambiental, entre outras.

A principal competição inclui o "ASView Control System" da L3 ASV, o "K-MATE" da Kongsberg Group, o "ME120" da OceanAlpha, o "MANTAS" da MARTAC, "SM300" da Sea Machines e o "Wave Glider" da Liquid Robotics. As suas principais características estão descritas na figura

Competitors	Usage	Type	Key Features
Q-Boat1250 (Teledyne Marine)	Scientific Research	Trimaran	Portable and Affordable
ME120 (OceanAlpha)	Defense	Catamaran	Collision Avoidance
MANTAS (MARTAC)	Scientific Research	Catamaran	High Speed Operation
Wave Glider (Liquid Robotics)	Oceanography	Floating Vessel	Programmable Course

Figura 2: Competição Principal

Uma maior descrição da competição e do mercado pode ser encontrado no documento de Análise de Mercado, do Anexo B.

2.2 Desafio proposto

A partir do catamarã já operacional, o desafio proposto resumiu-se à sua integração com o software do Laboratório de Sistemas e Tecnologia Subaquática (LSTS). Esta alteração permite que, de forma integrada com outros veículos da marinha, se coordene, planeie, controle e monitorize operações. A vantagem da utilização da toolchain[4] prende-se com a expansão de funcionalidades de um veículo já funcional e de operação individual, para uma atuação integrada a baixo custo e implementada num curto horizonte temporal.

2.3 Apresentação do veículo

O catamarã X-2601, representado na Figura 3, pertence à frota da Marinha Portuguesa. O veículo é um sistema marítimo Autónomo Hidrográfico (UMAHS), desenvolvido pela Célula de Experimentação Operacional

de Veículos Não Tripulados (CEOV), que visa ao apoio do Instituto Hidrográfico nos estudos e levantamentos hidrográficos, como rios, estuários e zonas costeiras.



Figura 3: Catamarã X-2601

Para além de permitir o aumento da segurança e alcance dos estudos, o catamarã poderá participar em situações de catástrofe, quando solicitado pela Autoridade Nacional de Emergência e Proteção Civil [5]. Isto obriga a que o USV (*Unmanned Surface Vehicle*) tenha que atingir um grau significativo de autonomia e decisão na navegação, ser robusto ao nível de comunicações e preciso na sua localização. Consideramos que estes é com estes fatores chave bem presentes que devemos abordar o projeto do X-2601.

2.4 Arquitetura de Software Original

Originalmente, a solução implementada consistia num Arduino como peça intermédia do sistema: comunicava com a *Ground Control Station*, era programado (*Flashed*) pelo LattePanda e controlava diretamente os atuadores e sensores.

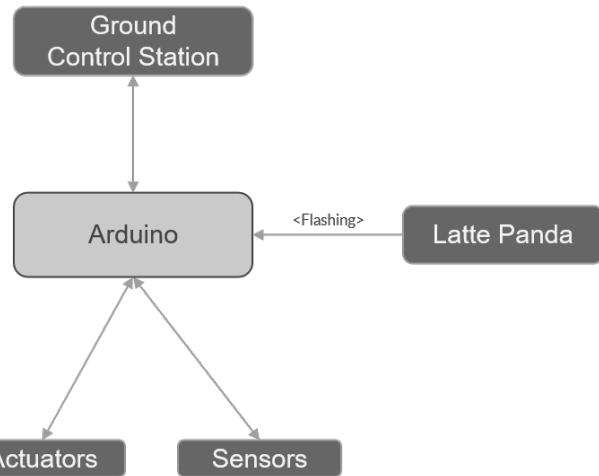


Figura 4: Arquitetura Original do Sistema, antes dos desenvolvimentos

O Arduino, o LattePanda e os periféricos (I/O) encontram-se na embarcação e a *Ground Control Station*

encontra-se em terra, servindo de controlo de alto nível e encarregue da monitorização do estado do sistema. O diagrama que traduz esta arquitetura encontra-se na Figura 4. As funcionalidades da GCS encontram-se resumidas na Figura 7.

2.4.1 On-Board

No USV encontram-se diversos módulos que se utilizaram também na nova arquitetura de forma a manter o correto funcionamento e reutilizar o material, diminuindo os custos da solução. A bordo do X-2601, tal como se visualiza na Figura 5, é possível encontrar:

- Um microcontrolador Arduino Mega [6] responsável por gerir o sistema motor, através de sinais PWM.
- Módulo de comunicações LoRa [7] que funciona a 433MHz e potência de 5W para comunicação com o Arduino. A LoRa (Long Range) é uma tecnologia de comunicação a longa distância e de baixa potência, porém com baixa largura de banda. É comumente usado para telemetria para controlo remoto com alcance até 8Km, sendo adequado para comunicações com o veículo perto da costa, por constituir uma baixa utilização da bateria a par com a simplicidade do protocolo.
- Um módulo AIS (Automatic Identification System) [8] que identifica a embarcação para outras embarcações e para a rede em terra.
- Bússola com GPS integrado na placa LT1000 [3], que fornece informações ao Arduino sobre posição e proa do sistema.
- 2 drivers para os motores [9] controlados por modulação PWM.
- 2 servos[10] para controlo dos lemes.
- O LattePanda [11], um SBC (single board computer) com sistema operativo Windows 10, que se conecta através de um router à internet para controlo e/ou programação remota do Arduino. Este SBC foi adaptado para correr um Sistema Operativo Linux por motivos de fiabilidade, segurança, custo e integração com a plataforma de controlo.

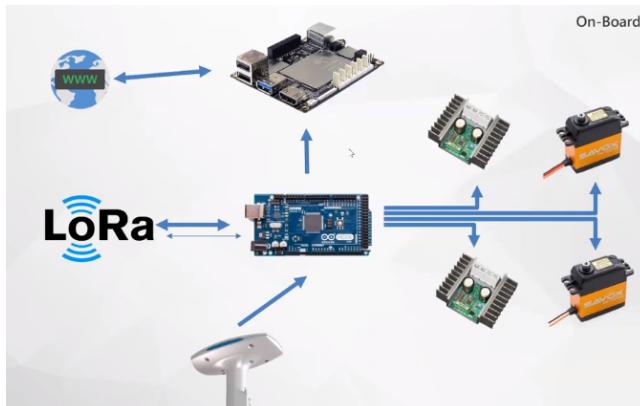


Figura 5: Subsistema On-board e seus componentes

Visto que o principal objetivo do projeto é a integração do sistema na plataforma, todo o Hardware foi incorporado na nova solução.

2.4.2 Software Instalado

Atualmente, o sistema tem algum software instalado: um de uso livre e outro desenvolvido pela marinha, nomeadamente o GCS e o código do Arduino, analisado de seguida.

O LattePanda funcionava com o sistema operativo Windows, onde era possível usar o TeamViewer para alterar o código do Arduino e trabalhar remotamente na plataforma. O código do Arduino foi desenvolvido em C++ e controla os motores.

O formato de mensagem NMEA é um conjunto de combinações elétricas e de dados para a comunicação entre dispositivos eletrónicos marítimos, como por exemplo eco-localizadores, receptores GPS, etc. A utilização deste protocolo permite uniformizar o formato de mensagens e facilitar a integração de outras soluções, caso se proceda a alterações. É mantido na solução atual pelas mesmas razões.

O OpenCPN é um software livre que cria um plotter gráfico, semelhante a uma carta náutica e que dá para visualizar posição e várias informações do sistema. Pode ser usado em andamento ou como ferramenta de planeamento. Este software é desenvolvido por uma equipa de velejadores ativos, que utilizam condições do mundo real para testes e melhoramentos do software. A título demonstrativo, a Figura 6 mostra a interface do OpenCPN.

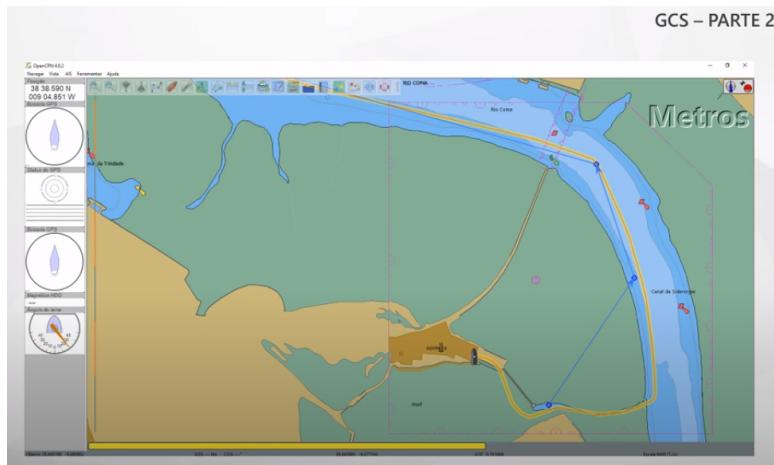


Figura 6: Software OpenCPN

O GCS (Ground Control Station) foi desenvolvido em C#. Esta linguagem de programação é desenvolvida pela Microsoft e é completamente suportada pela plataforma .NET Framework. A WinForms foi uma das frameworks de C# utilizada neste ponto, o que facilitou a criação da GUI do GCS. O resultado da solução original para controlo em terra é o que pode ser visto na Figura 7. No entanto, de forma a conciliar todos os veículos da marinha numa só ferramenta, terá de ser descartada a utilização da GCS em questão.

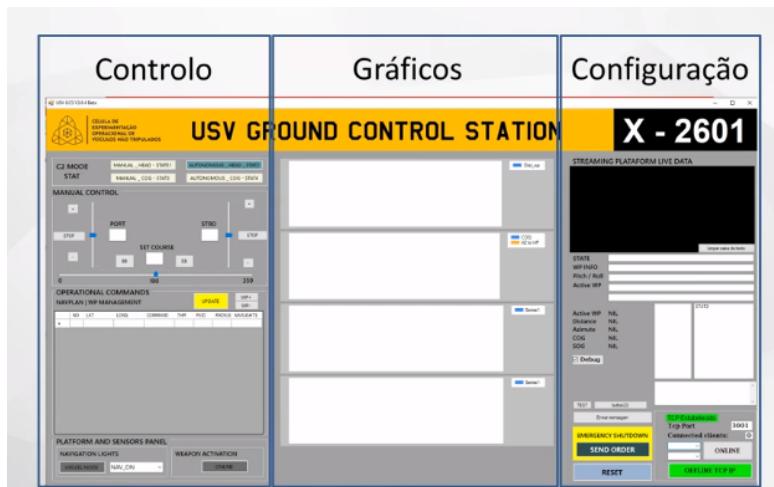


Figura 7: Interface Original da GCS

2.5 Objetivos a cumprir

A par com o cliente, definiu-se que a expansão das funcionalidades do catamarã deveria traduzir-se, de forma geral e obrigatória, no cumprimento dos seguintes requisitos:

- N1: O sistema deve possuir interoperabilidade entre os demais veículos da Marinha;
- N2: O sistema deverá dirigir-se autonomamente para os pontos de destino;

- N3: O sistema deve ser eficiente, a nível energético;
- N4: O sistema deverá conseguir comunicar, mesmo quando longe da costa.

Traduzindo as necessidades enumeradas, chegou-se a uma lista de requisitos funcionais e independentes da implementação, que se referem às tarefas de alto nível que o sistema deve conseguir desempenhar. Os requisitos funcionais estão identificados na Tabela 3. Os requisitos de performance, design, alocados e derivados podem ser encontrados no anexo C.

Tabela 1: Requisitos funcionais

Código de Requisito	Requisito Associado	Descrição	Classificação
RF1	N1	A GCS deverá ser capaz de comunicar a missão ao USV.	Essencial
RF2	N2	A GCS deve conseguir planejar missões simples (escolha de pontos de passagem).	Essencial
RF3	N2	O USV deve conseguir executar missões simples (escolha de pontos de passagem).	Essencial
RF4	N2	Dado um ponto de destino, o USV deve ser capaz de se deslocar para esse mesmo destino, com uma tolerância de 1 metro.	Essencial
RF5	N1	A GCS deverá ser capaz de monitorizar a missão, com dados atualizados no máximo a cada 15 segundos.	Essencial
RF6	N1, N2	A GCS deve conseguir planejar missões complexas (manobras avançadas, de trajetos curvos definidos, etc.).	Opcional
RF7	N1, N2	O USV deve conseguir executar missões complexas (manobras avançadas, de trajetos curvos definidos, etc.).	Opcional
RF8	N2	Implementar uma rotina de segurança (retornar à base) no caso do USV perder comunicação por mais de 1 minuto com a plataforma de planeamento e execução.	Opcional
RF9	N2	No caso da bateria/depósito estiver inferior a 5%, o USV deverá retornar à base.	Opcional

3 Estudo da Aplicação da LSTS Toolchain do Projeto

A introdução das ferramentas do LSTS Toolchain acontece em 2 níveis: na GCS com o Neptus e no centro de Processamento On-Board com o DUNE. A filosofia base da toolchain do LSTS é a modularidade cumprida à custa de threads que trocam mensagens, constituindo, a par da flexibilidade, uma das vantagens da escolha.

O DUNE[12] é responsável pela interface entre o nível da missão e o USV, bem como com os sensores e atuadores, a nível de execução.

O software Neptus[13] tem o papel de comandar, controlar e monitorizar o sistema USV. Faz o planeamento de alto nível e constitui a interface entre a decisão e a lógica do sistema propriamente dita.

3.1 Papel do DUNE no sistema

O DUNE, *Unified Navigation Environment*, é um software que permite gerir as interações com sensores e atuadores, coordenar as comunicações, executar o plano de navegação e supervisionar o veículo.

O software desenvolvido em DUNE segue um padrão de design de software "publish-subscribe". Cada funcionalidade é dividida em *Tasks* que correm numa *thread* à parte e que deverão implementar algo simples e atómico que seja independente de outras funcionalidades (modularidade). Isto torna as funcionalidades mais robustas e facilita a adição de funcionalidades, mais tarde, simplesmente adicionando mais *Tasks* unitárias e independentes (escalabilidade). A comunicação entre *Tasks* é realizada através de mensagens IMC que são enviadas por diversas *Tasks* para um barramento IMC a que todas as *Tasks* têm acesso. Cada *Tasks* subscreve ao tipo de mensagens que lhe interessa (por exemplo, uma *Task* de análise de temperatura, está interessada em mensagens do tipo *IMC::Temperature*) e recebe essas mensagens, sem ter conhecimento de onde vêm (se isso for relevante, é também possível identificar a fonte), assim como as *Tasks* que enviam mensagens não sabem, necessariamente, que *Tasks* vão recebê-las.

As *Tasks* implementadas no DUNE foram:

- LT1000 (source/src/sensors)- tarefa para receção, conversão e reencaminhamento da informação lida pelo módulo de sensores LT1000;
- SimpleHeadingAndSpeed (source/src/control/USV)- com base na direção e velocidade de referência recebidos, implementa controlo PID para direção e outro para a velocidade. Os resultados são combinados para se obter o controlo de motores e o sinal de direção para controlo dos lemes;
- UART_Comm (source/src/actuators)- tarefa que recebe os dados dos lemes e dos motores, da tarefa anterior e transforma-os para o protocolo definido e envia para o Arduino via USB;
- MinAltitude (source/src/Supervisors/MinAltitude)- com os dados da profundidade medidos, avalia-se se é viável ou não para a navegação, parando em caso insuficiente.

Cada veículo deve ser identificado pelo seu endereço IMC, que deve ser incluído no ficheiro *imc-addresses.ini* (dune/source/common) para possibilitar a comunicação e interação entre sistemas. Atribuiu-se o *IMC address* 0x0804 para o caso.

3.1.1 Ficheiro de configuração

Todos os veículos criados no DUNE necessitam de um ficheiro de configuração (xxx.ini) onde se colocam as tarefas que correm em modo hardware, simulação ou sempre. Além disso é possível alterar os valores de constantes relevantes às funções que se desejam criar, como o baud rate, os ganhos dos PID ou as tarefas monitorizadas.

Para a embarcação desenvolvida neste projeto foi criado o ficheiro *ceov-asv.ini* (dune/source/etc), onde são inicializadas as tarefas mencionadas.

Para inicializar o *ceov-asv.ini* e as respetivas tarefas, deve ser executado dentro da pasta "build" o seguinte comando:

- **Hardware:** ./dune -c ceov-asv -p Hardware
ou
- **Simulação:** ./dune -c ceov-asv -p Simulation

3.2 Papel do Neptus no sistema

O Neptus é o software utilizado pela Estação de Controlo, cuja comunicação é realizada principalmente por IMC, facilitando a integração de embarcações com o DUNE. Outras funcionalidades a implementar são facilmente adicionadas ao software do Neptus através de *plug-ins*.

O Neptus dá suporte em todo o ciclo de vida da missão: planeamento, simulação e execução. Recorrer a uma única ferramenta de alto nível é uma vantagem para sistemas críticos como o USV em questão. Desta forma, o caráter adaptativo e flexível vai ao encontro com os requisitos do cliente, da Tabela 3:

1. N1: Interoperabilidade do USV com veículos da Marinha

Ao garantir a homogeneidade entre o protocolo de comunicação, garante-se também a interacção e interoperação do catamarã com a restante frota. Adicionalmente, o planeamento de missões integrado dos veículos é agilizado e tornado mais eficiente pela utilização de uma única plataforma de controlo e comando

2. N2: O USV dirige-se autonomamente para o destino

É o Neptus que, neste sistema, tem a função de planear as rotas a adoptar pelo catamarã, tornando-se assim uma ferramenta fundamental para o controlo de alto nível e da navegação autónoma deste. Esta framework recorre a diversos standards e normas, o que torna a solução final mais robusta.

3. N3: O USV deve ser eficiente a nível energético

O Neptus permite planear missões recorrendo a diferentes tipos de dados geográficos, incluindo standards de navegação integrados. Assim, podem-se identificar rotas e manobras mais eficientes a níveis energéticos, previamente simulados e validados de acordo com as capacidades do sistema.

Adicionalmente, o Neptus é importante tanto na interface com o utilizador, nomeadamente planeamento de missões futuras e visualização de dados em tempo real (p.ex. localização e velocidade), como na revisão de missões concluídas (histórico de missão). Estas duas interfaces estão indicadas na Figura 8

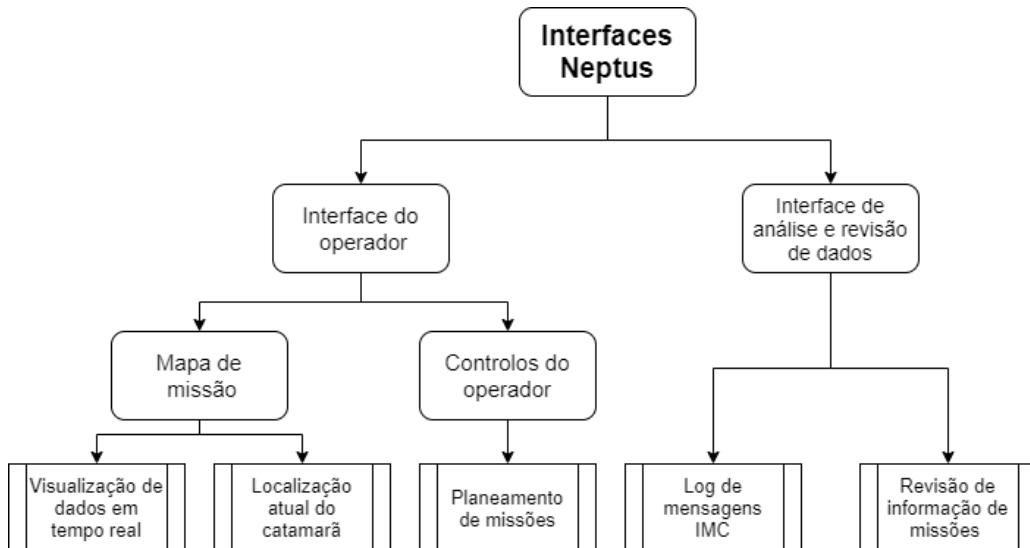


Figura 8: Interfaces oferecidas pelo Neptus

3.2.1 Ficheiro de configuração

Para um veículo ser reconhecido no Neptus é necessário um ficheiro com a sua identificação. Para tal foi criado o ficheiro 09-ceov-asv.nvcl (neptus/vehicle-defs) usando como ponto de partida o 09-caravela.nvcl (neptus/vehicle-defs), dadas as semelhanças das embarcações. Alterou-se o id na linha 5 para corresponder ao declarado no imc-addresses.ini (dune/source/common). Na linha 363 utilizou-se o host-address 10.0.200.80, IP atribuído a embarcação. Por fim na linha 371 utilizou-se o imc-id 08:04, o mesmo que foi atribuído no imc-addresses.ini (dune/source/common) no DUNE.

4 Conceito do Sistema

Esta secção descreve a versão atual conceito do sistema que é a atualização do documento de Conceito de Sistema, no anexo D. Apresenta-se inicialmente o SBS para contextualizar os sistemas do veículo.

4.1 SBS

A nível dos componentes do sistema, este pode ser decomposto em duas partes: a estação de controlo (GCS) e a embarcação (USV). A solução culminou na implementação de software dos dois lados. A estação de controlo corre num computador que suporta o Neptus. O USV é controlado pelo software DUNE, que corre num LattePanda. O Arduino é apenas responsável pelo envio dos comandos PWM para os atuadores. A escolha de manter o Arduino conectado com os atuadores advém de uma restrição imposta pelo cliente, justificada com as ligações físicas já feitas no sistema. Esta decomposição hierárquica pode ser visualizada na Figura 9, cujo alvo de desenvolvimento é o software rodeado a azul.

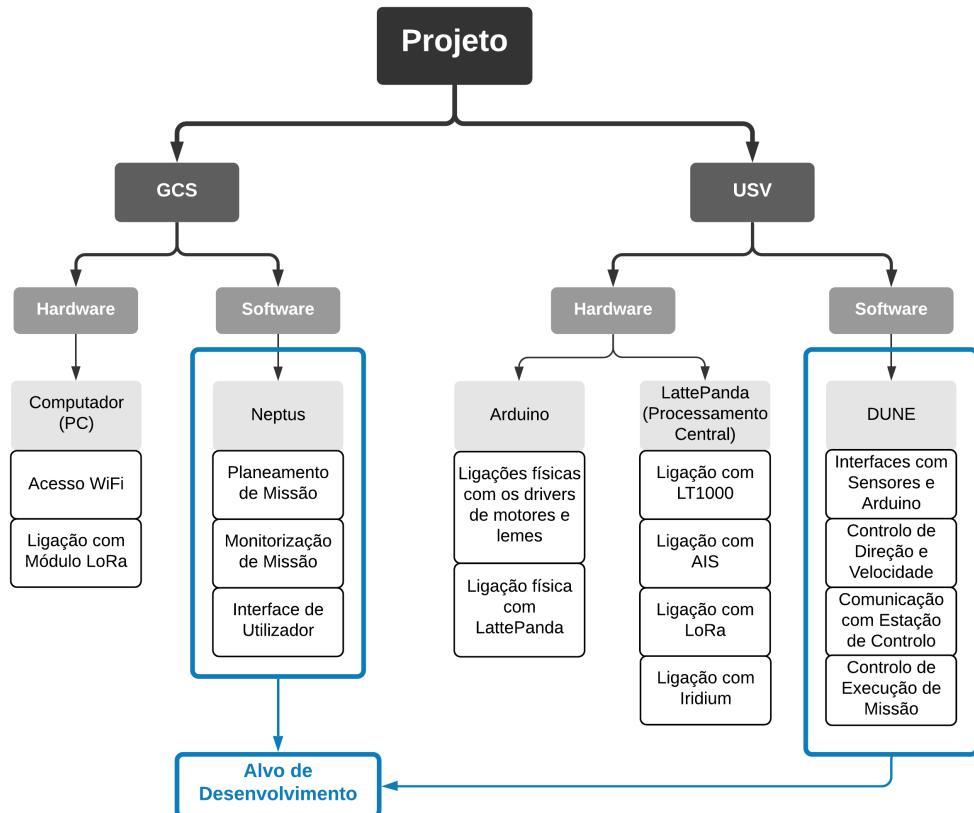


Figura 9: System Breakdown Structure

4.2 Arquitetura Funcional e Mapeamento em Dispositivos Computacionais

Em termos funcionais, a solução pode dividir-se em 2 campos: o *Ground Control Station* (GCS) e o veículo propriamente dito, por sua vez constituído pelo centro de processamento a bordo e os atuadores (dito o baixo nível do sistema). Esta organização está representada na Figura 10 que ilustra o conceito de sistema a desenvolver.

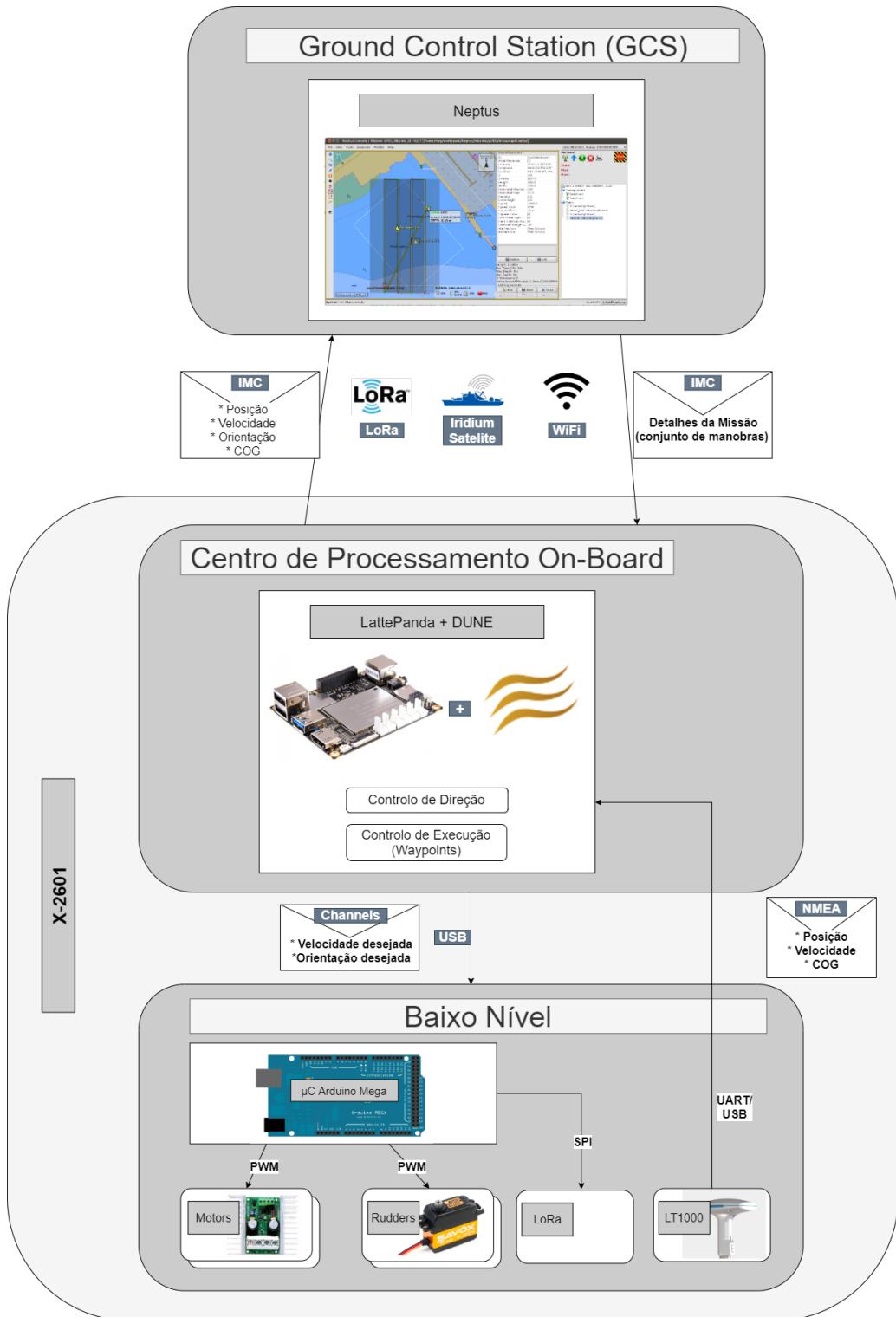


Figura 10: Estado final do sistema

Os sensores e atuadores do catamarã integram-se no baixo nível do sistema. Os sensores estão todos integrados na unidade de navegação marítima LT1000 (ver secção 5.1) e os atuadores incluem os lemes e os motores. Portanto, o baixo nível do sistema encontra-se a bordo do catamarã e necessita de camadas hierárquicas superiores para controlo e comando das missões: um centro de processamento intermédio a bordo do veículo (que também corre no DUNE) e um controlo de missão em terra, a GCS, que usa o Neptus.

O centro de Processamento On-Board encontra-se também no veículo e corre num computador dedicado, o LattePanda. É aqui que o DUNE controla o movimento do veículo com base nas medições em tempo real. Iterativamente, ao receber os dados de missão a implementar, pelas camadas hierárquicas superiores, o centro de processamento calcula os pontos (*waypoints*) que o USV deve seguir e a direção da trajetória a tomar.

Consequentemente, o centro de processamento *On-Board* envia uma mensagem com a velocidade e orientação a impor nos atuadores.

O nível hierárquico superior em terra, na GCS, está encarregue pela gestão da missão e pela supervisão do USV. Nesta vertente, recorreu-se ao software Neptus que terá o papel de comandar, controlar e monitorizar o catamarã.

As alterações realizadas face ao implementado originalmente, incluem:

- O planeamento e controlo da missão passa a ser feito pelo Neptus;
- O LattePanda passa a ser agora o centro de lógica do sistema: recebe o planeamento de alto nível do Neptus e transforma-o em informação comprehensível para os atuadores, recorrendo ao DUNE;
- O Arduino torna-se o intermediário entre o centro de lógica e os atuadores (motores e lemes);
- O OpenCPN dá lugar às cartas náuticas incorporadas no Neptus.
- A comunicação entre as partes de alto e baixo nível do sistema realiza-se com LoRa, até um alcance de 8Km. Em caso de perda de comunicação, será utilizada comunicação via satélite.

Com esta implementação pretendeu-se dotar o catamarã de mais autonomia de navegação, tornar a sua ação mais precisa, as tarefas de controlo mais fiáveis e promover a segurança por rotinas de *fail-safe*.

4.3 Casos de Uso do Sistema

Do ponto de vista do utilizador, a solução proposta permite a obtenção da informação do veículo, como posição, planear missões e comandar o veículo. Numa linguagem mais gráfica, esta descrição pode ser realizada por meio do diagrama de Casos de Uso, da Figura 11.

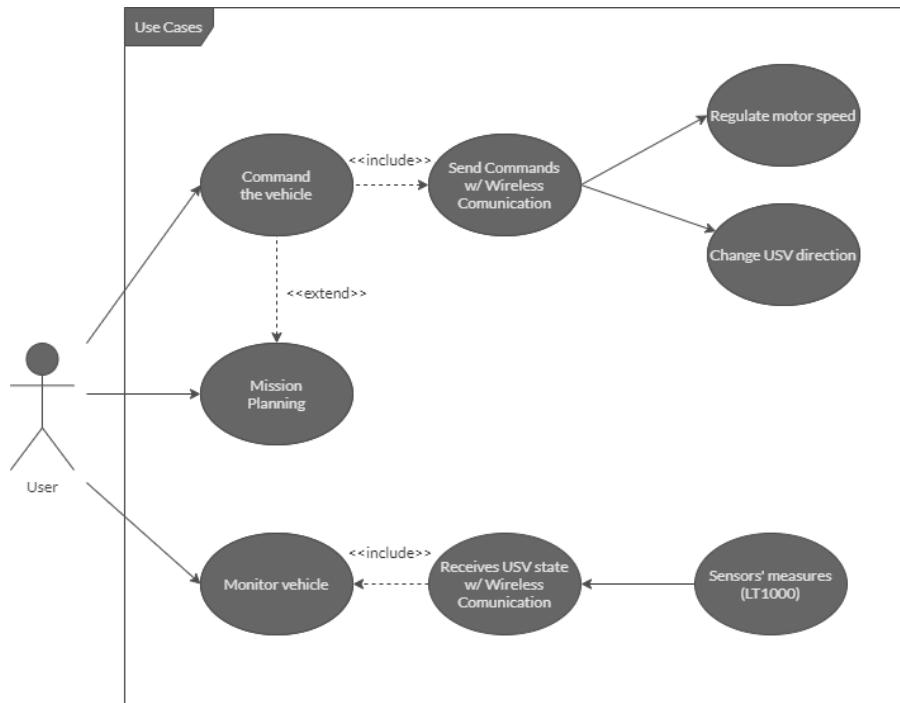


Figura 11: Diagrama de Casos de Uso

4.4 Diagramas de Fluxo de Mensagens

Em termos de interações entre os diversos elementos constituintes do sistema USV, a Figura 12 pretende clarificar estas relações. Os blocos funcionais de software garantem a troca de informação ao longo dos vários níveis de controlo, desde o Neptus como sistema de planeamento aos sensores e atuadores.

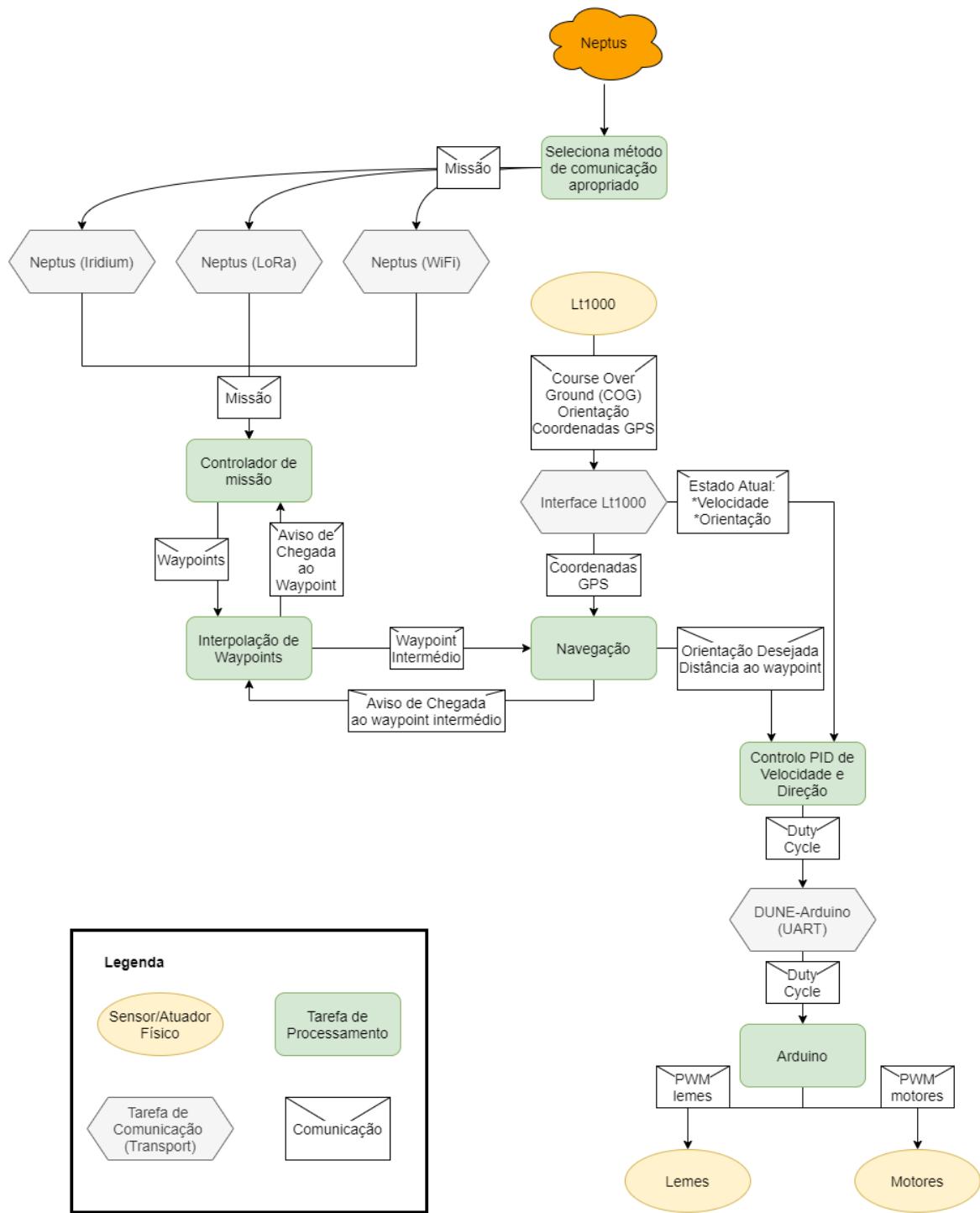


Figura 12: Descrição das comunicações trocadas entre as partes do sistema

Terá de ser atualizado o esquema por causa das comms?

5 Trabalho realizado

As implementações desenvolvidas enquadram-se em 3 grandes áreas - sensorização, atuação e comunicação - onde foram focados os esforços da equipa. Quanto à sensorização, foi criada uma tarefa que recebe informações do LT1000, descodificando a mesma e enviando informação útil para as camadas superiores de navegação. Existem 2 tarefas relacionadas com os atuadores: `SimpleHeadingAndSpeed` e `UART_Comm`, que calculam a direção e velocidades, respetivamente, a impor ao veículo por meio dos motores e lemes e enviam a informação PWM à interface do Arduino. Existe a tarefa de comunicações que valida o conceito de envio de planos da GCS para o módulo LoRa e do módulo LoRa a bordo para o Dune. A um nível hierárquico superior, desenvolveu-se uma tarefa de controlo de profundidade a que o veículo navega, a `MinAltitude`. É de realçar que, para que o controlo seja feito de forma correta, os ficheiros devem ser colocados nas pastas indicadas, do diretório do DUNE.

5.1 LT1000

O LT1000 Navigation Reference Unit (NRU), Figura 13, é um produto de navegação marítima desenvolvido pela Lars Thrane A/S. Na verdade, é empregue tanto em ambiente de navegação de lazer como no mercado marítimo profissional. A unidade de sensorização inclui 12 sensores de precisão num único módulo, nomeadamente magnetómetros, giroscópios, acelerómetros, GNSS, barómetro e termómetro. Os sensores cumprem as normas e certificados necessários ao equipamento de navegação marítima.



Figura 13: LT-1000

O módulo já integra um filtro de Kalman e fusão sensorial, garantindo uma maior precisão na medição dos vários indicadores de navegação:

- *True heading*- ângulo entre o Norte Geográfico e o eixo longitudinal da embarcação, a laranja na Figura 14;
- *Magnetic heading*- ângulo entre o Norte Magnético e o eixo longitudinal da embarcação, a azul na Figura 14;
- *Magnetic deviation*- erro induzido pelo campo magnético da embarcação
- *Magnetic variation*- causado pela variação do campo magnético terrestre;
- *Roll*- ângulo con o eixo longitudinal do veículo (consultar Figura 15) ;
- *Pitch*- ângulo de rotação em relação ao eixo transversal do veículo (consultar Figura 15) ;
- Posição no sistema de coordenadas WGS84;
- *Ground speed*- velocidade na direção de deslocamento (SOG)
- *Course over ground*- ângulo entre o norte geográfico e a direção de deslocamento ;

- Hora e data;
- Pressão do ar;
- Temperatura em tempo real;

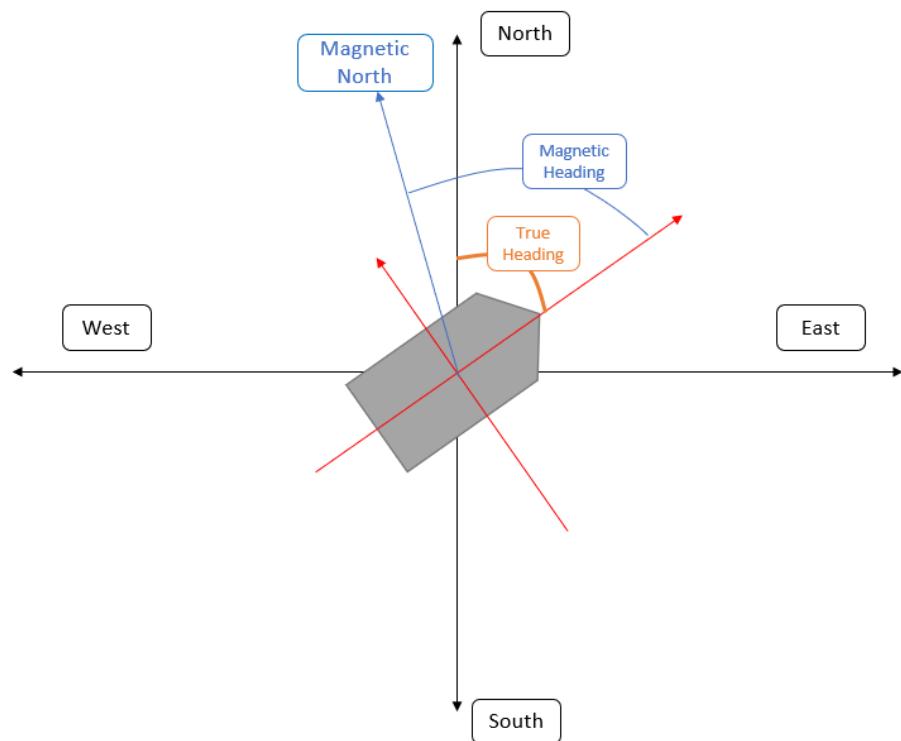


Figura 14: True Heading vs Magnetic Heading

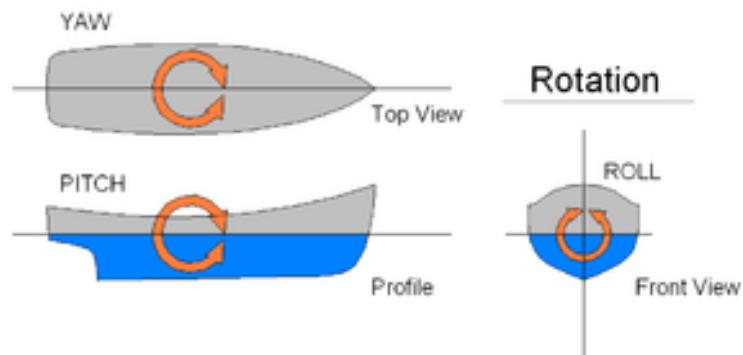


Figura 15: Roll, Pitch, Yaw. Fonte: [1]

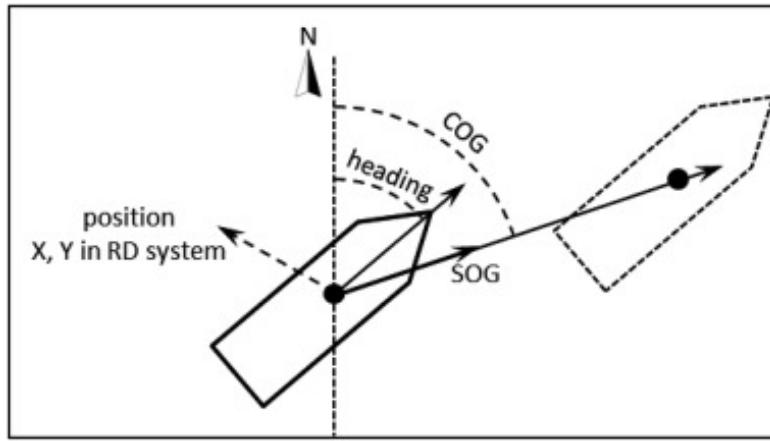


Figura 16: COG vs SOG. Fonte: [2]

O LT-1000 NRU utiliza a mais recente tecnologia em receptores GNSS e possui apenas um cabo que é compatível com NMEA 0183, NMEA 2000 e alimentação.

5.1.1 NMEA

NMEA é um conjunto de especificações de dados e ligações elétricas desenvolvidos pela *National Marine Electronics Association* para comunicação de dispositivos eletrônicos de navegação. Este protocolo é utilizado em dispositivos como anemómetros, bússolas, piloto automático e receptores GPS. Quando a saída NMEA está habilitada, um subconjunto de mensagens NMEA pode ser enviado, para equipamentos externos, conectados às portas série do receptor. Os formatos destas mensagens variam com a informação que está a ser enviada. Um exemplo de mensagem para posição, velocidade e tempo é a mensagem RMC que, a título exemplificativo, apresenta o formato do tipo:

\$GNRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

no qual cada posição separada por uma vírgula corresponde a um dado parâmetro:

- ID da mensagem: GNRMC;
- UTC de correção de posição: 123519;
- Estado: A=ativo;
- Latitude: 4807.038,N;
- Longitude: 01131.000,E;
- SOG (Speed over the ground): 022.4;
- Ângulo (graus): 084.4;
- Data: 230394;
- Variação Magnética (graus): 003.1,W;
- Verificação da soma: *6A;

NMEA 0183		
SENTENCE	DESCRIPTION	RATE
4800 BAUD		
GNRMC	Recommended Minimum Specific GNSS Data	1 Hz
HCHDG	Heading and Magnetic Heading Variation	1 Hz
HCHDM	Magnetic Heading	1 Hz
HCHDT	True Heading	10 Hz
HCROT	Rate of Turn	1 Hz
PFFC,GPatt	Altitude	1 Hz
WIMDA ¹	Meteorological Composite	0.5 Hz
38400 BAUD		
GNDTM	Datum Reference	1 Hz
GNNGA	GPS Fix Data	1 Hz
GNGLL	Position Latitude/Longitude WGS84	1 Hz
GNGSA	GNSS DOP and Active Satellite	1 Hz
GNRMC	Recommended Minimum Specific GNSS Data	1 Hz
GNVTG	Course Over Ground and Ground Speed	1 Hz
GNZDA	Time and Date	1 Hz
GPGSV ²	GNSS Satellites in View	1 Hz
HCHDG	Heading and Magnetic Heading Variation	10 Hz
HCHDM	Magnetic Heading	10 Hz
HCHDT	True Heading	10 Hz
HCROT	Rate of Turn	10 Hz
HCTHS	True Heading and Status	10 Hz
PFFC,GPatt	Altitude	10 Hz
WIMDA ¹	Meteorological Composite	2 Hz
WIXDR ³	Transducer Measurements	2 Hz

Figura 17: NMEA 0183 - LT1000 *Fonte: Datasheet LT1000[3]*

Para a tarefa que se pretendeu desenvolver, foram criadas funções de leitura das mensagens do tipo ZDA, GGA, RMC, HDT e VTG, utilizando-se um baud rate de 38400, como sugerido na figura 17.

5.1.2 Tarefa LT100

No desenvolvimento da tarefa encarregue pelo módulo LT1000, foi utilizado como ponto de partida a tarefa Sensors.GPS. A interpretação das mensagens ZDA, GGA, HDT e VTG foram recicladas e desenvolveu-se o suporte de outras mensagens construídas pelo módulo de sensores. A leitura e descodificação de mensagens RMC foi conseguida com a criação de uma função dedicada, nesta tarefa. A função lê todos os parâmetros da mensagem recebida, guardando os dados numa instância IMC GpsFix [14]. Para a leitura de algumas variáveis das mensagens NMEA foram criadas funções de leitura da Latitude (readLatitude) e da Longitude (readLongitude), para converter os dados recebidos no formato WGS 84 para ângulos em graus.

Feita a descodificação dos dados de input, são enviadas mensagens IMC do tipo GpsFix [14] e EulerAngles [15] às tarefas de Navegação, ou seja, enviam-se os dados às camadas hierárquicas superiores do DUNE de forma a proceder ao controlo de trajetória de alto nível e supervisão.

No desenvolvimento da *task* teve-se em conta a possibilidade de testar a leitura de mensagens NMEA, sem utilizar um dispositivo ligado por porta série (como detalhado no capítulo 5.2.3). De forma a tornar o processo mais flexível e agilizar a alternância entre debug e funcionamento normal, a receção de mensagens está preparada para receber informação tanto por porta série como por TCP. Para alternar entre estas opções basta alterar o *Serial Port -Device* da tarefa Sensors.LT1000 no ficheiro de configuração do *ceov-asv.ini*, como no seguinte exemplo:

- **TCP:** Serial Port - Device = tcp://127.0.0.1:xxxx
- **Porta Série:** Serial Port - Device = /dev/xxxx

Aqui, o IP será aquele pelo qual o simulador/sensor está a enviar mensagens. A porta série é a porta do LattePanda na qual o sensor está fisicamente ligado. Para mais informações, consultara documentação: <https://sites.google.com/g.upporto.pt/seai-equipae#h.17bt8u6jiotf>.

5.1.3 Simulador de Mensagens NMEA

Para verificar o funcionamento da tarefa do LT1000 no DUNE, foi utilizado um simulador de mensagens NMEA, enviando-as por TCP. Este simulador gera um fluxo de dados NMEA que imita o movimento da embarcação, o status do motor, profundidade da água, entre outros. O simulador permite também registar as saídas NMEA num arquivo de registo que pode ser reutilizado noutras experiências.

O simulador e todas as instruções de instalação encontram-se disponíveis em [16].

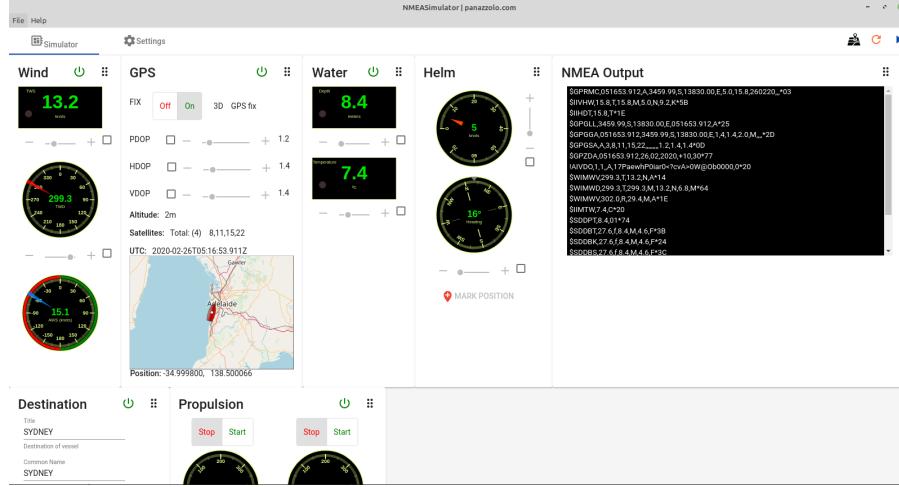


Figura 18: Simulador Mensagens NMEA

No capítulo de resultados ?? é possível encontrar uma demonstração da tarefa de sensorização com o referido simulador.

5.2 Atuadores

Segundo a ideia de utilizar os atuadores já existentes no X-2601, os drivers dos motores a controlar são os de [9] e os drivers dos lemes encontram-se em [10].

Uma vez que o Arduino está ligado diretamente aos drivers e sendo um requisito do cliente manter as ligações assim, escolheu-se a arquitetura previamente mencionada na Figura 10.

De forma a controlar os atuadores, desenvolveram-se 2 tarefas no DUNE: SimpleHeadingAndSpeed e UART_Comm. A tarefa SimpleHeadingAndSpeed calcula a velocidade e orientação desejadas e aplica um controlo PID para os motores (velocidade+direção) e outro para os lemes(direção). A tarefa UART_Comm vem na sequência da SimpleHeadingAndSpeed comunica ao Arduino os valores PWM a impor.

5.2.1 Tarefa SimpleHeadingAndSpeed

A tarefa foi criada de raiz com o objetivo de fazer um controlo PID, tanto da velocidade como da direção, e está no diretório src/Control/USV/. Pode dizer-se que é uma tarefa reativa, já que só é atuada caso se recebam mensagens IMC. De forma rápida, reage às mensagens EstimatedState [17], DesiredHeading [18], DesiredSpeed [19], Abort [20], ControlLoops [21], num estilo de consumidor.

De cada vez que é despoletada, podem tomar-se vários comportamentos, consoante a natureza da mensagem:

- IMC::ControlLoops [21]-> procede-se à ativação/destivação do controlo PID
- IMC::DesiredHeading [18] e IMC::DesiredSpeed [19]-> procede-se ao cálculo da referência de direção/- velocidade
- IMC::EstimatedState [17]-> controlo PID com o erro de direção e velocidade

Desta forma, as mensagens enviadas também são IMC, cumprindo com a arquitetura do Neptus e denominam-se SetServoPosition [22] e SetThrusterActuation [23].

Para mais informações, consultar a documentação: <https://sites.google.com/g.upporto.pt/seai-equipae#h.17bt8u6jiotf>.

5.2.2 Tarefa UART_Comm

Após o cálculo das referências de velocidade e direção para o USV e implementado o controlo PID, é necessário passar essa informação aos atuadores. Como existe um Arduino num ponto intermédio, entre o centro de processamento a bordo e os drivers dos motores e lemes, é necessário passar esses comandos na porta série.

Ora, tal como a tarefa anterior, a tarefa UART_Comm, que se localiza em `src/Actuators/` é executada forma reativa, consumindo mensagens IMC. Cada vez que se recebem dados de controlo dos motores (IMC: :SetThrusterActuation [23]) e lemes (IMC: :SetServoPosition[22]), estes são formatados para serem escritos como PWM na porta série (UART). Definiu-se que as mensagens tomam o formato:

“Id do atuador no arduino” + “valor a atualizar no atuador”,

por exemplo "M 1500" para o motor 2 e 1500 o valor a impor.

Desta vez, o resultado da task é a escrita para a porta série respetiva. Para mais informações, consultar a documentação: <https://sites.google.com/g.upporto.pt/seai-equipae#h.17bt8u6jiotf>.

5.3 Comunicações

Para garantir a monitorização e controlo remotos do USV, ou seja, para se obter um controlo em terra na *Ground Control Station*(GCS) planearam-se comunicações de 3 formas: wifi, LoRa e Satélite. Ao conseguir-se redundância nas comunicações, pretende-se otimizar a troca de informação e minimizar as falhas de comunicação, visto tratar-se de um sistema crítico. Como objetivo final, pertende-se uma comunicação que se estenda por grandes áreas navegáveis (longe da costa), mantendo tempos de comunicação fiáveis.

Considerando o alcance do Wifi, LoRa e satélite e sabendo que:

- o WiFi caracteriza-se pela rapidez de transmissão e pela largura de banda permitida;
- o Wifi permite alcançar até cerca de 250 metros;
- o LoRa, com pouca potência dispendida consegue atingir distâncias superiores ao WiFi, até 15km. Tem vantagens mais longe da costa por garantir mais autonomia, poupando bateria e evitando várias viagens para carregamento;
- o Iridium, embora implique um custo associado a cada mensagem, permite comunicação em qualquer ponto

Considera-se que se deve recorrer ao Wifi, LoRa e Satélite, por ordem decrescente de preferência. O objetivo esperado seria uma comunicação com maior grau de mobilidade para longas distâncias da costa, ao mesmo tempo que se mantinha fiável e redundante.

5.3.1 Wifi

Segundo a arquitetura do Neptus, se tanto a GCS (Neptus) como o USV (DUNE) estiverem ligados à mesma rede local conseguem encontrar-se por broadcast. Portanto, o único requisito para ter controlo e monitorização em wifi é, por exemplo, usar VPN. No caso dos testes unitários e demonstrações, utilizaram-se os diferentes equipamentos ligados à VPN FEUP.

5.3.2 LoRa

Os módulos LoRa utilizados são LoRa6500Pro, da Figura 19, que é um módulo robusto para distâncias abertas até 15Km a 91bps. A comunicação série é TTL, RS232 e RS485 configuráveis e as configurações de rede e sinal RF podem ser editados, em caso de necessidade.



Figura 19: Módulo LoRa6500Pro

Sucintamente, a configuração dos módulos foi segundo:

- comunicação série TTL, por USB do lado da GCS e RS232 no USV;
- baud rate recomendado de 9600bps, 8bits de dados, 1bit de paridade Par e 2 stop bits
- mensagens enviadas por protocolo Lora/LoraWAN
- conexão ponto a ponto entre GCS e USV
- comunicação segundo a European Telecommunications Standards Institute (ETSI), com sinal RF de 868MHz (menores interferências)

O software desenvolvido, no diretório `src/Transports/`, envolve a receção de mensagens de plano e do envio do estado do USV. No que toca à receção, as mensagens que chegam por serial e correspondem ao plano simplificado (mensagem `char* Plan`), formato json, para posterior parse. As mensagens do estado da embarcação são enviadas também por serial USB-TTL, ao módulo LoRa, para posterior envio à GCS segundo o protocolo específico.

A tarefa consegue eficazmente fazer o envio, receção e parsing da informação e constitui a prova de conceito para o envio do plano do Neptus (por mensagem IMC) para o Dune. Para mais informações, nomeadamente a configuração, consultar a documentação: <https://sites.google.com/g.upporto.pt/seai-equipae#h.17bt8u6jiotf>.

5.4 Controlo de Profundidade

A tarefa `MinAltitude` atua ao nível de decisão, atuando como um supervisor à navegação e encontra-se no diretório (`src/Supervisors/MinAltitude/`). Enquanto os controladores de velocidade e direção (ver secção 5.2) estiverem ativos, a tarefa corre em modo de supervisão de profundidade, de acordo com a profundidade mínima definida no ficheiro de configuração. Para esta supervisão, consomem-se mensagens IMC do tipo `IMC::EstimatedState` [17], `IMC::Abort` [20] e `IMC::ControlLoops` [21].

Em caso de profundidade insuficiente, a missão é abortada e o veículo permanece imóvel. Do ponto de vista de navegação, este fail-safe protege o UVS de situações que comprometam a segurança do veículo. Para parar a navegação, produz-se uma mensagem `IMC::Abort` [20] adequada.

Para mais detalhes sobre a tarefa `MinAltitude`, consultar a documentação: <https://sites.google.com/g.upporto.pt/seai-equipae#h.17bt8u6jiotf>.

Parte II

Resultados e Análise

6 Resultados

De acordo com a metodologia bottom-up, começou-se por realizar testes individuais aos módulos e só depois todo o sistema integrado, garantindo um integração mais fluída e agilizando os testes conjuntos. Começou-se pelos sensores, depois tarefas de atuadores e termina-se com um teste de integração, para várias manobras possíveis.

6.1 Resultados Simulador NMEA(panazzolo.com)

Seguindo uma abordagem de testes unitários, procedeu-se à simulação do sensor com o NMEA Simulator e a tarefa desenvolvida no DUNE.

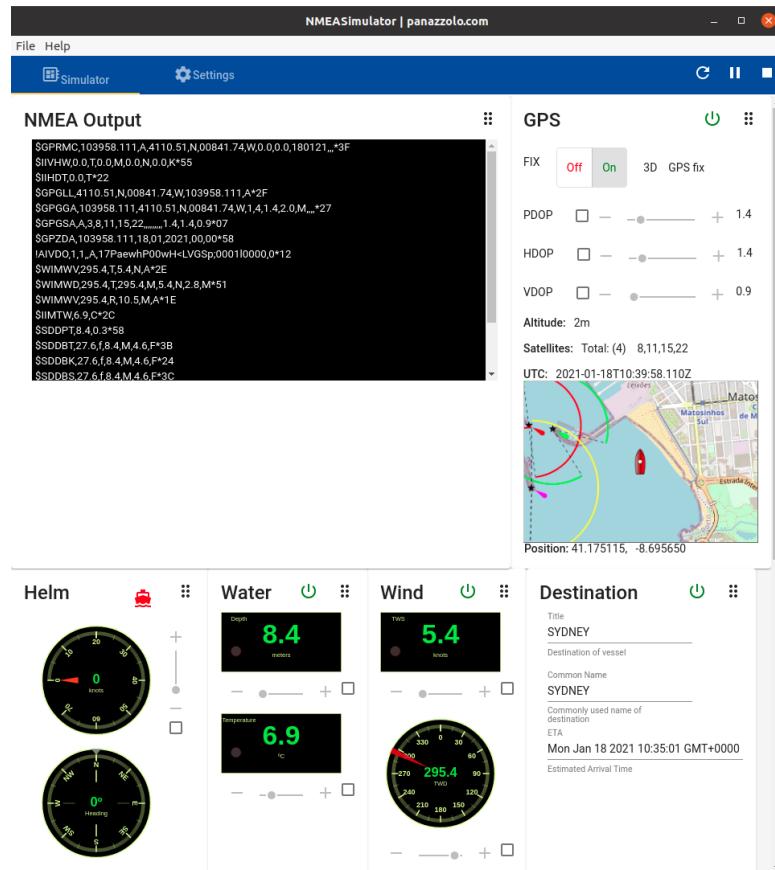


Figura 20: Valores enviados pelo Simulador de NMEA

A terminal window titled 'ricardo@ricardo-N552VX: ~/Desktop/SEAI-E/dune/build'. The window displays a log of messages from January 18, 2021, at 10:40:10. The messages are: [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> RMC; [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> latitude(graus): 41.175167; [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> longitude(graus): -8.695667; [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> latitude(radianos): 0.718642; [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> longitude(radianos): -0.151768; [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> cog: 0.000000; [2021/01/18 10:40:10] - MSG [Sensors.LT1000] >> sog: 0.000000.

Figura 21: Resultado da leitura dos dados no DUNE

Do simulador da Figura 20, retiram-se os dados de localização:

- **Latitude simulador NMEA:** 41.175115

- **Longitude simulador NMEA:** -8.695650

Enquanto que da Figura 21 resulta a leitura:

- **Latitude no DUNE:** 41.175167
- **Longitude no DUNE:** -8.695667

Deste modo, foi possível fazer o cálculo do erro entre o valor enviado pelo sensor simulado e o valor traduzido no DUNE:

- **Erro Latitude** = $|41.175115 - 41.175167| = 0.000052$
- **Erro Longitude** = $|-8.695650 - (-8.695667)| = 0.000017$

Convertendo-se os erros calculados em metros, para uma análise de resultados mais natural e perceptível, obteve-se um erro de distância de, aproximadamente, 6 metros. Para esta conversão recorreu-se ao website <https://www.sunearthtools.com/pt/tools/distance.php>.

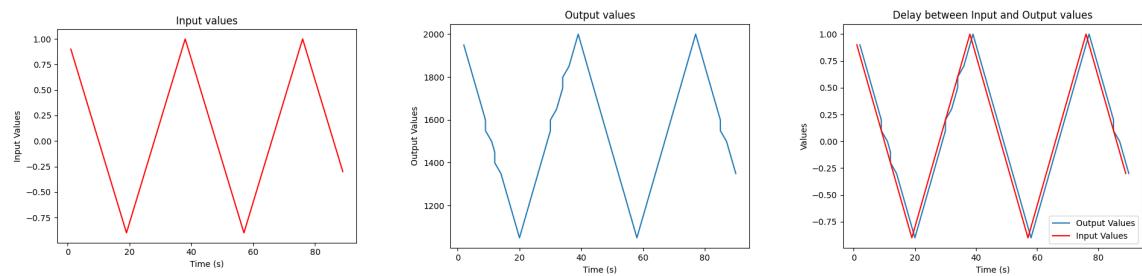
Este erro deve-se principalmente às conversões efetuadas na manipulação de dados entre formatos de coordenadas. As mensagens NMEA escrevem as coordenadas segundo a norma WGS84, tendo uma menor precisão dado apresentar menos algarismos significativos. Posteriormente, as coordenadas são convertidas para ângulos em graus no DUNE, acumulando-se erros de cálculo.

No seguinte link <https://youtu.be/0DQLyG5FBkg> é possível visualizar uma pequena demonstração destes resultados.

6.2 Atuadores

Para a validação dos atuadores, procedeu-se ao teste do módulo de envio para o Arduino (UART_Comms com as tarefas de atuação e Arduino).

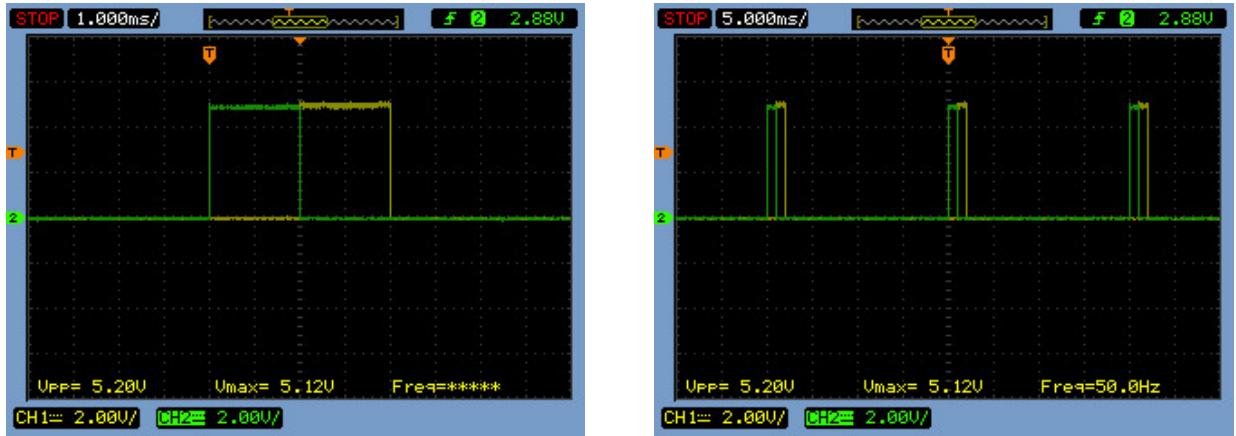
Primeiro criou-se uma tarefa de simulação para gerar valores de input em toda a gama, ou seja, entre 1000 e 2000, que simulam as mensagens IMC de IMC::SetThrusterActuation e IMC::SetServoPosition. Esses valores, que são normalizados para o intervalo [-1;1], permitem comandar os motores para a frente ou para trás, com valores positivos ou negativos respectivamente. Para os 1500, ou seja, normalizado em 0 a ordem é de paragem. Verifica-se que os valores recebidos e enviados coincidem, com um atraso de cerca de 900ms, como na figura



(a) Dados de Entrada- comandos das mensagens IMC (b) Dados de Saída- comandos para o Arduino (c) Comparação dos sinais de entrada/saída e análise do atraso (900ms)

Figura 22: Sinais de entrada e saída do teste UART

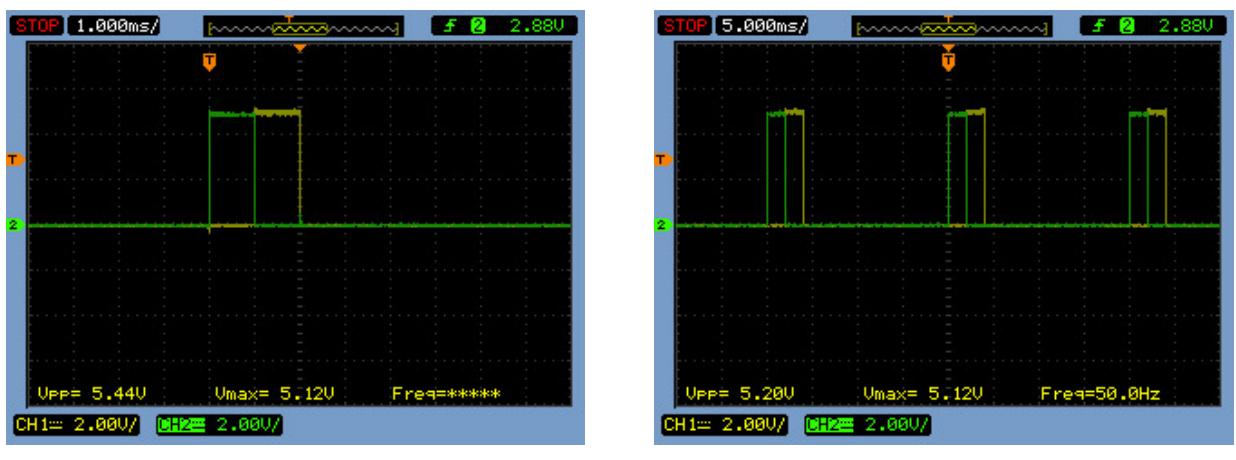
Quanto ao teste dos sinais de PWM, procedeu-se à variação de direção no Neptus com a respetiva medição dos sinais de PWM no Arduino. Estes sinais ligariam aos lemes, sendo que nas figuras 23 e 24, o leme da esquerda está representado no canal 1 a amarelo e o leme da direita está a verde no canal 2. Os testes incluíram manobras de viragem tanto à direita (Figura 23) como à esquerda (Figura 24).



(a) Zoom in

(b) Zoom out

Figura 23: Manobra de virar para a direita



(a) Zoom in

(b) Zoom out

Figura 24: Manobra de virar para a esquerda

Verifica-se que o período dos sinais respeita as especificações dadas pelas *datasheets*, variando entre 1ms e 2ms.

6.3 Simulação de Manobras

Através dos *logs* guardados no Neptus é possível fazer uma análise mais detalhada sobre a execução das missões. Nomeadamente, é possível analisar o caminho efetivamente percorrido pelo sistema, dados de profundidade, tempos para cada manobra associada a um plano (pontos de referência, por exemplo). Para teste do sistema em simulação foram aplicados 3 planos: um plano de varrimento, um plano de trajetória curva, um plano em que parte do trajeto incide numa área de altitude demasiada reduzida (testando-se que a manobra é interrompida). O modelo de simulação utilizado foi o de um catamarã já implementado no Neptus, cujos parâmetros como dimensões e peso foram atualizadas para os valores do X-2601: 4x2x1 (comprimento*largura*calado) e 300 kg. Todos os planos foram realizados em ambiente de simulação com uma corrente marítima de 0.2 m/s N e 0.2 m/s E.

6.3.1 Plano de Varrimento

Este plano pretende simular um percurso típico de varrimento de uma área para efeitos de estudo batimétrico. O percurso é constituído de pontos de referência que perfazem uma trajetória em onda quadrada de forma que a embarcação que a execute cubra uma área quadrada. Este plano constitui o teste mais simples, com corrente a nordeste (0.2 m/s N e 0.2 m/s E). O vídeo da simulação encontra-se em https://youtu.be/id-51h_EN1Q.

A Figura 25 ilustra a trajetória executada em relação ao plano original. A embarcação segue o caminho desejado, apesar das correntes, evitando fazer curva de cão (a embarcação favorece o alinhamento da direção de deslocação - COG - em relação à proa).

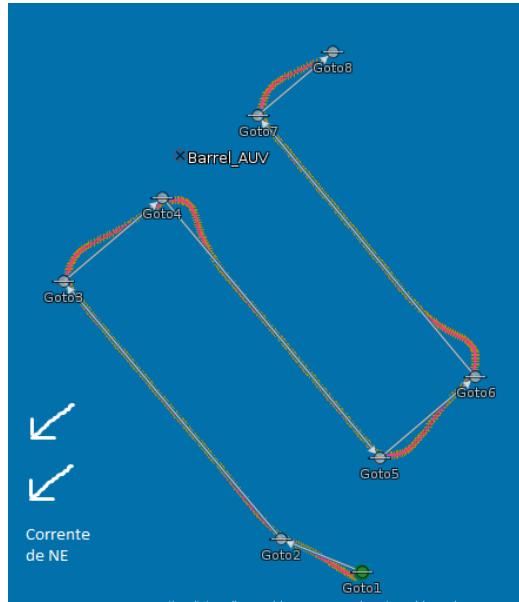


Figura 25: Trajetória efetivamente executada (a laranja) em comparação com a trajetória original planeada (a branco).

A Figura 26 representa a direção desejada em relação à direção efetiva (proa). Constata-se que, no momento de curvatura (ângulos retos), há um atraso de resposta por causa da mudança abrupta da direção. Consequentemente, ocorre uma sobrecompensação (de forma a realinhar com a trajetória desejada) que implica um maior desvio em relação à trajetória desejada. Não obstante, atinge-se um erro nulo, em regime permanente.

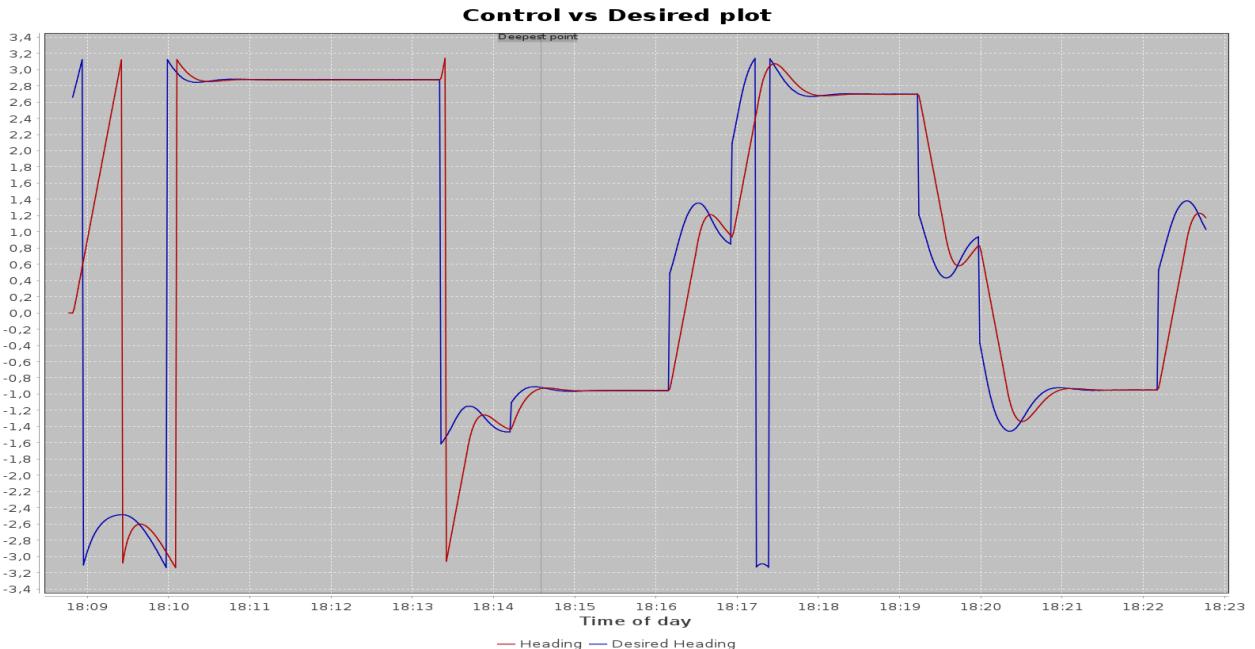


Figura 26: Direção desejada ao longo do percurso (azul) em comparação com a direção (proa) ao longo do percurso (vermelho).

A Figura 27 apresenta o tempo que o sistema demorou a executar cada um dos troços do percurso, onde se constata que o tempo entre pontos que estão a mesma distância é aproximadamente igual. A missão demorou

um total de 9 minutos e 26 segundos, com uma velocidade média de 1.020 m/s (555 m de distância). As várias retas azuis representam os vários troços do percurso, ou seja, os segmentos de reta constituintes do plano.

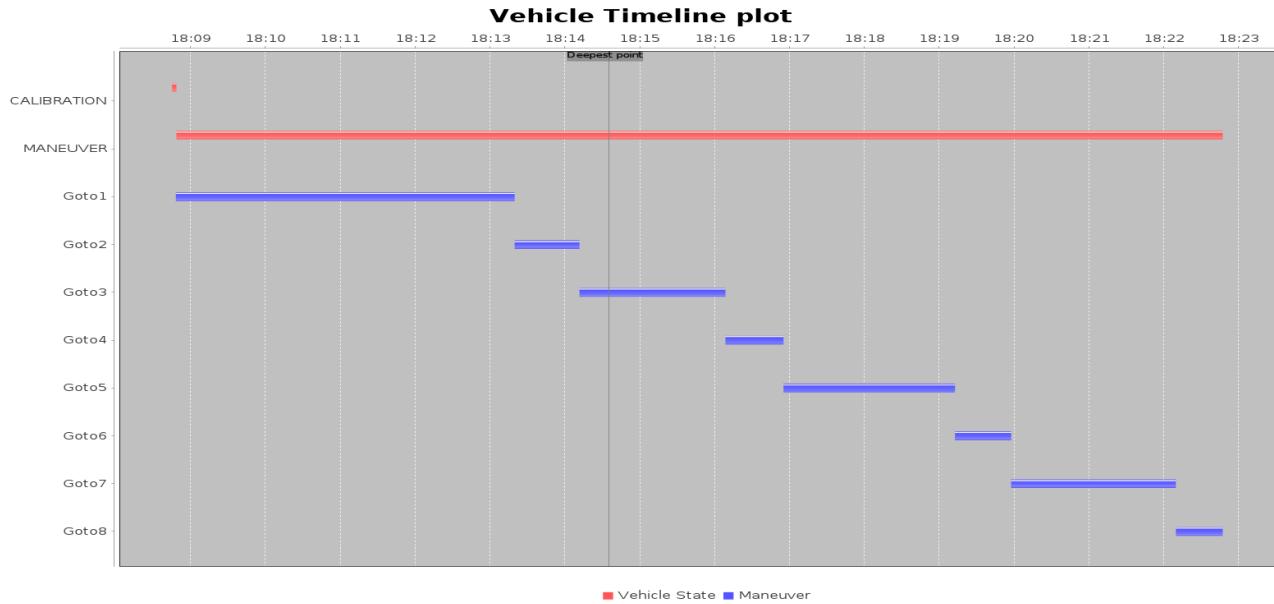
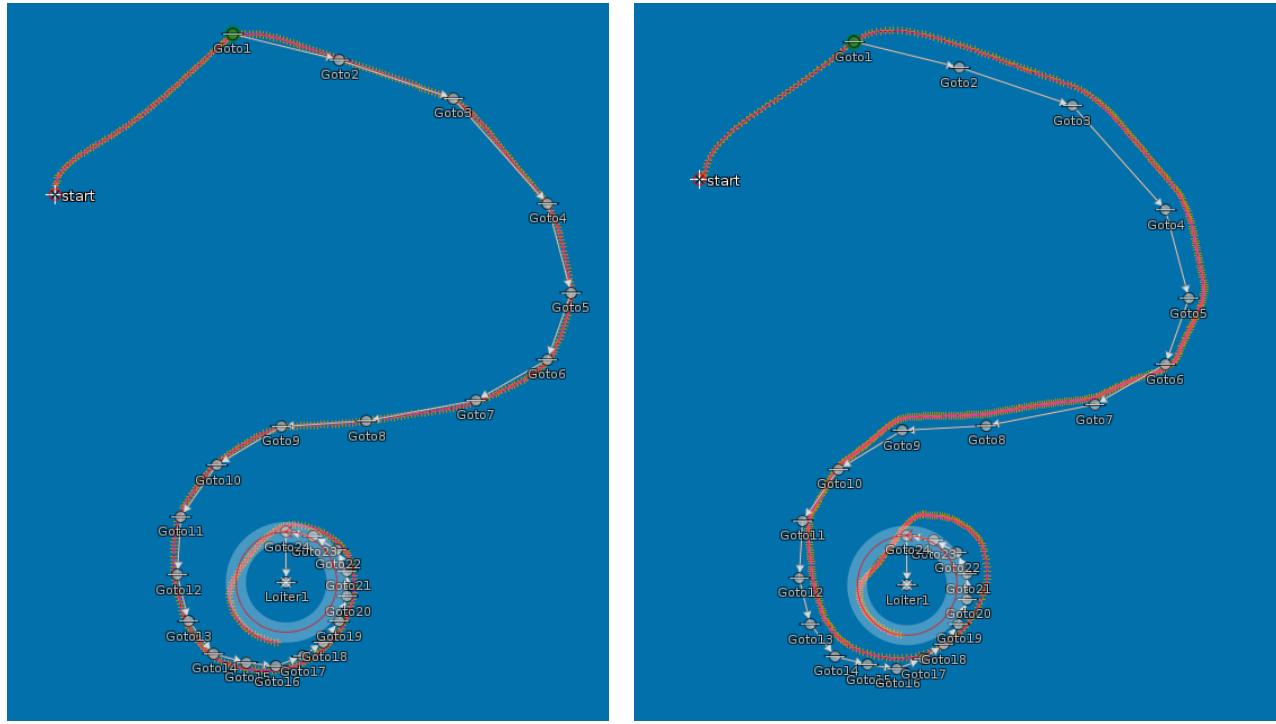


Figura 27: Distribuição temporal a executar cada subcomponente do plano (manobra).

6.3.2 Plano de Trajetória Curva

Este plano pretende simular um percurso de trajetória curva. A trajetória é constituída por pontos de referência consecutivos, em forma de espiral. Este plano constitui um teste de uma manobra mais complexa, com e sem corrente a nordeste. O vídeo da simulação encontra-se em <https://youtu.be/mcH5VpDI9XM>.

A Figura 28 ilustra a trajetória executada em relação ao plano original. A embarcação segue o caminho desejado, apesar das correntes, chegando ao ponto final seguindo uma trajectória em espiral. Em ambos os casos, verifica-se que as trajetórias conseguem ser cumpridas com alguma proximidade, havendo mais ruído na presença de correntes contraditórias. Esta comparação encontra-se na figura 28



(a) Trajetória em Espiral sem ruído

(b) Trajetória em Espiral com ruído

Figura 28: Trajetória efetivamente executada (a laranja) em comparação com a trajetória original planeada (a branco).

A Figura 29 demonstra a diferença entre a direção ideal para o percurso e a direção real da proa do USV. Como a trajetória descreve uma forma em espiral, o sistema apresenta a todo o momento um atraso em relação ao pretendido, por tentar mudar constantemente a direção.



Figura 29: Direção desejada ao longo do percurso (azul) em comparação com a direção (proa) ao longo do percurso (vermelho).

A Figura 30 apresenta o tempo que o sistema demorou a executar cada um dos troços do percurso, onde se constata que os pontos vão ficando cada vez mais próximos, refletindo-se num menor tempo para percorrer

os pontos ao longo do trajecto. A missão demorou um total de 8 minutos e 49 segundos, com uma velocidade média de 0.069 m/s (365.03 m de distância).

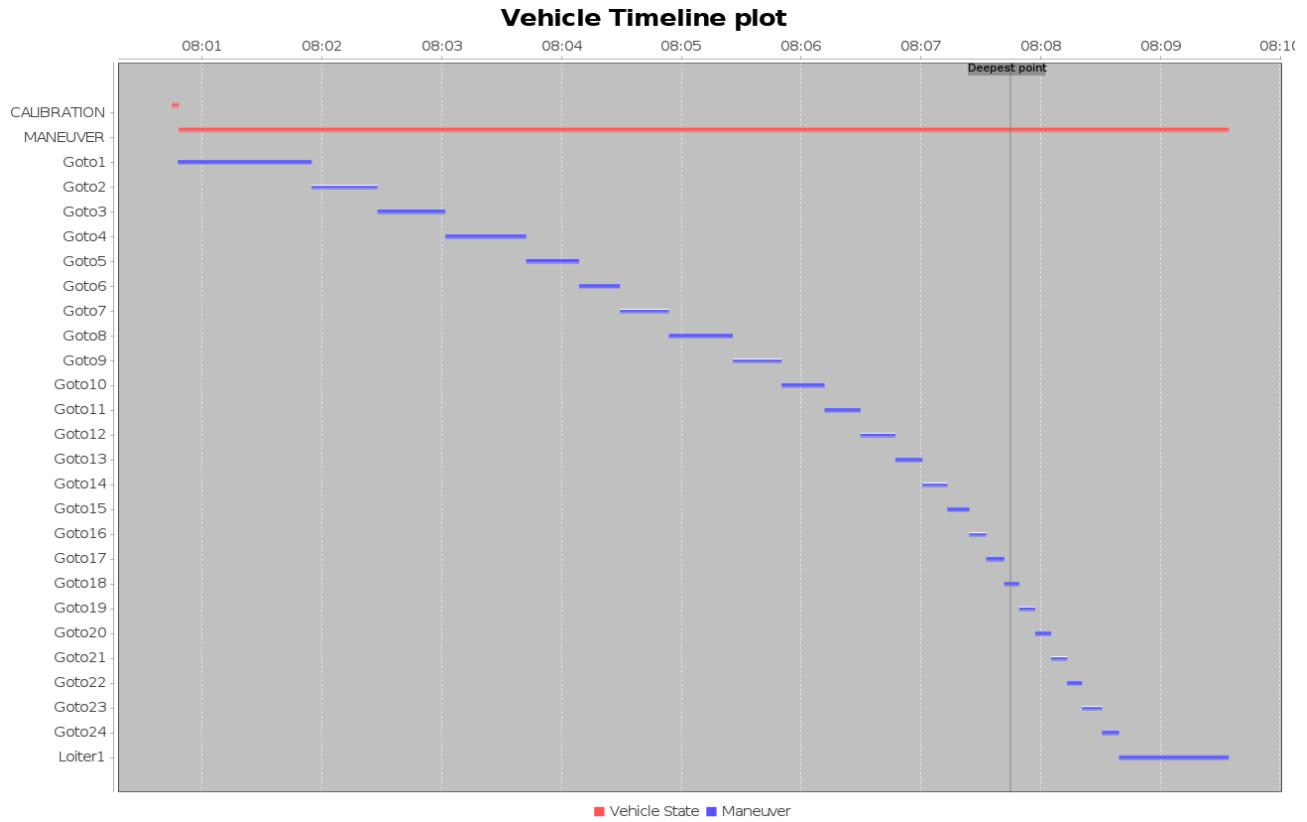


Figura 30: Distribuição temporal a executar cada subcomponente do plano (manobra).

6.3.3 Plano com variação de profundidade

O plano com variação de profundidade testa a capacidade de deteção de obstáculos abaixo do USV. Foi definida uma profundidade mínima de 4.2 metros no ficheiro de configuração 3.1.1, à qual o USV irá abortar a missão. O vídeo da simulação encontra-se em <https://youtu.be/WfcdVY6NcRU>. A Figura 31 ilustra as trajetórias planeada e executada, ficando claro que o USV aborta o seu percurso num dado momento. Efectivamente, este momento é quando é detetado um obstáculo a uma profundidade igual ou inferior ao limite definido.

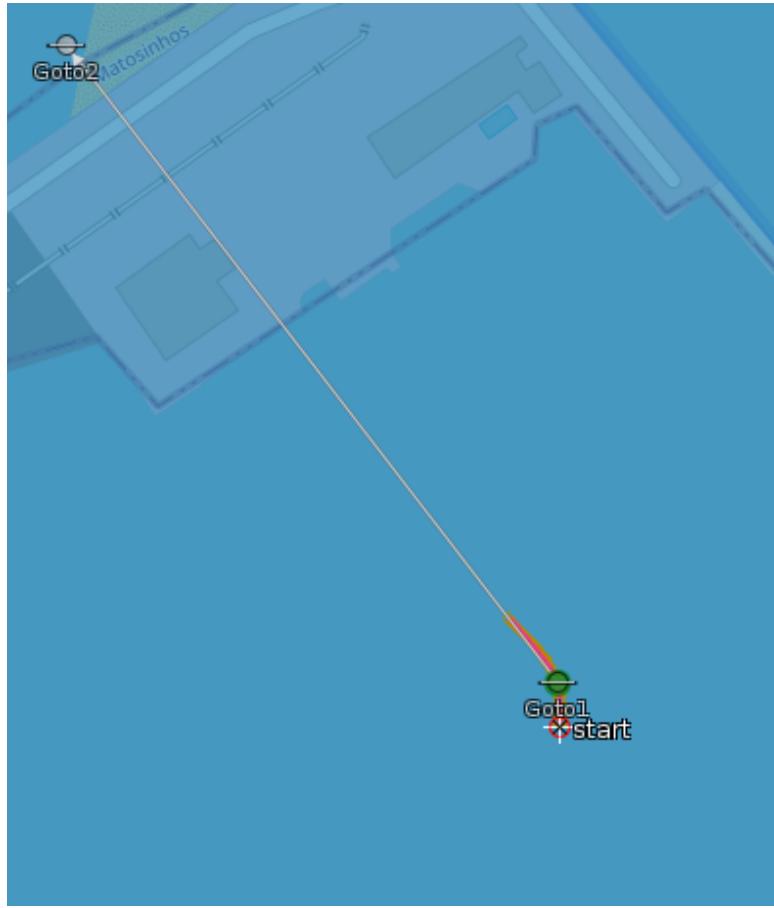


Figura 31: Trajetória efetivamente executada (a laranja) em comparação com a trajetória original planeada (a branco).

A Figura 32 apresenta o tempo que o sistema demorou a executar cada um dos troços do percurso. A missão demorou um total de 46.15 segundos, com uma velocidade média de 0.73 m/s (33.84 m de distância).

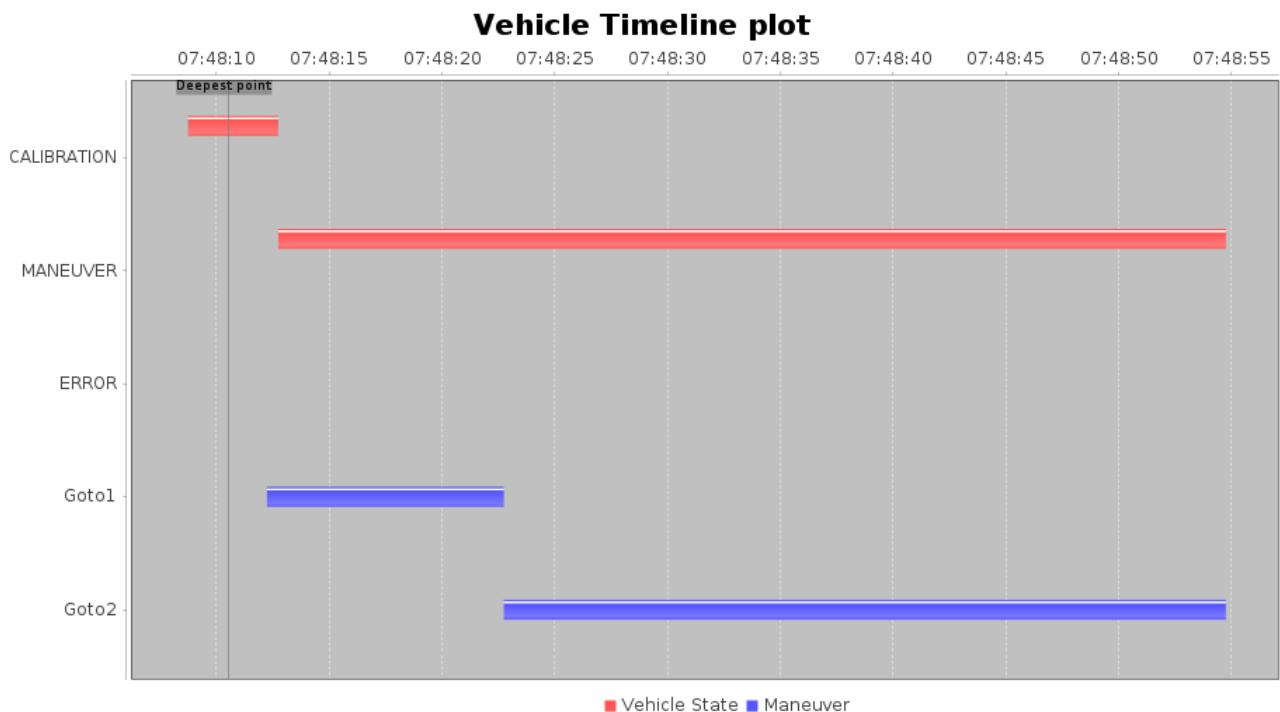


Figura 32: Distribuição temporal a executar cada subcomponente do plano (manobra).

Por fim, com o modelo batimétrico obtido ao longo do percurso do USV, apresentado na Figura 33, é possível verificar que missão terminou com a aproximação ao limite de profundidade definido, 4.2 metros. Este gráfico dá-nos ainda uma ideia do relevo submarino, ao inverter o mesmo horizontalmente.

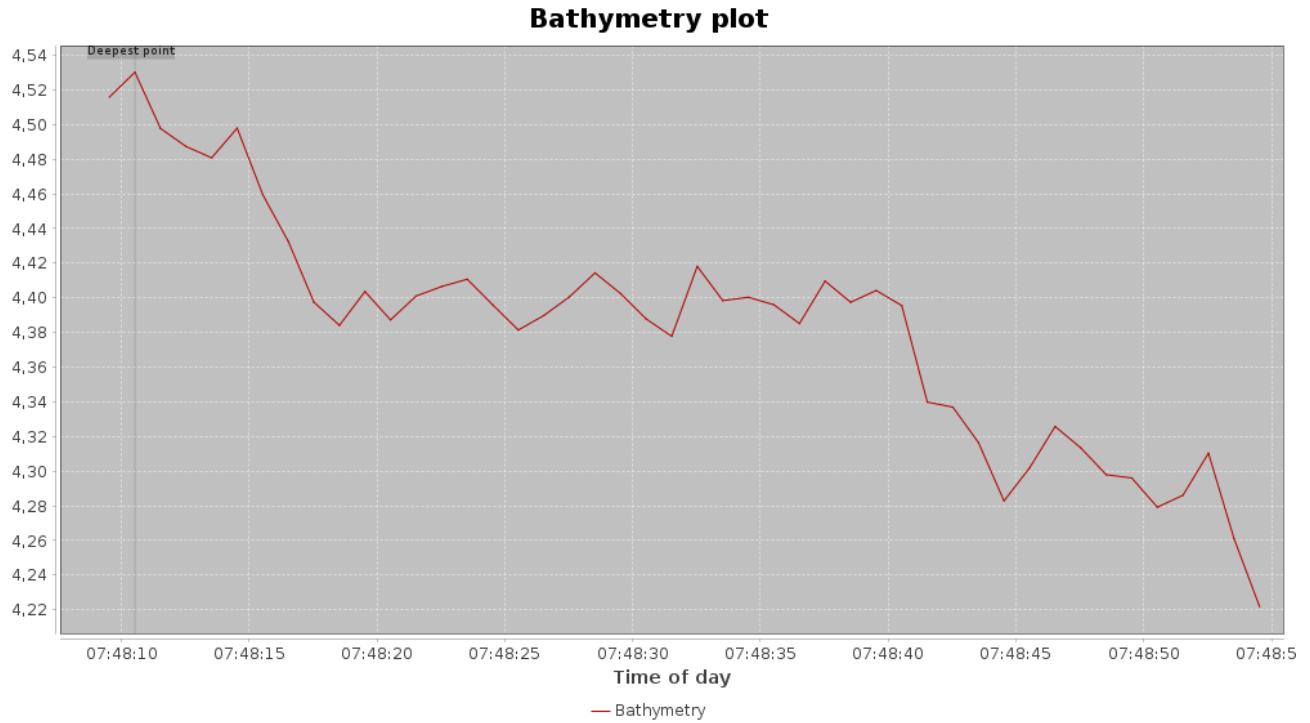


Figura 33: Batimetria ao longo do percurso

6.4 Simulação de Todo o Sistema

Para simular o sistema, utilizou-se um LT1000 ligado ao centro de processamento on board do LattePanda, que corre no DUNE. Por sua vez, o DUNE comunica com o Neptus e o ciclo culmina no controlo dos motores e dos lemes, representados respetivamente por LEDs e servos. A configuração utilizada encontra-se representada na Figura 34

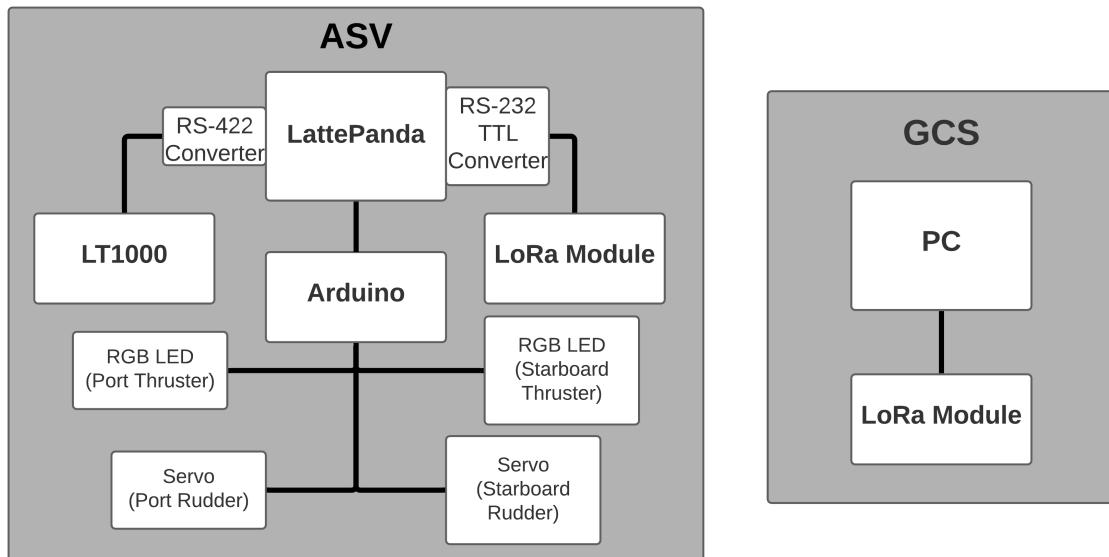


Figura 34: Configuração utilizada para teste de sistema completo.

Os vídeos da apresentação podem ser encontrados em: <https://sites.google.com/g.upporto.pt/seai-equipae>.

6.5 Dificuldades e Desvios face ao plano inicial

Desde o início do desenvolvimento, deparámo-nos com dificuldades que nos fizeram procurar outras alternativas e tentando minimizar o impacto no produto final. As dificuldades surgiram-se tanto a nível de software como de hardware.

Quanto ao software, verificou-se que a curva de aprendizagem e familiarização com a toolchain foi mais morosa do que o inicialmente esperado. As principais dificuldades surgiram da necessidade de ter exemplos práticos para a criação de veículos e explicitação de como os processos do DUNE interagiram entre si em background. Para minimizar o impacto, consultámos a documentação com os diagramas de classes das mensagens IMC, fizemos sessões de estudo da plataforma com vários elementos do grupo e tirámos dúvidas específicas com 2 membros do LSTS: a Eng^a Maria Costa e o Eng^o Paulo Dias, a quem reiteramos os agradecimentos.

Quanto ao hardware, deram-se atrasos na sua chegada (14/Dezembro). Para conseguirmos testar funcionalidades do sistema, procedeu-se com 2 planos de contingência:

1. procura e integração de um sistema de envio de mensagens NMEA (mencionado na secção 5.1.3) para a validação do módulo de sensorização, na ausência de um LT1000 real
2. emulamento dos atuadores para validação dos sinais de PWM com LEDs a simular os motores e servos a simular os lemes.

No caso dos LEDs, definiu-se um código de cores para a potência pedida aos motores (m), para cada canal RGB:

- $r = 100 - \text{abs}((m - 1500) / 5);$
- $g = (m > 1500) * (m - 1500) / 5;$
- $b = (m < 1500) * (1500 - m) / 5;$

Estas percentagens resultam em:

- vermelho- motor parado
- vermelho+verde- movimento para a frente
- verde- máxima potência para a frente
- vermelho+roxo- movimento para trás
- azul- movimento para trás à máxima potência

Para os servos, a variação de posição resulta na variação de posição dos eixos.

A foto 35 demonstra o setup encontrado para ultrapassar as dificuldades iniciais.

Após a receção do material (14/12), verificou-se que parte dos componentes não se encontravam operacionais. No que toca ao módulo de sensores LT1000, que apresentava sinais de corrosão por água salgada, foi deixado em álcool propílico um dia, em estufa mais umas horas e teve-se de soldar um condensador manualmente. Como após este tratamento a placa continuou sem dar sinal em nenhum dos pontos de teste, descartou-se a possibilidade de utilização do equipamento. Imagens do antes e do depois podem ser vistos nas Figura 36 e 37

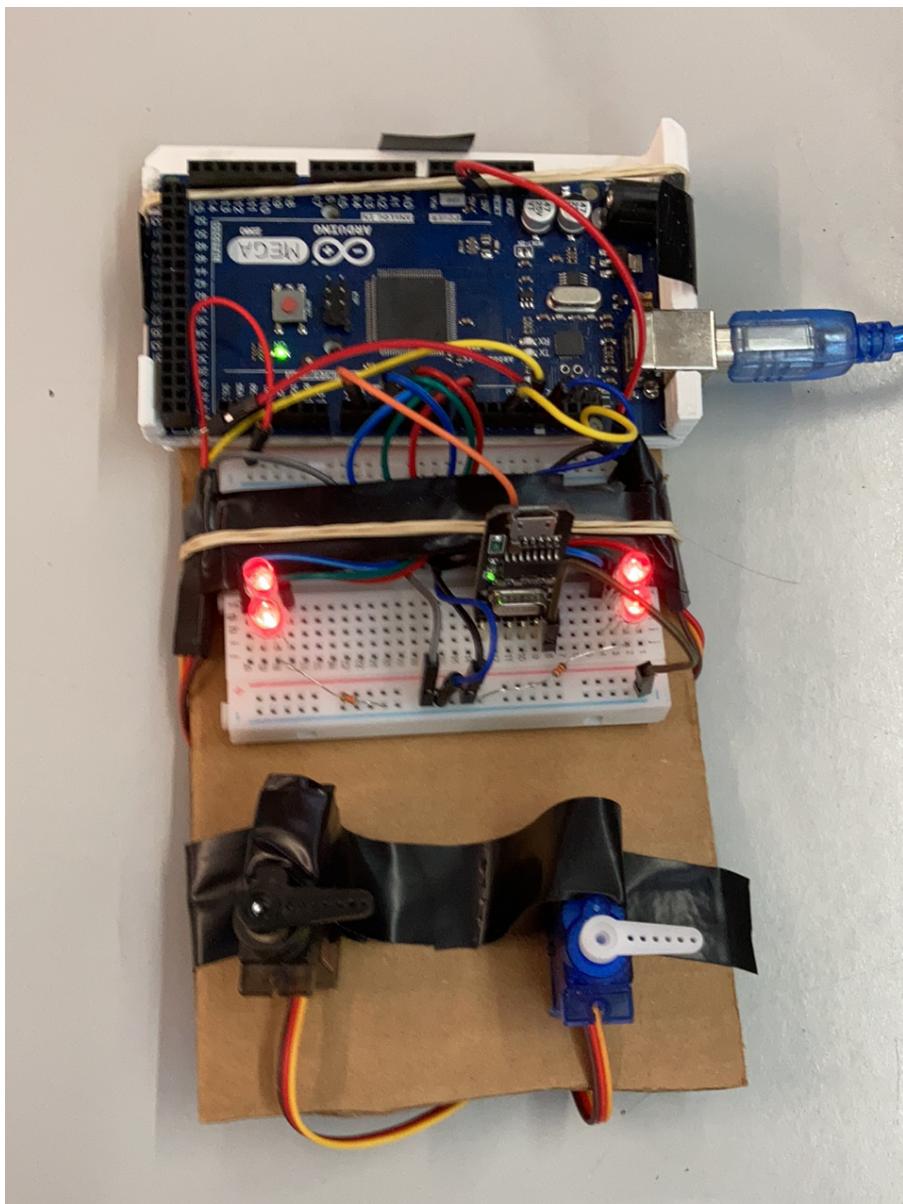


Figura 35: Setup dos atuadores



Figura 36: LT1000 - estado à chegada do material

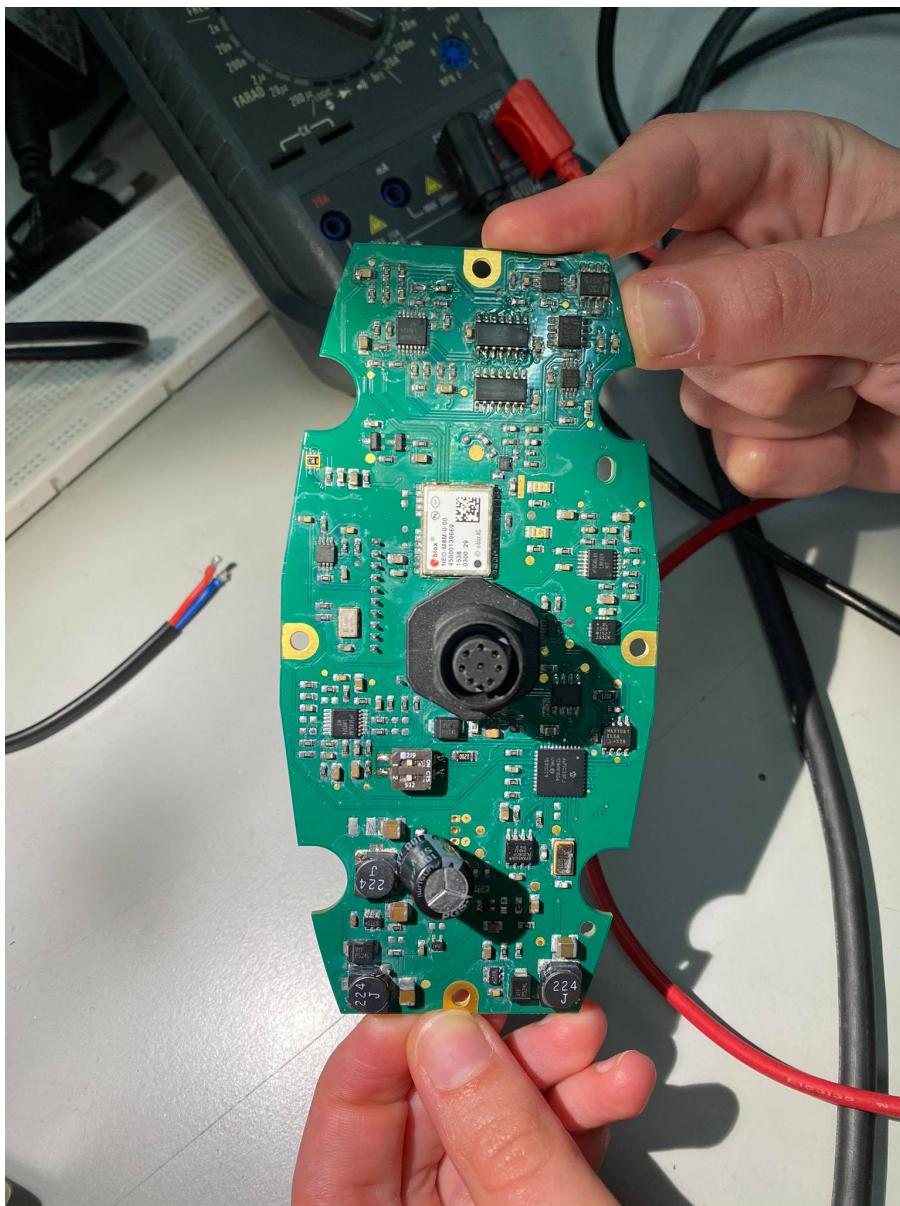


Figura 37: LT1000 - estado pós tentativa de reparação

No seguimento de danos por água salgada, um dos dois LattePanda enviados, um dos módulos de LoRa e o recetor de satélite também se encontravam queimados por danos com líquidos, o que atrasou o sistema tanto em horas de tentativa de reparação, procura de soluções alternativas e dependência de testes e validação.

Face às circunstâncias continuou-se o desenvolvimento com os simuladores dos planos de retaguarda, reportando o sucedido ao cliente, foi-nos disponibilizado material funcional na semana de 14 de Janeiro. Face aos prazos definidos e necessidade de escrita de documentação de suporte, uma semana não foi suficiente para completar a validação de todos os módulos em hardware, nomeadamente a integração completa das comunicações LoRa. A comunicação por satélite também não foi cumprida, em parte por dificuldades em obter uma licença, bem como a acumulação dos contratemplos encontrados. Não obstante, a equipa considera que a situação do satélite poderia ter sido abordada com emprego de mais homens-hora na tarefa.

7 Análise do Cumprimento dos Requisitos

Neste capítulo serão analisados os resultados e o desenvolvimento do projeto face aos requisitos estabelecidos numa fase inicial de planeamento. Cada requisito será caracterizado como "Completo" ou "Não completo", sendo devidamente justificado.

Os requisitos enumerados ao longo desta secção correspondem aos requisitos descritos na Lista de Requisitos, incluída no anexo C.

7.0.1 Requisitos do Cliente

Tabela 2: Requisitos do cliente.

Código de Requisito	Classificação	Estado
N1	Essencial	Cumprido
N2	Essencial	Cumprido
N3	Essencial	Cumprido
N4	Essencial	Parcial

- N1: Estando o sistema integrado no Neptus possui interoperabilidade com outros veículos da marinha que também estejam integrados no Neptus.
- N2: O sistema dirige-se para os pontos enviados pelo Neptus.
- N3: O sistema dirigi-se aos pontos enviados pelo caminho mais curto possível, tornando o veículo eficiente.
- N4: O sistema comunica por Wifi, permitindo uma comunicação de até 250 metros longe da costa. O LoRa não foi integrado no DUNE o qual permitia ter uma distância de comunicação de até 15km, embora esteja provado o conceito de funcionamento. O iridium (satélite) também não foi integrado no sistema o qual permitiria ter comunicações em alto mar.

7.0.2 Requisitos Funcionais

Tabela 3: Requisitos funcionais.

Código de Requisito	Requisito Associado	Classificação	Estado
RF1	N1	Essencial	Cumprido
RF2	N2	Essencial	Cumprido
RF3	N2	Essencial	Cumprido
RF4	N2	Essencial	Cumprido
RF5	N1	Essencial	Cumprido
RF6	N1, N2	Opcional	Cumprido
RF7	N1, N2	Opcional	Cumprido
RF8	N2	Opcional	Não Cumprido
RF9	N2	Opcional	Não Cumprido
RF10	N2	Opcional	Cumprido

- RF1: A GCS é capaz de enviar planos de missões para o USV.
- RF2: Na GCS é possível criar missões com vários pontos.
- RF3: O USV é capaz de executar um plano com vários pontos de passagem.
- RF4: Com base na analise de resultados dos sensores é possível ver que existe um erro médio de 6 metros na leitura dos sensores. A partir desta analise conclui-se que o sistema chega a um ponto de destino com uma tolerância inferior a 10 metros.
- RF5: Com o auxilio do IMC Monitor do Neptus verifica-se que o ceov-asv comunica com uma frequência de 40HZ, ou seja um período de 25ms. O período máximo acontece na inicialização do ceov-asv e é de 8 segundos. Cumprindo o requisito.
- RF6: A GCS é capaz de planejar missões complexas como trajetos curvos e manobras avançadas.
- RF7: O USV consegue executar missões complexas como trajetos curvos e algumas manobras avançadas como a manobra de barometria.
- RF8: Não foram abordadas as falhas de comunicação.
- RF9: A bateria não foi utilizada no projeto.
- RF10: O USV analisa a profundidade do local onde se encontra se for menor que um certo valor aborta a missão e segue outro caminho.

7.0.3 Requisitos de Performance

Tabela 4: Requisitos de performance.

Código de Requisito	Requisito Associado	Classificação	Estado
RP1	N2	Essencial	Cumprido
RP2	N1	Essencial	Cumprido
RP3	N2, N4	Essencial	Parcial
RP4	N3	Opcional	Cumprido
RP5	N2, N3	Opcional	Parcial
RP6	N3	Opcional	Não Cumprido

- RP1: O DUNE verifica o estado das tarefas o que permite uma maior fiabilidade e certeza no sistema.

- RP2: Com o auxilio do IMC Monitor do Neptus verifica-se que o ceov-asv comunica com uma frequência de 40HZ, ou seja um período de 25ms. O período máximo acontece na inicialização do ceov-asv e é de 8 segundos. Cumprido o requisito.
- RP3: Apenas foi testado o conceito do LoRa, não tendo sido implementado.
- RP4: Os lemes são empregues para mudanças de direção abruptas, conseguindo controlar-se pequenas variações pelos motores diferenciais.
- RP5: Pela arquitetura do DUNE, os waypoints são criados por interpolação da trajetória escolhida. Não se liga apenas aos pontos definidos, criando vários pontos intermédios sobre a linha.
- RP6: A bateria não foi utilizada no projeto.

7.0.4 Requisitos de Design

Tabela 5: Requisitos de design.

Código de Requisito	Requisito Associado	Classificação	Estado
RD1	N1	Essencial	Cumprido
RD2	N2	Essencial	Cumprido
RD3	N1	Opcional	Não Cumprido
RD4	N3	Opcional	Cumprido
RD5	N2	Opcional	Cumprido

- RD1: O USV foi integrado com a toolchain LSTS... ?
- RD2: Foi criado um ficheiro de configuração para o USV (ceov-asv.ini) onde é definido o seu perfil e as tarefas a este associadas.
- RD3: O AIS não foi integrado no projeto.
- RD4: Tendo em conta que o controlo dos atuadores é realizado com base na trajetória real, os fatores que originam a curva de cão são mitigados, corrigindo-se a trajetória à partida.
- RD5: Todas as funcionalidades desenvolvidas em software são independentes e modulares, podendo ser habilitadas ou não no ficheiro de configuração do USV através do parâmetro *Enabled* que pode tomar os valores *Hardware*, *Software*, *Always* e *Never*.

7.0.5 Requisitos Alocados

Tabela 6: Requisitos Alocados

Código de Requisito	Requisito Associado	Classificação	Estado
RA1	RF4	Essencial	Cumprido
RA2	RD1	Essencial	Cumprido
RA3	RD1	Essencial	Cumprido
RA4	RF3, RF7	Essencial	Cumprido
RA5	RF5	Essencial	Cumprido
RA6	RP3	Essencial	Não Cumprido
RA7	RP3	Essencial	Parcial
RA8	RP3, RP2	Essencial	Não Cumprido
RA9	RF4	Essencial	Cumprido
RA10	RF4	Essencial	Cumprido
RA11	RF10	Opcional	Cumprido
RA12	RF6	Opcional	Não Cumprido

- RA1: A interface dos sensores e dos atuadores foi possivel atraves do DUNE e do Neptus que possui a sua propria interface para planeamento e execução.
- RA2: Todo o software no LattePanda foi desenvolvido no DUNE.
- RA3: Todo o software na GCS foi desenvolvido no Neptus.
- RA4: Com recurso as tarefas dos Actuadores e com base num PID é possível seguir o caminho da melhor maneira para o ponto de destino.
- RA5: O Neptus já tem integrado mapas e cartas de navegação.
- RA6: A comunicação de satélite não foi implementada por falta de conta para utilizar.
- RA7: Apenas o conceito do LoRa foi testado, não foi implementado.
- RA8: Como as comunicações não funcionavam na totalidade, não foi se desenvolveu código para alterar entre a sua utilização.
- RA9: O LattePanda controla os motores e os lemes por intermédio do arduino, permitindo o catamarã seguir todos os waypoints.
- RA10: O LattePanda recebe os dados do LT1000 com uma cadência ajustável pelo utilizador sendo menor que 0.5 leituras/segundo se necessário.
- RA11: Com recurso a tarefa de batimetria é possível seguir um caminho seguro mesmo quando a profundidade é menor que um determinado valor.
- RA12:

8 KPI

Nesta secção são apresentados os KPI's (em inglês *Key Performance Indicator*) que foram usados durante o desenvolvimento do projeto. Os KPI's são valores mensuráveis que demonstram a eficácia dos objetivos. Uma monitorização dos KPI's ao longo do projeto permite uma intervenção rápida da equipa, para corrigir e reavaliar estratégias. Por exemplo se os prazos não estiverem a ser cumpridos, isso vai refletir-se nos indicadores, alertando a equipa para encontrar novas soluções. Assim sendo, estes indicadores são uma grande ferramenta para controlar e aumentar o desempenho da equipa.

Neste projeto os KPI's escolhidos foram divididos em KPI's de equipa e do projeto, os quais se encontram nas Tabelas 7 e 8, respectivamente.

8.1 KPI's de equipa

Os KPI's referentes à equipa encontram-se condensados na tabela 7, com o respetivo valor limítrofe na coluna (valor de referência). As fórmulas para o cálculo dos indicadores estão identificados nas equações 1 a 3.

Tabela 7: KPI's de equipa

Descrição do Indicador	Valor de referência
Presença do membros da equipa na reuniões	>85%
Percentagem de objetivos alcançados semanalmente	>80%
Percentagem de tarefas atrasadas	<15%

Presença dos membros da equipa nas reuniões

Este indicador dá *feedback* a equipa sobre o empenho e motivação de cada elemento.

$$P_p = \frac{N_r - F}{N_r} * 100 \quad (1)$$

P_p = Percentagem de presenças

N_r = Número de reuniões

F = Faltas

Percentagem de objetivos alcançados semanalmente

Este indicador permite ter uma visão geral do rendimento de cada elemento no projeto e da sua situação atual.

$$P_o = (1 - \frac{N_o - N_{oa}}{N_o}) * 100 \quad (2)$$

P_o = Percentagem de objetivos alcançados

N_o = Número de objetivos

N_{oa} = Número de objetivos alcançados

Percentagem de tarefas atrasada

Este indicador permite a equipa verificar as tarefas que não foram concluídas por cada membro, tentando minimizar os atrasos de tarefas que comprometam o resto do projeto.

$$P_{ta} = \frac{N_t - N_{tf}}{N_t} * 100 \quad (3)$$

P_{ta} = Percentagem de tarefas atrasadas

N_t = Número de tarefas

N_{tf} = Número de tarefas finalizadas

8.2 KPI's do Projeto

De igual forma, os KPI's referentes ao projeto encontram-se condensados na tabela 8, com o respetivo valor limítrofe na coluna (valor de referência). As fórmulas para o cálculo dos indicadores estão identificados nas equações 4 a 6.

Tabela 8: KPI's do Projeto

Descrição do Indicador	Valor de referência
Percentagem de requisitos essenciais concluídos	100%
Erro máximo do caminho planeado (metros)	20 m
Conseguir executar uma missão padrão, com cinco pontos de referência, até ao fim.	Qualitativo: (sim ou não)

Percentagem de requisitos essenciais concluídos

Este indicador permite verificar se os pedidos do cliente foram cumpridos.

$$P_{rc} = \left(1 - \frac{N_r - N_{rc}}{N_r}\right) * 100 \quad (4)$$

P_{rc} = Percentagem de requisitos concluídos

N_r = Número de requisitos

N_{rc} = Número de requisitos concluídos

Erro do caminho percorrido em comparação com o caminho planeado (centímetros)

Este indicador permite ter uma visão da percisão e fiabilidade do nosso sistema.

$$E_c = C_{pe} - C_{pl} \quad (5)$$

E_c = Erro no caminho percorrido

C_{pe} = Caminho percorrido

C_{pl} = Caminho planeado

Atraso entre as ações de nível inferior e sua monitorização (milissegundos)

Através deste indicador é possível analisar o tempo de resposta do sistema as alterações do ambiente.

$$A = I_m - I_{bn} \quad (6)$$

A = Atraso entre a ações e monitorização

I_{bn} = Intante de tempo de ação de baixo nível

I_m = Instante de tempo da monitorização

Parte III

Gestão de Projeto

Quanto à gestão do projeto, definiu-se uma organização de equipa, ferramentas de trabalho (WBS) e construiu-se um leque de ferramentas para a avaliação do estado dos desenvolvimentos, isto é lista de indicadores de performance e lista de riscos e seus impactos. Desta forma, pretendeu-se agilizar o trabalho individual e a integração das partes no projeto final, ultrapassando dificuldades e culminando no cumprimento dos requisitos definidos.

9 Organização da equipa

A equipa é constituída por 9 membros cada um com um papel definido para o decorrer do projeto. Os membros e os respetivos papéis são apresentados de seguida:

Membro	Papel
André Teixeira	Desenvolvedor
Carlos Pinto	Líder
Fernando Silva	Desenvolvedor
Francisco Oliveira	Desenvolvedor
João Santos	Coordenador de Comunicação
João Fernandes	Revisor de Documentação
Leonor Santos	SubLíder
Miguel Lousada	Facilitador
Ricardo Martins	Secretário

Cada papel tem inerente a si, um conjunto de responsabilidades que são especificadas a baixo:

- **Líder:** É responsável por organizar e planejar as reuniões, fazer a divisão de tarefas, definir prazos e é o Porta-Voz da equipa.
- **SubLíder:** Auxilia o líder e é responsável pelas suas tarefas na ausência do mesmo.
- **Coordenador de Comunicação:** É responsável pela comunicação com qualquer elemento externo à equipa e comunicar as informações que advém de terceiros para a equipa.
- **Secretário:** Responsável por anotar informações relevantes em qualquer reunião e pela escrita das atas.
- **Revisor de Documentação:** Responsável por aprovar os documentos de índole informativa, fazer correções que considere pertinentes e garantir que os documentos são submetidos nos devidos repositórios atempadamente.
- **Facilitador:** Responsável por garantir a participação de todos os membros nas reuniões, avaliar a participação e progresso da equipa como um todo e resolver conflitos entre membros da equipa.
- **Desenvolvedor:** Responsável por realizar as tarefas que lhe são atribuídas com qualidade expectável ao projeto e dentro dos prazos estabelecidos para a realização das mesmas.

Apesar de alguns membros terem papéis individuais, todos os membros realizam o papel de desenvolvedor. Salienta-se que, devido à desistência de um dos elementos do grupo, foi necessária a redistribuição das tarefas pelos elementos do grupo, resultando na organização indicada

9.1 Subequipas

De modo a estabelecer uma divisão de tarefas que permitisse o desenvolvimento eficiente das mesmas a equipa foi dividida em 3 subequipas, cada uma com um foco principal no desenvolvimento do projeto. De seguida é apresentada a configuração de cada subequipa e o foco associado:

Subequipas		
Atuação	Sensorização	Comunicações
Carlos Pinto	Fernando Silva	André Teixeira
João Santos	Leonor Santos	João Fernandes
Francisco Oliveira	Ricardo Martins	Miguel Lousada

A subequipa de Atuação concentrou-se no desenvolvimento das tarefas DUNE e Arduino para o controlo efetivo dos motores e lemes. A subequipa de sensorização focou-se no desenvolvimento da tarefa de interpretação das medidas. A subequipa de Comunicações tratou da interligação dos dispositivos GCS-Processamento a Bordo com Wifi, Satélite e LoRa.

A distribuição de tarefas de cada sub-equipa por cada membro da mesma foi da responsabilidade dos membros que a integram. Ainda se salienta que embora o foco de cada equipa seja o mencionado, em pontos do projeto houve necessidade de empregar horas noutras equipas para culmatar os atrasos sentidos. A documentação inicial, nomeadamente Manual de Qualidade, Análise de Mercado, Requisitos e Conceitos do Sistema foi dividido pelos membros. Quanto ao relatório final e documentação, ficou ao encargo da subequipa de Sensorização. A demonstração geral foi atribuída à subequipa de Atuação.

10 WBS

No que toca à metodologia de trabalho, explicitado na Figura 38 nomeadamente a Work Breakdown Structure (WBS), segmentou-se em 6 áreas distintas: gestão de projeto, sistemas de engenharia, sistemas on-board de controlo e interfaces, sistemas on-board de planeamento e execução, sistemas em terra e, por último, testes e validação.

Na gestão de projeto inclui-se toda a documentação necessária tanto para o bom funcionamento da equipa e também como registo dos desenvolvimentos realizados. Alguns desses documentos elaborados incluem os relatórios, atas e documentação de software. Para o controlo de versão de software, utilizou-se a plataforma GitHub, visto ser conhecido por todos os membros e fácil de utilizar em todos os sistemas operativos. Os repositórios, igualmente incluídos nesta secção, serviram como local de organização e partilha de todos os documentos à parte de código. Neste caso, escolheu-se o Google Drive por ser um serviço oferecido pela faculdade e também amplamente conhecido pelos membros. Por fim, os papéis dos membros da equipa, mencionados na secção 9, são importantes para garantir as funcionalidades e definir bem que tarefas são necessárias realizar e por quem.

Quanto à engenharia de sistemas incluímos todos os documentos pedidos e que permitem um melhor conhecimento do sistema: a análise de requisitos, estudo de mercado, culminando no desenvolvimento do conceito de sistema. Todos estes documentos permitem fundamentar tanto o processo de design e decisões de implementação, aquando das reuniões de equipa. Desenvolveram-se ainda documentação específica de cada tarefa e o presente relatório final.

As seguintes secções do WBS coincidem com as milestones do projeto. A primeira Milestone *Functional System* refere-se ao funcionamento offline do USV: controlar a direção e velocidade ao nível do veículo. A segunda Milestone *Mission Control* refere-se ao plano de controlo: definição dos diversos waypoints e controlo com o objetivo da chegada a cada um deles. A terceira Milestone denomina-se *Neptus Command*, significando o envio e monitorização de missões por parte dos sistemas em Terra.

Por fim, a última componente da Work Breakdown Structure é a componente de testes e validação, sempre presente na fase de desenvolvimento, de forma a aferir o estado e comportamento do sistema, em diversos instantes do que está presente durante todo o decorrer do projeto.

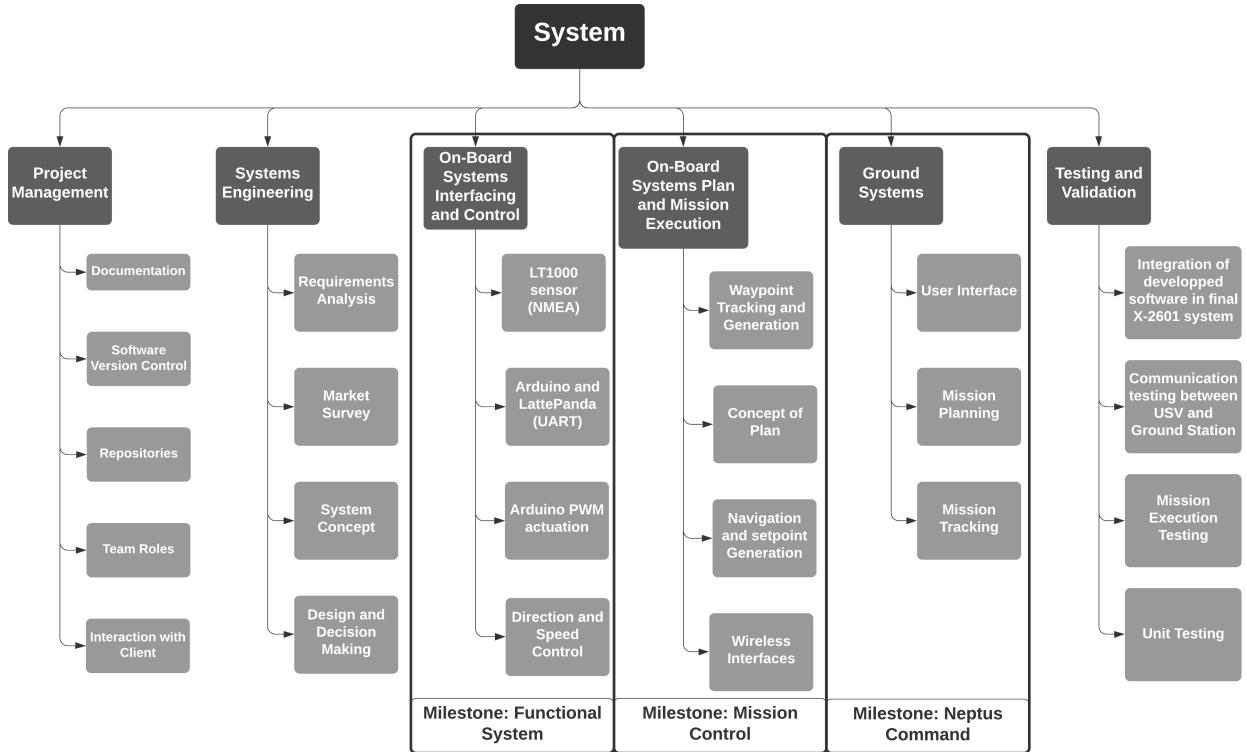


Figura 38: Diagrama do Work Breakdown Structure

10.1 Distribuição temporal de tarefas

Considerou-se que uma representação gráfica das tarefas a realizar seria crucial para o bom funcionamento da equipa, pelo que desenvolveu-se um Gantt para melhor planear todo o desenvolvimento. Esse diagrama temporal está indicado na Figura 39.

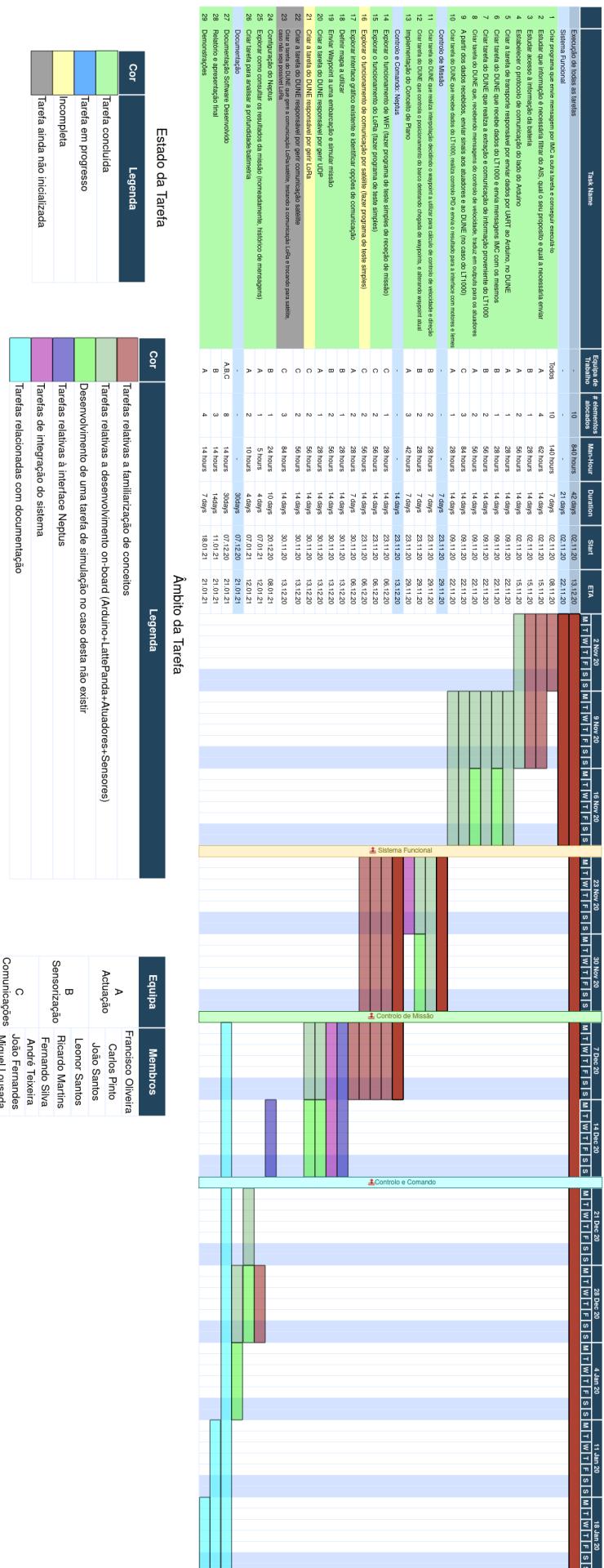


Figura 39: Diagrama de Gantt do desenvolvimento do projeto

As tarefas dividem-se em 4 grandes grupos: familiarização com o sistema, montagem da test bed, implementação do sistema de controlo de bordo e conexão com a interface Neptus. Na Figura 39, as tarefas distinguem-se pelas cores diferentes. Ainda se destacam 3 milestones diferentes, que marcam a o término de uma nova funcionalidade do sistema.

11 Orçamento

O orçamento, com o objetivo de assegurar a viabilidade económico-financeira do projeto, está representado na Tabela 9. Dadas as especificidades do projeto, a maior fatia do orçamento refere-se à mão de obra dividida em tarefas e subtarefas. A conversão da estimativa de horas para um valor concreto é simples e direta, tendo em conta o preço do mercado.

Para este projeto, foi estimado um total de 840 homens-horas de trabalho, divididas pelos 9 membros do grupo. Tendo em conta a estimação de homens-hora, facilmente se obtém um orçamento estipulado, para um dado preço de mercado em euros por hora. Por exemplo, assumindo um preço de mercado, para um engenheiro licenciado, como 8€/hora e tendo em conta o total homens-hora apresentado na Figura 39, pode assumir-se um total de 6720€ na componente de mão de obra.

Relativamente à testbed, os custos assentam nos componentes necessários para a realização do projeto. Os componentes utilizados e respetivos custos foram:

- LattePanda - 182,84€
- Módulo de LoRa - 27,90€
- LT1000 NRU - 1493,34€
- Driver Para Motor DC - 121,87€
- Servo Digital - 89,90€

Prefazendo um total de 1915,85€. Todo o material acima listado é uma contribuição da Marinha e será emprestado pelo CEOV, no âmbito deste projeto.

Estimou-se um custo de viagem (que comporta o combustível e portagens para três carros) até à base de Alfeite, de 300€. O alojamento seria uma contribuição da Marinha. Para as refeições estima-se um valor 1350€, assumindo 3 refeições diárias, com um custo médio de 5 euros, durante os 10 dias. No caso deste orçamento ser apresentado pela FEUP, seriam cobrados overheads de projeto no valor de 25%, perfazendo um total de 12857,31€.

Tabela 9: Orçamento estimado do Projeto

Rúbrica	Iten	Valor
Equipamento	LattePanda	182,84€
	Módulo LoRa	27,90€
	LT1000 NRU	1493,34€
	Driver Para Motor DC	121,87€
	Servo Digital	89,90€
Recursos Humanos	Mão de Obra	6720€
Deslocações	Transportes	300€
Consumíveis	Refeições	1350€
Subtotal		10285,85€
Outras Despesas		2571,46€
Total		12857,31€

12 Gestão de Riscos

A análise de risco consiste na avaliação da probabilidade de ocorrência de um contratempo e no cálculo do seu possível impacto e prejuízo para a equipa. A avaliação da eventualidade de ocorrência de um evento e das suas possíveis consequências é essencial para estabelecer uma estratégia de prevenção que priorize os riscos mais prováveis ou potencialmente mais graves. A fim de minimizar impactos de imprevistos foi elaborada a Tabela 10 com exemplos de riscos, juntamente com as respetivas estimativas dos seus impactos e possíveis soluções na eventualidade de estes ocorrerem.

Tabela 10: Riscos

Risco	Probabilidade	Severidade	Mitigação	Milestones Afetadas
Falha na entrega do hardware	Baixa	Crítica	Comunicação com o cliente e FEUP	1 e 2
Falha nas interfaces do software	Baixa	Crítica	Testes unitários ao software em questão	1
Comunicação com o cliente insuficiente	Baixa	Crítica	Estabelecimento de diversos meios de comunicação com o cliente	Afeta todo o projeto
Perda do trabalho ou partes do mesmo	Baixa	Crítica	Uso de diferentes locais de armazenamento de informação e repositórios	Afeta todo o projeto
Erro nas comunicações GSC-ASV	Média	Séria	Testes independentes dos meios de comunicação	3
Dificuldades no uso da toolchain	Média	Séria	Comunicação com pessoas que conhecem o seu funcionamento	Afeta todo o projeto
Impossibilidade de testar fisicamente o sistema	Média	Séria	Utilização de simuladores para validação de resultados	Afeta todo o projeto
Erro no seguimento do caminho	Média	Severa	Simulações e analises	1
Falta de coordenação no repositório de software	Média	Severa	Incluir a publicação do trabalho feito no github como algo mandatório a fazer nas reuniões semanais	Afeta todo o projeto
Atraso de uma tarefa	Média	Média	Rearranjo de subequipes e aumento de carga semanal	Afeta todo o projeto
Perda de um elemento da equipa	Baixa	Média	Distribuição de tarefas de forma flexível e facilmente ajustável	Afeta todo o projeto

13 Metodologia de Teste: Bottom-up Testing

Ao fazer-se o desenvolvimento do projeto desde a atuação ao planeamento da missão e, consequentemente teste da solução sequencial, considerou-se uma abordagem do tipo **bottom-up** para o teste e integração da solução. Na metodologia *bottom-up* considera-se que o fluxo de teste (e desenvolvimento) segue do baixo nível do sistema para as camadas hierárquicas superiores. A vantagem deste tipo de abordagem, face aos existentes, prende-se com 2 aspetos do sistema em questão: mitigar os erros que ocorrem mais frequentemente na camada inferior e garantir a funcionalidade básica do sistema, aumentando sucessivamente as suas capacidades.

Com esta metodologia, à medida que as tarefas estão devidamente concluídas e validadas, caminha-se no sentido de dotar o USV de capacidades e gradualmente melhorar a sua performance como veículo autónomo.

Seguindo a metodologia *bottom-up*, o desenvolvimento é mais uniforme e integrado, o que não seria possível de acordo com outras abordagens, nomeadamente *top-down*. A lógica de controlo e comando do sistema não seria possível de implementar sem as funcionalidades básicas garantidas, muito menos validadas. Por outro lado, não se seguiria uma abordagem de desenvolvimento ágil, visto que não tínhamos versões operacionais do produto ao longo das implementações. A longo prazo, não se beneficiando nem o cliente nem progredindo para a sua solução.

Posto isto, considera-se que a abordagem *bottom-up*, identificada na figura 40 é a que se enquadra melhor no caso do X-2601, alinhando-se com as necessidades sugeridas e o seu desenvolvimento consistente.

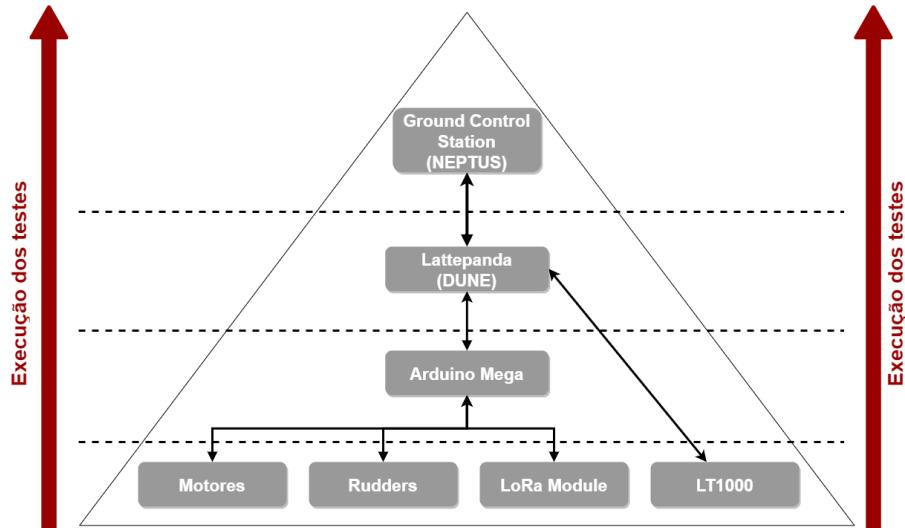


Figura 40: Ilustração da metodologia Bottom-up no projeto

14 Execução do Projeto

No que diz respeito ao desenvolvimento prático do projeto, verifica-se que todas as tarefas das *milestones*, documentação e validação estão concluídas, à exceção das comunicações sem fios. As comunicações estão concluídas no que diz respeito ao WiFi e existe a prova de conceito do LoRa, integrado já com o DUNE.

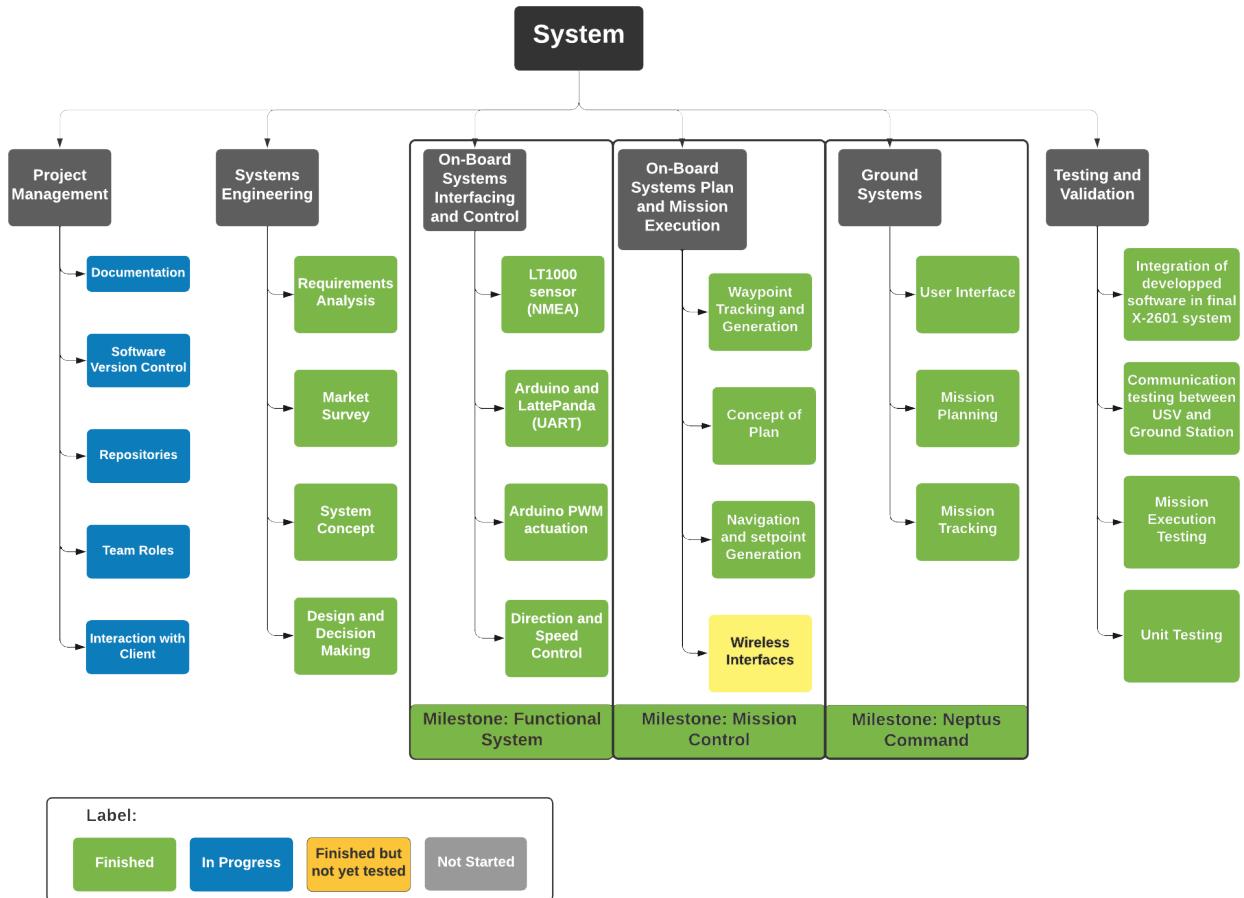


Figura 41: Diagrama do WBS com execução das tarefas

15 Interação com o cliente

O cliente é a secção CEOV, divisão da Marinha Portuguesa, com intermediário o Tenente Castro Fernandes.

A interação com o cliente ficou planeada com uma frequência semanal, com possibilidade de ajuste, para esclarecer dúvidas específicas de hardware e necessidades. Adicionalmente mantém-se mais 2 vias de comunicação: repositório de documentação no Google Drive para partilha de Datasheets e chat para dúvidas rápidas e pontuais que não requeiram reunião por vídeo conferência.

Salienta-se que o CEOV se disponibilizou a fornecer todo o material necessário para realização de testes, nomeadamente os módulos de Hardware, a quem reiteramos os agradecimentos.

Foi-nos ainda solicitado pelo cliente, a criação de documentação específica sobre a solução de software desenvolvida, de forma a tornar mais claros os desenvolvimentos e agilizar futuros melhoramentos.

16 Avaliação de equipa

A avaliação de equipa foi feita recorrendo ao Google Forms. Cada elemento avaliou cada um dos restantes membros da equipa numa escala de -2 a 2, considerando que a soma das suas avaliações teria de ser igual a 0 ou 1. Os resultados encontram-se na tabela :

Tabela 11: Avaliações de equipa

Elemento	Avaliação
André	0
Carlos	0.875
Fernando	0.25
Francisco	-0.125
João Marcelo	0
João Vasco	-0.625
Leonor	0.5
Miguel	-0.5
Ricardo	0.375

Referências

- [1] Figura para diferenciar os angulos.
- [2] Figura para diferenciar cog e sog.
- [3] Manual de produto da unidade de referência à navegação lt1000.
- [4] Toolchain.
- [5] Notícia referente ao x-2601, pela marinha.
- [6] Arduino mega 2560.
- [7] Lora, módulo industrial anti-interferência wireless data transceiver module.
- [8] Manual de produto do transdutor ais.
- [9] Decsrição técnica para o driver dos motores syren.
- [10] Descrição técnica para o driver dos lemes savox servo.
- [11] Datasheet do módulo lattepanda, com 4gb/64gb e licença empresarial.
- [12] Toolchain lsts dune.
- [13] Toolchain lsts neptus.
- [14] Documentação dune: Gpsfix.
- [15] Documentação dune: Euler angle.
- [16] Simulador nmea.
- [17] Documentação dune: Estimated state.
- [18] Documentação dune: Desiredheading.
- [19] Documentação dune: Desiredspeed.
- [20] Documentação dune: Abort.
- [21] Documentação dune: Controlloops.
- [22] Documentação dune: Setservoposition.
- [23] Documentação dune: Setthrusteractuation.

Anexos

Anexo A Manual de Qualidade

Faculdade de Engenharia da Universidade do Porto



**Sistema de Planeamento e Controlo de Execução para Veículo de
Superfície Não Tripulado - Manual de Qualidade**

Equipa E

Orientador João Sousa

Líder da Equipa Carlos Pinto

Manual de Qualidade realizado no âmbito da unidade curricular
Sistemas de Engenharia - Automação e Instrumentação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

24 de novembro de 2020

Registo de Versões

Versão	Data	Descrição
0.1	05/10/2020	Google Docs: Outline básico do Documento (criação da estrutura)
0.2	06/10/2020	Google Docs: Protótipo de conteúdos de cada secção
0.3	06/10/2020	Google Docs: Revisão em equipa dos conteúdos de cada secção
0.4	10/10/2020	Migração do Documento para L ^A T _E X(Overleaf)
0.5	11/10/2020	Inclusão das secções de Listas de Figuras e Listas de Tabelas Adição de regras e Work Tools
0.6	12/10/2020	Revisão final

Índice

1 Objetivo do Manual	3
2 Composição da Equipa	3
3 Organização Interna	3
3.1 Organização Funcional	3
3.2 Sub-Equipas	4
4 Regras	4
4.1 Regras Gerais	4
4.2 Regras de Qualidade de Trabalho	5
4.3 Regras de Decisão	5
4.4 Regras de Gestão de Conflitos	6
4.5 Regras de Reuniões	6
4.5.1 Marcação	6
4.5.2 Funcionamento	6
4.5.3 Ata	7
5 Work Tools	7
5.1 Google Drive/Docs	7
5.2 Overleaf	7
5.3 GitHub	7
5.4 Discord	7
5.5 Messenger	7
5.6 Zoom	8
5.7 Doodle	8
6 Avaliação	8
7 Templates	8
Lista de Tabelas	18
Lista de Figuras	19

1 Objetivo do Manual

O presente manual de qualidade foi escrito no âmbito da unidade curricular de Sistemas de Engenharia Automação e Instrumentação. Este documento tem como objetivo estabelecer um conjunto de normas e princípios a seguir pelos membros da equipa ao longo da realização deste projeto. Para além disso, estão definidos os cargos de cada membro, seguido de uma breve explicação da função de cada um.

Este manual pode sofrer modificações, que contribuam para um melhor desenvolvimento do projeto, caso a equipa considere necessário e após um acordo entre todos.

2 Composição da Equipa

A composição da equipa, bem como os respetivos contactos e especializações, consta na tabela 1

Tabela 1: Composição da equipa, contactos de cada elemento e sua respetiva especialização.

Nome	E-mail	Especialização
André Costa Teixeira	up201605590@fe.up.pt	Automação - Eletrónica e Sistemas
Carlos Manuel Santos Pinto	up201605439@fe.up.pt	Automação - Microeletrónica e Sistemas Embarcados
Fernando Jorge Pinheiro da Silva	up201604125@fe.up.pt	Automação - Gestão Industrial
Francisco Xavier Gomes Oliveira	up201604934@fe.up.pt	Automação - Microeletrónica e Sistemas Embarcados
João Marcelo Casanova Almeida Tomé Santos	up201605317@fe.up.pt	Automação - Robótica e Sistemas
João Vasco Gonçalves da Costa Fernandes	up201603219@fe.up.pt	Automação - Gestão Industrial
Leonor Baptista da Costa Silva Santos	up201504515@fe.up.pt	Automação - Robótica e Sistemas
Miguel Amorim Aroso Lousada	up201603379@fe.up.pt	Automação - Robótica e Sistemas
Ricardo André Sanches Martins	up201608378@fe.up.pt	Automação - Robótica e Sistemas
Ricardo de Paula Pinto Sousa Ribeiro	up2104304373@fe.up.pt	Automação - Eletrónica e Sistemas

3 Organização Interna

Optou-se por dividir a equipa a dois níveis: organização funcional, estabelecendo responsabilidades de índole mais permanente, não dependentes do projeto em si ou das tarefas específicas a desempenhar, e sim em aspectos mais universais (nomeadamente: documentação, validação de trabalho, motivação, entre outros); e em organização em sub-equipas, sendo que cada sub-equipa será responsável por desempenhar uma tarefa, inserida contextualmente no projeto, e terá uma constituição e responsabilidades que serão dinâmicas, isto é, serão alteradas consoante o que for preciso desenvolver no projeto, e consoante os pontos fortes e fracos de cada elemento da equipa.

3.1 Organização Funcional

Foram estabelecidos 6 papéis a desempenhar: líder, sub-líder, secretário, coordenador de comunicação, revisor de documentação, e facilitador. Seguem-se as responsabilidades de cada papel, bem como o elemento da equipa a quem esse papel foi atribuído.

- Líder - Carlos Pinto. Responsável por: organizar as reuniões (quando irão ocorrer; tópicos a discutir); divisão de tarefas entre os elementos do grupo; definir prazos; servir como porta-voz da equipa em relação à comunicação com o orientador de projeto.
- Sub-Líder - Leonor Santos. Responsável por: tomar o cargo de líder quando este estiver indisponível; auxiliar o líder nas suas tarefas.
- Secretário - Ricardo Martins. Responsável por: apontar informações relevantes em qualquer reunião; escrita das atas.
- Coordenador de Comunicação - João Santos. Responsável por: comunicar com qualquer elemento externo à equipa, exclusive o orientador de projeto, e comunicar as informações que advém de terceiros para a equipa; garantir que os membros da equipa estão a atualizar o estado das suas tarefas no GitHub (ver secção 5).
- Revisor de Documentação - João Fernandes. Responsável por: aprovar os documentos de índole informativa (relatórios, atas, etc.) e fazer correções que considere pertinentes; garantir que os documentos são submetidos nos devidos repositórios atempadamente.
- Facilitador - Miguel Lousada. Responsável por: garantir a participação de todos os membros nas reuniões; avaliar a participação e progresso da equipa como um todo, expondo problemas que encontre; resolver conflitos entre membros da equipa; em caso de argumentação entre membros da equipa, garantir que esta é feita de forma estruturada e civilizada.

3.2 Sub-Equipas

As sub-equipas serão organizadas consoante as tarefas a desempenhar. Quando uma tarefa é completada, a sub-equipa à qual a tarefa foi atribuída é redistribuída para uma nova tarefa (ou uma tarefa já pertencente a outra sub-equipa, aumentando o número de elementos de equipa ativos para essa tarefa). Cabe ao líder reintegrar os membros da equipa em sub-equipas apropriadas consoante os pontos fortes e fracos de cada elemento.

4 Regras

Nesta secção são apresentadas as regras básicas de funcionamento da equipa de modo a definir um modo de funcionamento objetivo, de forma a que a equipa possa trabalhar em concordância.

4.1 Regras Gerais

As regras a seguir apresentadas destinam-se a promover uma boa comunicação, inclusão e divisão de tarefas entre todos os membros da equipa:

- Todos os membros deverão conhecer e seguir todas as regras expostas neste documento;
- Todos os membros deverão estar sempre disponíveis a serem contactados em pelo menos uma das seguintes formas e verificar as mesmas, pelo menos, uma vez por dia:
 - Telemóvel;
 - Email;
 - Chat de equipa em redes sociais (ver secção 5);
- Todos os membros deverão ter conhecimento de todas as atualizações do projeto, sendo o papel do Líder, Sub-líder ou Facilitador garantir que isso aconteça;
- A cada membro deverá ser dada a oportunidade de expressar de forma sucinta, ideias e opiniões de caráter relevante para o desenvolvimento do Projeto e tarefas a ele relacionadas;

- Aquando da atribuição de uma tarefa, cada elemento deverá atualizar o respetivo “card” presente no quadro de projeto do Github da equipa de acordo com o desenvolvimento dessa mesma tarefa;
- Qualquer documentação de caráter relevante ao desenvolvimento do projeto deverá ser submetido de forma organizada no diretório do Google Drive da equipa;
- Qualquer desenvolvimento de software relevante ao desenvolvimento do projeto deverá ser submetido de forma organizada no GitHub da equipa;
- Cada tarefa deverá ser entregue de forma completa até ao prazo estabelecido pela equipa, excetuando-se eventuais casos de dificuldades técnicas ou pessoais que impeçam a entrega da mesma, as quais deverão ser reportados ao Líder, Sub-líder e/ou Facilitador;
- Todos os membros deverão ser assíduos e não exceder o limite de tolerância de 15 minutos no que diz respeito à pontualidade, exceto em casos de impossibilidade, que deverão ser reportados previamente ao Líder, Sub-líder e/ou Facilitador e onde poderá ser discutida uma alternativa viável à presença do(s) elemento(s).

4.2 Regras de Qualidade de Trabalho

As regras a seguir apresentadas destinam-se a promover que cada elemento cumpra, não só com as suas responsabilidades, mas que contribua para o projeto com maior qualidade:

- Cada elemento é responsável por garantir a devida qualidade do seu próprio trabalho;
- Caso algum elemento considere que a qualidade do trabalho seja inferior ao exigido, deverá comunicá-lo ao Líder, Sub-líder e/ou Facilitador;
- Qualquer atraso ou dificuldade sentida por parte de um membro da equipa deverá ser comunicado, de imediato, ao Líder, Sub-líder e/ou Facilitador;
- Caso 4 ou mais elementos da equipa concordem que determinada tarefa não apresenta o nível de qualidade exigido, então estes elementos deverão explicitar em que aspectos a tarefa falhou, e os devidos responsáveis pela realização da mesma deverão agir de forma a satisfazer as exigências.
- Consoante a disponibilidade dos elementos da equipa, poderão ser realocados membros de umas tarefas para outras, consoante uma tarefa demore muito tempo a desenvolver-se e comece a apresentar-se como um *bottleneck*.

4.3 Regras de Decisão

As regras a seguir apresentadas destinam-se a descrever como serão feitas as decisões que influenciem diretamente o produto final:

- Decisões que influenciem diretamente o funcionamento do produto final deverão ser decididas em equipa, por meio de voto;
- Nenhum membro deve fazer omitir decisões que fez no seu trabalho individual, ainda que estas sejam, aparentemente, irrelevantes para o projeto no seu todo. Tais decisões deverão ser comunicadas na reunião semanal intra-equipa (ver secção 4.5.1).
- A decisão deve ser tomada de acordo com o maior número de votos. No caso da votação incidir num tópico evidentemente pertencente ao domínio de uma sub-equipa, os elementos da sub-equipa terão maior impacto de voto (votam por dois);
- As decisões deverão ser tomadas em reunião de equipa;
- No caso de empate de votos, deverá ser discutida a decisão, de forma a atingir um consenso;
- Contactar-se-á o coordenador, como último recurso, de forma a obter conhecimento para proporcionar uma escolha mais informada.

4.4 Regras de Gestão de Conflitos

As regras a seguir apresentadas destinam-se a promover um ambiente de trabalho saudável e produtivo:

- Se um elemento não estiver de acordo com uma qualquer decisão deverá comunicar à equipa ou escrever nos canais de comunicação intra-equipa para que se resolva o mais rapidamente possível;
- Caso ocorra uma divergência de opiniões, deverá ser realizada uma votação entre todos os membros presentes. Em caso de empate, cabe ao facilitador a resolução do problema;
- Em caso do facilitador estar envolvido no conflito, o seu papel deverá ser feito pelo líder;
- Como último recurso, deverá recorrer-se ao coordenador para auxiliar na resolução do conflito.

4.5 Regras de Reuniões

As regras a seguir apresentadas destinam-se a definir a marcação de reuniões, o funcionamento das mesmas, e a sua documentação.

4.5.1 Marcação

- Por omissão, estará marcada uma reunião intra-equipa semanal predefinida todas as segundas-feiras, a dar início às 21h30m. Esta reunião tem como objetivo partilhar o que foi desenvolvido ao longo da semana entre os elementos da equipa (avanços, dificuldades, questões, ...), bem como o que se planeia realizar nessa semana;
- Haverá uma reunião semanal entre a equipa e o coordenador, que deverá ser da iniciativa da equipa, e que terá lugar, tirando eventuais exceções que deverão ser comunicadas, às sextas-feiras, a dar início entre as 16h15m e as 16h45m. Apenas metade da equipa (a decidir pelo Líder) estará fisicamente presente nestas reuniões, para evitar grandes agregados face à situação de COVID-19.
- Qualquer reunião deverá preceder com o envio atempado (mínimo 12 horas antes da hora prevista) de uma agenda com os tópicos que se pretendem discutir, e hora de início de reunião, pelo chat utilizado pela equipa (no caso de ser uma reunião intra-equipa) ou por email (reunião que envolva membros não integrantes da equipa, nomeadamente, o orientador);
- No caso das reuniões terem de ser alteradas, deverá ser feito, se possível com 1 dia de antecedência, num horário a combinar através de uma votação (poll).

4.5.2 Funcionamento

- Nas reuniões semanais, todos os membros devem, individualmente: abordar o que foi realizado durante a semana; comunicar passos seguintes; falar sobre eventuais problemas encontrados.
- Indo de encontro com as regras de qualidade de trabalho, os membros deverão discutir a integração do trabalho de cada um no contexto geral do projeto, validando se os progressos estão a ser eficientes.
- O facilitador terá o papel de supervisionar e avaliar a integração dos membros na equipa e comunicar em caso de interação de má qualidade;
- As decisões mais relevantes para o ciclo de vida do projeto terão de ser efetuadas em reunião.
- Da reunião deverá surgir um plano de trabalho (com tarefas distribuídas) a curto prazo, nomeadamente para a semana seguinte. De igual forma, deverá ser elaborado um relatório sucinto dos progressos feitos no projeto, próximos passos e dificuldades sentidas (se alguma), a enviar por email ao orientador do projeto, Professor João Sousa, juntamente com o convite da reunião semanal equipa-orientador.

4.5.3 Ata

De todas as reuniões deverá ser elaborado um documento em formato de ata, para relatar os temas tratados. A ata seguirá as regras de template infra mencionados. Os temas a incluir: membros presentes, agenda da reunião, exposição e revisão do trabalho realizado, temas concluídos na reunião, temas da agenda que não se concluíram e agendamento da próxima reunião.

5 Work Tools

Para garantir uma gestão eficiente dos recursos humanos é necessário utilizar diversas ferramentas de trabalho. Nas seguintes secções estas ferramentas vão ser enumeradas, sendo descritas as suas principais aplicações.

5.1 Google Drive/Docs

Google Drive é um serviço onde rapidamente se conseguem partilhar ficheiros entre toda a equipa. O Google Docs é um serviço que tem as funcionalidades necessárias para a criação rápida de documentos como atas e permite a edição simultânea e consulta por vários elementos da equipa.

Assim, foram criadas as seguintes pastas: "Documentation", onde estão contidos o relatório final e os manuais de qualidade, e onde podem ser adicionados outros documentos com estas características; "Templates", onde se encontram os templates para se fazer as atas ou para as avaliações do trabalho; Tasks, onde rapidamente se pode responder às tarefas que sejam necessárias ao trabalho; "Presentations", onde se desenvolvem as apresentações e por fim "Meetings" onde se encontram as atas e documentos que derivam das reuniões.

5.2 Overleaf

O Overleaf é uma plataforma de produção partilhada de documentos L^AT_EX, sendo intuitiva e prática. Na elaboração de documentos mais extensos, será utilizado o Overleaf para se trabalhar concorrentemente com outros elementos da equipa, de forma a produzir um documento que utilize o espaço de forma mais eficiente, e que permita a referenciamento automática de fontes bibliográficas.

5.3 GitHub

O GitHub é uma ferramenta essencial para desenvolvimento de software. Funciona como um repositório com funcionalidades dedicadas que permitem: manter um controle de versões (git), planejar e distribuir tarefas e rastrear bugs.

Desta forma, nas reuniões semanais, são criados issues e atribuídos aos elementos responsáveis pela sua realização e manutenção do estado no project management (To do, In progress, Reviewing, Done). Assim, a utilização do GitHub não se restringe a desenvolvimento de software, mas também a manter informação sobre o estado de qualquer tarefa a desenvolver pela equipa.

5.4 Discord

O Discord é uma plataforma de comunicação através de canais dedicados a temas, onde é possível partilhar texto, imagens e fazer reuniões com todos os elementos da equipa. Com estas funcionalidades, foram usadas 3 categorias: informações, canais de texto e canais de voz. Nas informações encontram-se as localizações das outras ferramentas de trabalho, bem como qualquer nota essencial ao bom funcionamento da equipa. Os canais de texto incluem vários sub-canais como geral, planeamento de sessões, e tópicos não relacionados. Por fim, como canal de voz temos as reuniões intra-equipa.

5.5 Messenger

O Messenger é um serviço de mensagens rápidas, que é principalmente utilizado para discussões breves, atualizações pontuais cuja divulgação por todos seja necessária e votações.

5.6 Zoom

O zoom é um serviço que permite fazer reuniões, e a sua utilização restringe-se principalmente a reuniões importantes, como por exemplo a reunião de introdução do grupo.

5.7 Doodle

O Doodle é um serviço de votação, o qual será usado pela equipa como calendário online, onde cada elemento pode colocar as suas disponibilidades, facilitando assim a marcação de reuniões.

6 Avaliação

De forma a avaliar o trabalho, comportamento e a responsabilidade de cada membro da equipa, foram estabelecidos cinco componentes de avaliação com pesos atribuídos, como consta na tabela 2.

Tabela 2: Componentes de avaliação individual dos elementos do grupo, seus pesos, e breve descrição.

Designação	Peso (%)	Descrição
Cumprimento de objetivos	35	Execução das tarefas de acordo com o que é proposto ou esperado - ser eficaz
Responsabilidade	25	Assiduidade, pontualidade e cumprimento das regras
Comunicação	15	Capacidade de comunicar as suas ideias e opiniões com os restantes elementos da equipa
Qualidade de trabalho	15	Execução das tarefas da melhor maneira possível e com o mínimo de erros - ser eficiente
Participação	10	Contribuição e participação ativa durante as reuniões da equipa

Cada elemento da equipa deve avaliar os restantes individualmente, atribuindo uma nota numa escala de -2 a 2 para cada componente descrito. Adicionalmente, a média da equipa deve ser 0.

A avaliação interna ocorrerá de duas em duas semanas, com prazo de um dia após a reunião intra-equipa semanal, com o intuito de promover o bom desempenho de cada elemento e de permitir identificar possíveis dificuldades. Esta deverá também ser feita de forma não anónima, com o objetivo de haver partilha de perspetivas em relação à quantidade e qualidade do trabalho realizado, bem como ao cumprimento dos objetivos sugeridos.

7 Templates

Nesta secção são apresentados os dois templates principais a utilizar para a redação de Atas e documentos mais extensos (o presente Manual de Qualidade já segue este template). As atas serão efetuadas no Google Docs consoante o template apresentado na figura 1, que se encontra na Google Drive da equipa, sob a pasta "Templates". Documentos mais extensos, que serão entregues ao orientador do projeto, seguirão o template apresentado nas figuras 2, 3, 4, 5, 6, 7, 8 e 9, cujo link para a respetiva página web do Overleaf está à disposição de todos os elementos da equipa. Para um dado documento, no caso da total ausência de tabelas, figuras ou referências, a respetiva secção será omitida.



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Título

(ex: Ata de Reunião de Equipa - Semana 1, 1^a

Reunião)

Data: Local:
Membros Presentes:
Membros Ausentes:
Hora de Início: Hora de Final:
Notas:

Agenda

Nesta reunião pretende-se abordar os seguintes tópicos:

1. A
2. B
3. ...

Revisão Semanal

O que cada membro fez, dificuldades sentidas e eventuais soluções encontradas para resolver essas dificuldades.

Tópicos Não Concluídos

Tópicos da agenda que não foram concluídos e uma breve descrição do porquê. Se um tópico tiver sido abordado e fez-se algum progresso para atingir uma conclusão/solução, deverá ser descrito o progresso feito, bem como o que falta resolver.

Tópicos Concluídos

Tópicos que foram discutidos e que se conseguiu chegar a conclusões/soluções. Deverá ser descrito como se chegou à conclusão/solução (por votos, alguém responsável pelo tópico tomou uma decisão, etc.)

Próxima Reunião

A próxima reunião está agendada para as 00:00 do dia 00/00/0000 na sala X000/através da plataforma Discord/no local X.

Figura 1: Template a seguir para a redação de atas.

Faculdade de Engenharia da Universidade do Porto



FEUP

**Sistema de Planeamento e Controlo de Execução para Veículo de
Superfície Não Tripulado - Template de Documento**

Equipa E

Orientador João Sousa

Líder da Equipa Carlos Pinto

Manual de Qualidade realizado no âmbito da unidade curricular
Sistemas de Engenharia - Automação e Instrumentação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

11 de outubro de 2020

Figura 2: Capa do template de documentos mais extensos.

Registo de Versões

Versão	Data	Descrição
0.1	dd/mm/aaaa	Descrição concisa de alterações feitas na versão 0.1 do documento
0.2	dd/mm/aaaa	Descrição concisa de alterações feitas na versão 0.2 do documento
...

André Teixeira, Carlos Pinto, Fernando Silva, Francisco Oliveira, João Santos, João Fernandes, Leonor Santos, Miguel Lousada, Ricardo Martins, Ricardo Ribeiro

1

Figura 3: Registo de versões do template de documentos mais extensos.

Índice

1 Objetivo do Documento/Introdução	3
2 Secção 1	3
2.1 Secção 1.1	3
2.1.1 Secção 1.1.1	3
3 Secção 2	3
A Apêndice 1	4
A.1 Sub-categoria do apêndice 1	4
B Apêndice 2	4
Lista de Tabelas	5
Lista de Figuras	6
Referências	7

André Teixeira, Carlos Pinto, Fernando Silva, Francisco Oliveira, João Santos, João Fernandes, Leonor Santos, Miguel Lousada, Ricardo Martins, Ricardo Ribeiro 2

Figura 4: Exemplo de índice do template de documentos mais extensos.

1 Objetivo do Documento/Introdução

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2 Secção 1

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.1 Secção 1.1

 Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.1.1 Secção 1.1.1

 Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

3 Secção 2

 Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Figura 5: Exemplo de corpo do template de documentos mais extensos.

A Apêndice 1

A.1 Sub-categoria do apêndice 1

B Apêndice 2

André Teixeira, Carlos Pinto, Fernando Silva, Francisco Oliveira, João Santos, João Fernandes, Leonor Santos, Miguel Lousada, Ricardo Martins, Ricardo Ribeiro 4

Figura 6: Exemplo de apêndices do template de documentos mais extensos.

Lista de Tabelas

André Teixeira, Carlos Pinto, Fernando Silva, Francisco Oliveira, João Santos, João Fernandes, Leonor Santos, Miguel Lousada, Ricardo Martins, Ricardo Ribeiro 5

Figura 7: Lista de Tabelas do template de documentos mais extensos.

Lista de Figuras

André Teixeira, Carlos Pinto, Fernando Silva, Francisco Oliveira, João Santos, João Fernandes, Leonor Santos, Miguel Lousada, Ricardo Martins, Ricardo Ribeiro 6

Figura 8: Lista de figuras do template de documentos mais extensos.

Referências

André Teixeira, Carlos Pinto, Fernando Silva, Francisco Oliveira, João Santos, João Fernandes, Leonor Santos, Miguel Lousada, Ricardo Martins, Ricardo Ribeiro 7

Figura 9: Referencias do template de documentos mais extensos.

Lista de Tabelas

1	Composição da equipa, contactos de cada elemento e sua respetiva especialização.	3
2	Componentes de avaliação individual dos elementos do grupo, seus pesos, e breve descrição. .	8

Lista de Figuras

1	Template a seguir para a redação de atas.	9
2	Capa do template de documentos mais extensos.	10
3	Registo de versões do template de documentos mais extensos.	11
4	Exemplo de índice do template de documentos mais extensos.	12
5	Exemplo de corpo do template de documentos mais extensos.	13
6	Exemplo de apêndices do template de documentos mais extensos.	14
7	Lista de Tabelas do template de documentos mais extensos.	15
8	Lista de figuras do template de documentos mais extensos.	16
9	Referencias do template de documentos mais extensos.	17

Anexo B Análise de Mercado

Faculdade de Engenharia da Universidade do Porto



Sistema de Planeamento e Controlo de Execução para Veículo de Superfície Não Tripulado - Market Survey

Equipa E

Orientador João Sousa

Líder da Equipa Carlos Pinto

Market Survey realizado no âmbito da unidade curricular
Sistemas de Engenharia - Automação e Instrumentação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

24 de novembro de 2020

Registo de Versões

Versão	Data	Descrição
0.1	16/10/2020	Criação do documento base segundo o template definido
0.2	17/10/2020	Contribuição individual de todos os membros para as respetivas secções
0.3	17/10/2020	Revisão em equipa
0.4	19/10/2020	Alteração da Segmentação de Mercado
0.5	19/10/2020	Revisão final pelo coordenador de documentação

Índice

1 Objetivo do Market Survey	3
1.1 Metodologia de Pesquisa	3
1.2 Limitações	3
2 Visão Geral do Sistema	3
2.1 Hardware do protótipo	4
2.2 Sistemas e subsistemas da arquitectura de controlo	5
2.2.1 On-Board	5
2.2.2 Software Instalado	6
2.3 Sistemas de controlo a implementar e integrar	7
3 Análise de Mercado	8
3.1 Mercado Potencial	8
3.2 Segmentação de Mercado	8
3.3 Dimensão de Mercado	9
3.4 Dinâmica de Mercado	9
4 Fornecedores / Competição	9
4.1 L3 ASV	10
4.2 Kongsberg	10
4.3 OceanAlpha	10
4.4 Sea Machines	11
4.5 Liquid Robotics	11
4.6 MARTAC	11
4.7 Análise geral dos fornecedores	11
5 Conclusões	12
Lista de Figuras	13
Referências	14

1 Objetivo do Market Survey

Este documento tem como objetivo adquirir uma perspetiva em relação a veículos não tripulados de superfície (*Unmanned Surface Vehicles - USVs*), a sua valorização no mercado, bem como algumas soluções já existentes. Pretende-se, portanto, decompor o nosso sistema e analisar a sua viabilidade no mercado.

O nosso projeto consiste no desenvolvimento de software que irá controlar a execução de um catamaran, convertendo-o num USV, com potencial alterações a nível de hardware. O âmbito do nosso projeto é, portanto, o de conversão de veículos, e não a produção de USVs. No entanto, particularmente para a análise de mercado - secção 3 - olhou-se para o mercado de USVs como um todo, sem particularizar o mercado de conversão de veículos normais (tripulados) em USVs, devido a alguma falta de informação em relação a este último.

1.1 Metodologia de Pesquisa

Para a realização deste documento, juntaram-se vários artigos das bibliotecas digitais: *IEEE, ACM, Research Gate e Science Direct*. Estes artigos foram analisados na tentativa de adquirir um melhor conhecimento do modo de operação típico destes sistemas, bem como a sua utilidade.

Com o âmbito de obter informação do mercado e de outras soluções, pesquisaram-se também bases de dados de *Market Surveys*, juntando informação de diversas fontes para obter uma perspetiva global e o mais correta possível.

No que diz respeito à análise da Competição e de fornecedores atualmente em Mercado, recorremos a companhias citadas em *Market Surveys* assim como a companhias apresentadas por um Website próprio para a pesquisa de fornecedores na área em questão, *Unmanned Systems Technology* [1].

1.2 Limitações

Este documento foi feito exclusivamente a partir da informação de terceiros, isto é, não se fez um estudo do mercado de forma direta (contactando empresas, especialistas, etc.). Desta forma, toda a informação presente neste documento está dependente da veracidade das fontes utilizadas. As bases de dados de *Market Surveys* só foram consultadas superficialmente, já que uma consulta dos próprios *Market Surveys* envolvia preços impraticáveis que não se justificam para o âmbito deste projeto, resultando em informação sobretudo qualitativa, com poucas métricas. Outro fator limitante à pesquisa, foi a quantidade de informação disponibilizada pelas companhias que atuam na área em relação aos produtos de interesse e às suas características.

2 Visão Geral do Sistema

A necessidade de realizar operações com um nível elevado de complexidade, longevidade e precisão levou ao desenvolvimento de veículos marítimos autónomos. Os campos de aplicação variam desde o apoio à autoridade nacional de proteção civil a estudos hidrográficos.

Assim, o projeto consiste na integração do software do Laboratório de Sistemas e Tecnologia Subaquática (LSTS) num catamarã X-2601 da marinha portuguesa. O sistema X-2601 é um sistema de prototipagem rápida de veículos de superfície autónomos que já se encontra operacional, e tem as funcionalidades básicas de navegação funcionais. A arquitetura desenvolvida pelo LSTS, que já está implementada em outros veículos da marinha, permite coordenar, planear, controlar e monitorizar operações. A junção destes dois sistemas é particularmente interessante pois permite acrescentar rapidamente as funcionalidades a um veículo já existente por um baixo custo e num prazo temporal baixo.

2.1 Hardware do protótipo



Figura 1: Protótipo X-2601 e os seus módulos de hardware

Acima está ilustrado os componentes de hardware do protótipo a converter.
No mastro multi-propósito encontra-se:

- Módulo gerador para alimentação do sistema de 3k W
- Carregador e baterias
- Modulo comando e controlo (módulo C3)
- Sensores

Em paralelo com o mastro é possível encontrar o módulo IH que controla o sonar de feixe simples e o sonar de feixe lateral, colocados em baixo do mesmo.

O movimento e direção do sistema é dado por 2 motores elétricos e 2 lemos.

Dimensões do sistema:

- Comprimento: 4m
- Boca (largura): 2m
- Calado: 60 cm

2.2 Sistemas e subsistemas da arquitectura de controlo



Figura 2: Sistema e subsistemas da arquitetura de controlo

2.2.1 On-Board

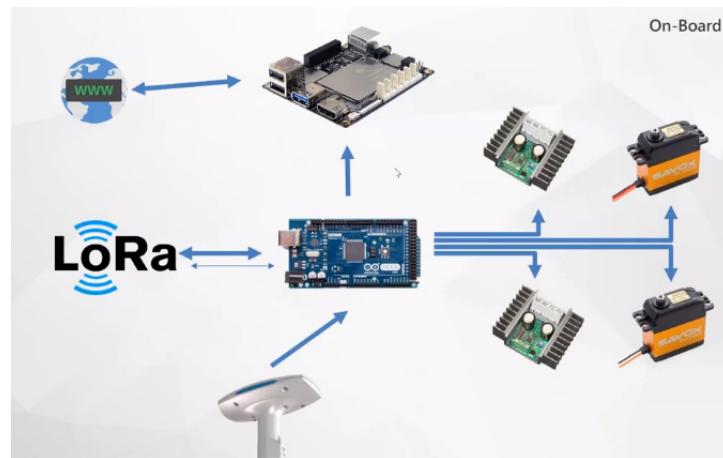


Figura 3: Subsistema On-board e seus componentes

Dentro da arquitetura On-Board é possível encontrar:

- Módulo de comunicações LoRa que funciona a 433MHz e potência de 5W para comunicação com o arduino.
- LoRa (Long Range) é uma tecnologia de rede de área ampla de baixa potência, possui grande alcance, mas baixa largura de banda. É usado para telemetria para controlo remoto com alcance até 8Km.
- Um microcontrolador Arduino Mega responsável por gerir o sistema motor.
- Um AIS (Automatic Identification System) que identifica a embarcação para outras embarcações e para a rede em terra.
- Bússola com gps integrado LT1000 que fornece informação ao arduino da posição e da proa do sistema.
- 2 Drivers para os motores controlados por modulação PWM.

- 2 Servos para controlar os lemes.
- 1 SBC (single board computer) LattePanda com Sistema Operativo Windows 10 que conecta através de um router à internet para controlo ou programação em remoto do Arduino.

Nota: Posteriormente, este SBC pode ser trocado para correr um Sistema Operativo Linux por motivos de fiabilidade, segurança e custo.

2.2.2 Software Instalado

Atualmente, o sistema tem vários software instalado, algum do qual de uso livre e outro desenvolvido pela marinha como é o caso do GCS e do código do arduino. A seguir, analisámos algum desse software.

O LattePanda funciona atualmente com o sistema operativo Windows, onde é possível usar o TeamViewer para alterar o código do arduino e trabalhar remotamente na plataforma.

O código do arduino foi desenvolvido em C++ e permite o controlo dos motores.

O protocolo NMEA é um conjunto de combinações elétricas e de dados para a comunicação entre dispositivos eletrónicos marítimos, como por exemplo eco-localizadores, receptores GPS, etc.

O OpenCPN é um software livre que cria um plotter gráfico, semelhante a uma carta náutica e que dá para visualizar posição e várias informações do sistema. Pode ser usado em andamento ou como ferramenta de planeamento. O OpenCPN é desenvolvido por uma equipa de velejadores ativos que utilizam condições do mundo real para testes e melhoramentos do software.

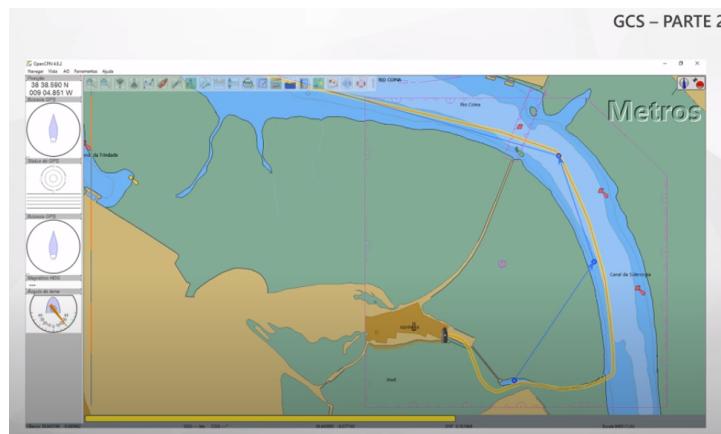


Figura 4: Software OpenCPN

O GCS (Ground Control Station) foi desenvolvido em C#. Esta linguagem de programação é desenvolvida pela Microsoft e é completamente suportada pela plataforma .NET Framework. A WinForms foi uma das frameworks de C# utilizada neste projeto, o que facilitou a criação da GUI do GCS. C# é também uma linguagem orientada a objetos, suportando conceitos como encapsulamento, herança e polimorfismo.



Figura 5: Ground Control Station

2.3 Sistemas de controlo a implementar e integrar

Neste segmento vamos analisar a toolchain do Laboratório de Sistemas e Tecnologia Subaquática (LSTS), onde vai ser integrado o veículo X-2401. Vamos concentrar os esforços nas ferramentas DUNE e Neptus. A ferramenta de software DUNE é independente tanto da arquitetura de CPU como do sistema operativo. Trata-se de um software muito compacto, não excedendo o tamanho de 16 Megabyte ([1]), sendo assim uma ferramenta de extrema modularidade. (Aspecto que pode ser diferenciador em diferentes áreas de mercado).

DUNE trata-se de um software de bordo especializado para veículos não tripulados. Contém módulos para controlo, navegação, simulação, comunicação. Assim como também pode ser responsável pelas interações entre atuadores e sensores. No contexto do projeto as ferramentas desenvolvidas DUNE, como a recolha de dados de diversos sensores e a disponibilização dos mesmos para processamento de forma organizada, permite uma maior optimização do sistema.

O DUNE foi desenvolvido na linguagem de programação orientada a objetos C++, com suporte para todos os sistemas compatíveis com POSIX, tornando portátil e eficiente. A arquitectura desenvolvida na ferramenta DUNE permite a criação de perfis por parte do utilizador. Estes perfis são ficheiros de configuração para diferentes tipos de UAVs e AUVs. A estrutura consiste em “tasks” que são executadas de forma concorrente (multi-threading) e independente (não existe partilha direta de memória). Na inicialização o DUNE instancia todas as “tasks” referidas no ficheiro de configuração, com as devidas parametrizações. De seguida executa as “tasks” em diferentes threads, que depois interagem assincronamente através de um IMC BUS (Inter-Module Communication). A comunicação inter-task segue o modelo “publish-subscribe”.

O Neptus trata-se de uma ferramenta de software utilizada para comando e controlo de UAVs. Permite o suporte a todo o ciclo de vida de uma missão, ou seja suporte para o planeamento e simulação, assim como a execução. Durante a fase de execução o Neptus permite a visualização, em tempo real, dos dados provenientes dos UAVs. Através desta ferramenta podemos fazer a análise dos dados provenientes da missão. A capacidade do Neptus de permitir, através de perfis, fazer planeamento especializado para os diferentes UAVs, torna esta ferramenta distintamente qualificada para este projecto. O acesso às diferentes cartas marítimas e a possibilidade de fazer simulações antes da execução são ambas qualidades de extrema importância também. Esta ferramenta de software foi desenvolvida em Java e é actualmente suportada pelos sistemas operativos do Microsoft Windows e Linux. A maneira como foi desenvolvida permite a criação rápida de plug-ins para satisfazer qualquer necessidade extra do utilizador. A interface de comunicações é através do IMC BUS, o que o torna interoperável com o DUNE.

O LSTS tem uma excelente bibliografia que permite uma maior adaptabilidade para eventuais novos projectos, como o caso deste projecto de integração.

3 Análise de Mercado

Como tal, o mercado USV's comprehende várias partes, como fornecedores de matéria-prima, fabricantes, fabricantes de produtos finais e organizações regulatórias na cadeia de abastecimento. O lado da demanda desse mercado é caracterizado por vários usuários finais, como organizações militares e compradores comerciais, enquanto que o lado da oferta é caracterizado por avanços na tecnologia USV, desenvolvimento de USVs e desenvolvimento de software.

3.1 Mercado Potencial

O nosso projeto foca-se na conversão de veículos tradicionais (tripulados) em USV's, pelo que, potencialmente, qualquer fabricante de embarcações tradicionais poderá ser visto como potencial mercado. Na mesma linha de pensamento, qualquer embarcação tradicional atualmente implementada pode ser vista como mercado viável para conversão. Sendo assim, considerou-se como mercado potencial qualquer setor que se encontre nestas circunstâncias, em todos os segmentos do mercado (a analisar na secção 3.2).

3.2 Segmentação de Mercado

Nesta secção tentar-se-á dividir os clientes em grupos de interesse. Existem diversos fatores sob os quais é possível fazer a segmentação de mercado. Analisámos apenas as segmentações que seriam de maior interesse para análise no âmbito do nosso projeto: aplicação e tipo de casco. Aplicação por ser o que influencia em grande parte a tecnologia incorporada no veículo. O tipo de casco será apresentado pelo facto de este também influenciar em grande parte o tipo de controlo (nomeadamente, a dois motores, no nosso caso do catamaran).

Segmentação por Aplicação:

- **Defesa:** a principal utilização em aplicações de defesa são em ISR [2] (*Intelligence, Surveillance and Reconnaissance*) sendo que a procura de USV's para desempenhar esta tarefa continua a aumentar, bem como na patrulha de portos e costa [3]. Também são utilizados para a deteção e caracterização de minas [4], operações de vigilância costeira e patrulha de portos [4]. São utilizados também para deteção de outros veículos de superfície. São também convenientes para missões de resgate, e combate a incêndios. Em âmbito militar, são utilizados em guerra anti-submarina [4]. Já há utilização de USV's em operações críticas, de forma a reduzir riscos a forças aliadas sem afetar a tripulação. Também são usados no âmbito de treino em artilharia naval.
- **Pesquisa Científica:** são altamente utilizados para a coleção de dados marítimos [2]. Estes dados marítimos incluem qualidade de água, salinidade, entre outras métricas de composição de água. São utilizados em mapeamento oceanográfico e de rios [5]. São também utilizados em testes de cascos, sensores, interoperabilidade com outros veículos, etc.
- **Ambiental:** maioritariamente monitorização ambiental, amostragens, avaliação árida, previsão e gestão de desastres, respostas de emergência [6] e medições de poluição e eventual recolha de resíduos (nomeadamente em casos de acidentes de veículos marítimos). Previsão de tempo.
- **Outras Aplicações:** exploração de minas, gás natural e petróleo. Construção de plataformas e tubagem e sua manutenção. Transportes [7]. Plataformas de abastecimento de veículos diversos. Gateway de comunicação, isto é, pode funcionar como relay à superfície e funcionar como um link de comunicação entre bases costeiras, marítimas, satélites e unidades subaquáticas [8].

Segmentação por Tipo de Casco:

- **Catamaran:** casco com segundo maior CAGR, em 2018. Tem um bom equilíbrio entre a capacidade de carga e estrutura conveniente do Kayak e a estabilidade do Trimaran.
- **Kayak:** casco preferido devido à sua estrutura conveniente, quer a nível de integração, quer a nível de potencial de carga.
- **Trimaran:** procura crescente devido à sua estabilidade e minimização de risco em águas agitadas.
- **Casco Insuflável:** adequados a aplicações militares devido à sua maior resiliência e capacidade de carga.

3.3 Dimensão de Mercado

A nível global, o mercado para USV's está valorizado a USD 1,513 milhões em 2019, a atingir USD 3,096.0 milhões em 2025, com Taxa de Crescimento Anual Composta (Compound Annual Growth Rate - CAGR) projetada em 12,5% para esse intervalo (de 2020 até 2025) [9]. Esta informação difere bastante das previsões referenciadas no ano anterior [10] que eram mais pessimistas, indicando que o crescimento de mercado tem tendência a aumentar. Com o impacto do surto de COVID-19 a CAGR caiu para 0.9% [11], sobretudo devido ao efeito que este surto, e as medidas tomadas quanto a este, tiveram na economia a nível global.

Geograficamente, a América do Norte possui a maior percentagem do mercado de USV's [12] e espera-se também que vá apresentar o maior lucro gerado até 2025. Tal deve-se ao aumento da violência em determinados países conjugado, com a atual evolução tecnológica. Por outro lado, existe um aumento do CAGR na Europa (onde existe o maior aumento de CAGR [12]) devido a um crescimento da procura de USV's por parte da marinha para estudos oceanográficos e atividades ISR relacionadas com o ambiente.

3.4 Dinâmica de Mercado

Os veículos autónomos estão a ganhar popularidade pelo que se estima uma crescente procura por aplicações relacionadas com a construção e integração destes equipamentos em tarefas quotidianas, bem como em atividades de tecnologia e inovação. Tem-se verificado nos últimos anos um maior interesse na patrulha marítima com veículos não tripulados, por permitirem grandes períodos de atividade ininterrupta e identificação em situações de pouca visibilidade (como por exemplo de noite com sistemas infra-red). Paralelamente à defesa da costa, a preocupação com a preservação e estudo dos ecossistemas subaquáticos e mapeamento dos fundos oceânicos têm vindo a impulsionar o desenvolvimento dos veículos em questão, por serem mais robustos e flexíveis. No entanto, o maior entrave à integração de sistemas autónomos em ambiente aquático é o grande investimento necessário nas etapas iniciais do desenvolvimento e as ainda existentes questões com manutenção. Este investimento é parcialmente ou completamente contrariado pelo facto de não ser necessário investir em tripulação, bem como o potencial aumento de eficiência em gastos de combustíveis (assumindo que o controlador de execução é mais eficiente que um operador humano).

Verifica-se que o mercado dos veículos não tripulados ainda está relativamente fragmentado [12], sendo que ainda há bastante competição. Até ao momento, as empresas mais competitivas são:

- L3 ASV
- Kongsberg
- Ocean Alpha
- Sea Machines
- MARTAC
- General Dynamics
- Atlas Elektronik
- Sea Robotics Inc.
- Liquid Robotics
- Rafael Advanced Defense Systems

4 Fornecedores / Competição

Por fim, é relevante analisar a potencial competição e fornecedores de equipamentos da mesma gama. Com o crescimento do mercado têm surgido cada vez mais companhias a inovar a tecnologia inerente aos USVs e a análise das respetivas características permite um maior leque de opções para o desenvolvimento do nosso produto.

De seguida apresentamos algumas companhias que atuam na área e respetivos produtos de interesse:

4.1 L3 ASV

Com sede nos EUA e no Reino Unido a *L3 ASV* [13] é uma fornecedora de Sistemas Marítimos Não Tripulados e Autónomos. Como produto da área de sistemas de controlo de USVs eles apresentam o *ASView Control System* que apresenta:

- Um sistema implementado em mais de 80 veículos não tripulados;
- Controlo direto remoto para lançamento e recuperação via ASView-Helm;
- Controlo e monitorização de uma variedade de motores e propulsores. Controlo assistido de direção e velocidade;
- ASView-Core: Sistema de controlo na embarcação. Ligado a sensores e atuadores por meio de uma rede, NMEA, CAN ou Ethernet;
- ASView-Bridge: uma interface de operador para controlo da embarcação. Fornece cartas náuticas, ferramentas de planeamento de missão, dados do sensor, vídeo e radar do veículo. Corre num sistema operativo Linux;
- Pode ser controlado externamente a partir de interfaces terceiras, ou sistemas operativos robóticos open-source como MOOS, ROS, SeeByte Neptune.

4.2 Kongsberg

Fundada em 1814 na Noruega, a *Kongsberg Group* [14] é uma das empresas líder tecnológicas no país. Em 2017 a divisão *Kongsberg Maritime* em conjunto com a *Norwegian Defence Research Establishment* desenvolveu o K-MATE, um sistema de controlo de veículos autónomos de superfície a ser integrado em USVs, com características prometedoras:

- Capacidade de instalação em grande parte dos USVs via interface elétrica;
- Seguimento de waypoints adaptativo enquanto recebe dados dos sensores para analisar terrenos e evitar colisões;
- Cumprimento de missões planeadas;
- Supervisão de operações;
- Capacidade de controlo direto por um operador em tarefas mais complexas.

4.3 OceanAlpha

A *OceanAlpha* [15], empresa Chinesa fundada em 2010, garante ser a "fornecedor líder mundial" de USVs contando com mais de 100 engenheiros e 92 patentes relacionadas aos mesmos. No início do ano 2020 foi construído o centro de pesquisa, desenvolvimento e teste de USVs da empresa constituído por piscinas para testes topográficos e por um "porto" capaz de acolher 108 testes do produto. A *OceanAlpha* é responsável pelo desenvolvimento de veículos para alto mar, para monitorização do ambiente, veículos de segurança e resgate, entre outros. Um destes, e o que mais se aproxima do projeto em questão, é o ME120 cujas propriedades são listadas a seguir:

- Capacidade de pré-programação de missões;
- Controlo direto do utilizador a partir de um controlador remoto;
- Deteção de obstáculos;
- Visualização do ambiente em tempo real durante o controlo manual.

4.4 Sea Machines

A *Sea Machines* [16] é uma desenvolvedora de sistemas de controlo marítimo; sistemas de comando autónomo, percepção avançada e controlo remoto do leme e assistência à navegação para embarcações de superfície com sedes nos EUA, Alemanha e Dinamarca. Desenvolveram vários produtos da gama SM, mas o que representa a maior inovação na área dos USVs é o *SM300* que apresenta as seguintes características:

- Ideais para embarcações de combate a incêndios, de pesquisa comercial, de resposta a derramamentos, embarcações de segurança / patrulha / busca e resgate e de aquicultura;
- Hub de controlo inteligente a bordo faz interface e processa dados de radar, câmeras, GPS, Sistema de Identificação Automática (AIS), Cartas Náuticas Eletrónicas (ENCs);
- Detecção e prevenção de obstáculos;
- Interface de apontar e clicar que permite aos usuários planejar, rastrear e registar missões simplesmente selecionando waypoints e inserindo parâmetros da embarcação.

4.5 Liquid Robotics

A *Liquid Robotics* [17], que pertence totalmente à *The Boeing Company*, conseguiu desenvolver um USV alimentado a energia solar e energia das ondas sem quaisquer emissões. O *Wave Glider* foi criado para alcançar locais de difícil acesso ou com elevadas implicações financeiras. Algumas outras características importantes deste USV são:

- Recolha e envio de informação em tempo real durante até um ano sem combustível;
- Capacidade de implementar novos sensores e novas funcionalidades de software;
- Capacidade de interagir com outros sistemas;
- Capacidade de navegação em atmosféricas adversas.

4.6 MARTAC

Maritime Tactical Systems ou *MARTAC* [18] é um fornecedor inovador de veículos de superfície não tripulados (USV) chamados *MANTAS* para missões e mercados militares, de segurança, comerciais e científicos desde 2012. Tem sede nos EUA. O *MANTAS* apresenta:

- Propulsão elétrica ou diesel / elétrica;
- Consola de comando e controlo operacional TASKER (Sistema próprio da MARTAC);
- Interface com mapa da missão e vídeo em tempo real;
- Modos de controlo do operador totalmente autónomo, semi-autónomo e manual;
- Sensores e comunicações agnósticas;
- Mecanismo de lançamento e recuperação.

4.7 Análise geral dos fornecedores

Em suma, é evidente que já existem bastantes fornecedores tanto de sistemas de controlo de USVs como de USVs propriamente ditos. Com o crescer deste mercado, ainda assim, é expectável que surjam ainda mais contribuições para a inovação na área dos veículos de superfície não tripulados.

Das companhias analisadas podemos retirar algumas características interessantes ao projeto no que toca à parte de interface, como a introdução de cartas náuticas e de um sistema de seguimento de waypoints. No entanto, se o projeto se estendesse para além da interface do sistema de controlo, existiriam mais especificações de interesse como um mecanismo manual de lançamento e recuperação do veículo, assim como a deteção e prevenção de colisão com obstáculos. Outra capacidade interessante seria um design promovendo a extensibilidade, ou seja, a possibilidade de adição de novos módulos sem a alteração da arquitetura do sistema.

5 Conclusões

Com este documento, conseguiu-se apresentar o sistema e respetivos subsistemas de forma concisa, com uma análise de mercado superficial, bem como uma análise de competição. Este estudo teve algumas limitações consideráveis, nomeadamente a falta de informação, ou limitação económica ao seu acesso.

Este estudo permitiu concluir que a conversão de veículos tradicionais em USVs é ainda um campo pouco explorado, no entanto com boa perspetiva de sucesso, visto o aumento constante de procura de USVs, sendo previsível que a conversão de embarcações tradicionais será, igualmente, viável.

Lista de Figuras

1	Protótipo X-2601 e os seus módulos de hardware	4
2	Sistema e subsistemas da arquitetura de controlo	5
3	Subsistema On-board e seus componentes	5
4	Software OpenCPN	6
5	Ground Control Station	7

- [17] ‘Reimagine ocean monitoring and operations. unmanned robots powered by nature,’ <https://www.liquid-robotics.com/wave-glider/overview/>, accessed: 2020-10-17.
- [18] ‘Mantas usv features,’ <https://martacsystems.com/mantas-vessels/>, accessed: 2020-10-17.

Anexo C Lista de Requisitos

Faculdade de Engenharia da Universidade do Porto



FEUP

**Sistema de Planeamento e Controlo de Execução para Veículo de
Superfície Não Tripulado - Lista de Requisitos**

Equipa E

Orientador João Sousa

Líder da Equipa Carlos Pinto

Lista de Requisitos realizado no âmbito da unidade curricular
Sistemas de Engenharia - Automação e Instrumentação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

18 de janeiro de 2021

Registo de Versões

Versão	Data	Descrição
0.1	20/10/2020	Criação do documento base segundo o template definido
0.2	20/10/2020	Outline inicial (estrutura)
0.3	23/10/2020	Objetivo do documento e estrutura geral da apresentação de requisitos
0.4	24/10/2020	Junção das secções de requisitos
0.5	24/10/2020	Adição de uma secção para legislação
0.6	24/10/2020	Revisão em equipa dos requisitos
1.1	2/11/2020	Alteração dos requisitos após consulta com cliente.
1.2	22/11/2020	LaTeX: utilização de macros para definir os códigos dos requisitos, de forma a tornar a sua alteração mais fácil.
1.3	23/11/2020	Alteração dos requisitos após melhor avaliação e conhecimento do projeto e feedback da apresentação intermédia.

Índice

1	Introdução	3
2	Objetivo da Lista de Requisitos	3
3	Visão Geral do Sistema	3
4	Estruturação das Tabelas de Requisitos	4
5	Necessidades	4
6	Requisitos	5
6.1	Requisitos Funcionais	5
6.2	Requisitos de Performance	5
6.3	Requisitos de Design	6
6.4	Requisitos Alocados	6
6.5	Requisitos Derivados	7
7	Legislações	9
7.1	AIS	11
7.2	Certificados de conformidade de alguns dos equipamentos eletrónicos	11
7.3	Algumas Fontes Relevantes	11
	Lista de Tabelas	12
	Lista de Figuras	13
	Referências	14

1 Introdução

A integração do sistema de controlo de um veículo (no caso USV) numa *toolchain* que consta com vários outros veículos necessita de uma análise de requisitos de modo a que todas as especificações do cliente e dependências da *toolchain* sejam cumpridas e consigam operar com o melhor nível de performance possível.

O sistema pode ser dividido em uma parte de baixo nível que envolve o controlo direto dos motores e lemes, outra de alto nível que envolve o planeamento de rotas, gestão de recursos, análise e tratamento de dados e por fim a comunicação entre estas partes chave do sistema. Sem esquecer que o veículo operará num ambiente que requer o cumprimento de normas e legislações, assim como uma interação contínua com outros veículos, que operam no mesmo meio de modo a promover a segurança de todas as embarcações.

De um outro ponto de vista, esta análise representa também uma organização inicial do caderno de encargos do projeto, promovendo uma análise mais aprofundada e detalhada das tarefas que a equipa realizará de modo a atingir o produto final.

2 Objetivo da Lista de Requisitos

Este documento tem como objetivo a identificação e classificação de todos os requisitos fornecidos pelo cliente (necessidades), gerando requisitos com base nessas necessidades. Os requisitos serão estruturados e apresentados de forma concisa, em tabelas, como consta na secção 4.

O sistema em questão é brevemente descrito na secção 3.

Em cada uma das secções relativas a requisitos segue uma breve descrição dos conteúdos da mesma, e a respetiva tabela de requisitos. Na secção 5 serão especificados os requisitos fornecidos pelo cliente, na secção 6 apresentam-se os requisitos com base nas necessidades do cliente, subdivididos em requisitos: funcionais (secção 6.1), de performance (secção 6.2), de design (secção 6.3), alocados (secção 6.4) e derivados (secção 7.2).

Alguns artigos de legislações e normas que considerámos mais relevantes para o âmbito do projeto estão enunciados na secção 7.

3 Visão Geral do Sistema

O sistema que se pretende conceber está representado na figura 1. O sistema atual, consiste apenas no catamarã - X-2601 - em si, com um microcontrolador Arduino implementado, com rotinas de controlo simples, e o LattePanda, que apenas é utilizado para programar o Arduino remotamente.

Todas as demais funcionalidades fazem parte do projeto: a implementação do DUNE na placa LattePanda, com os conceitos de supervisor e de manobras, bem como a implementação de uma *Ground Control Station* (GCS), que será essencialmente um computador com a plataforma Neptus, responsável por enviar missões e monitorizar o USV, sendo que este último envia dados relativos à embarcação e, eventualmente, relativos à missão. O software do Arduino também será alterado ao longo do desenvolvimento do projeto, e, potencialmente, irão ser acrescentados sensores e/ou atuadores.

Assim, o sistema pode ser dividido numa parte de baixo nível que envolve o controlo direto dos motores e lemes, outra de alto nível que envolve o planeamento de rotas, gestão de recursos, análise e tratamento de dados e por fim a comunicação entre estas partes chave do sistema. Sem esquecer que o veículo operará num ambiente que requer o cumprimento de normas e legislações, assim como uma interação contínua com outros veículos, que operam no mesmo meio de modo a promover a segurança de todas as embarcações.

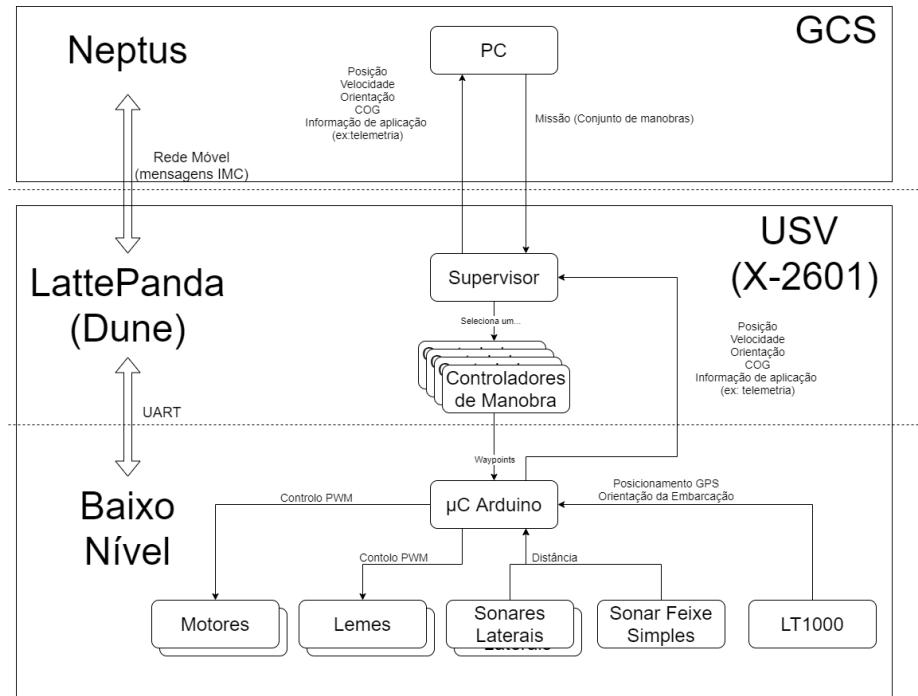


Figura 1: Esquemático geral do sistema que se pretende desenvolver.

4 Estruturação das Tabelas de Requisitos

As tabelas de requisitos serão estruturadas como consta na Tabela 1, em que os elementos da tabela estão ordenados, de uma forma não rigorosa, por prioridade (maior prioridade primeiro).

Tabela 1: Exemplo de tabela de requisitos.

Código de Requisito	Requisito Associado	Descrição	Classificação
Um código curto usado meramente para referenciar este requisito noutras tabelas, quando necessário	Código do requisito que levou à identificação do requisito em questão, maioria das vezes este requisito será uma necessidade do cliente. As necessidades do cliente não têm nenhum requisito associado.	Descrição sucinta do requisito em questão	Classificação do requisito em questão, sendo que pode ser um requisito "Essencial" ou "Opcional"

5 Necessidades

Entendem-se como necessidades os requisitos fornecidos diretamente pelo cliente. São estes que servem de base para os restantes requisitos, e formalizam o âmbito geral no qual se desenvolve o projeto.

Tabela 2: Requisitos do cliente.

Código de Requisito	Descrição	Classificação
N1	O sistema deve possuir interoperabilidade entre os demais veículos da Marinha	Essencial
N2	O sistema deverá dirigir-se autonomamente para os pontos de destino	Essencial
N3	O sistema deve ser eficiente, a nível energético	Essencial
N4	O sistema deverá conseguir comunicar, mesmo quando longe da costa	Essencial

6 Requisitos

Nesta secção serão apresentados todos os requisitos que derivam das necessidades do cliente, quer de forma direta, quer de forma indireta.

6.1 Requisitos Funcionais

Requisitos funcionais referem-se às tarefas que o sistema deve desempenhar, de forma genérica e de alto nível. Não inclui implementações específicas, nem definições detalhadas de como desempenhar a tarefa, nem de como desenvolver o sistema.

Tabela 3: Requisitos funcionais.

Código de Requisito	Requisito Associado	Descrição	Classificação
RF1	N1	A GCS deverá ser capaz de comunicar a missão ao USV.	Essencial
RF2	N2	A GCS deve conseguir planejar missões simples (escolha de pontos de passagem).	Essencial
RF3	N2	O USV deve conseguir executar missões simples (escolha de pontos de passagem).	Essencial
RF4	N2	Dado um ponto de destino, o USV deve ser capaz de se deslocar para esse mesmo destino, com uma tolerância de 250 centímetros.	Essencial
RF5	N1	A GCS deverá ser capaz de monitorizar a missão, com dados atualizados no máximo a cada 15 segundos.	Essencial
RF6	N1, N2	A GCS deve conseguir planejar missões complexas (manobras avançadas, de trajetos curvos definidos, etc.).	Opcional
RF7	N1, N2	O USV deve conseguir executar missões complexas (manobras avançadas, de trajetos curvos definidos, etc.).	Opcional
RF8	N2	Implementar uma rotina de segurança (retornar à base) no caso do USV perder comunicação por mais de 1 minuto com a plataforma de planeamento e execução.	Opcional
RF9	N2	No caso da bateria/depósito se encontrar com uma capacidade inferior a 5%, o USV deverá retornar à base.	Opcional
RF10	N2	O veículo deverá escolher um caminho seguro, considerando a profundidade mínima necessária	Opcional

6.2 Requisitos de Performance

O desempenho com que dadas tarefas têm de ser executadas. Estes requisitos terão, de forma geral, dados quantitativos concretos que permitam a validação (ou invalidação) do requisito de forma mensurável. Devido ao contexto e grau de incerteza do nosso projeto, deixaremos que os requisitos de performance descrevam o desempenho do sistema de forma mais qualitativa, na perspetiva de futuras iterações deste documento terão dados quantitativos mais concretos.

Tabela 4: Requisitos de performance.

Código de Requisito	Requisito Associado	Descrição	Classificação
RP1	N2	O sistema tem de ser seguro e validado.	Essencial
RP2	N1	As comunicações do USV, nomeadamente com a GCS, devem ser realizadas em tempo útil, com um atraso máximo de 15 segundos.	Essencial
RP3	N2, N4	A comunicação entre o USV e a GCS deve ser possível por mais do que um meio.	Essencial
RP4	N3	A utilização dos lemes a altas velocidades deve ser reservada a manobras rápidas, que envolvam um erro de direção superior a 90º.	Opcional
RP5	N2, N3	No caso de manobras: garantir a proximidade adequada entre pontos de destino adjacentes (máximo 40 metros entre pontos), de forma a promover maior eficiência energética e uma navegação mais suave.	Opcional
RP6	N3	No caso do carregador de baterias apresentar uma capacidade inferior a 20%, o USV deverá reduzir a sua velocidade de referência para metade.	Opcional

6.3 Requisitos de Design

A abordagem específica com a qual se vai obedecer às necessidades. Incluem detalhes mas específicos de hardware, algoritmos de software e, em geral, soluções mais definidas a implementar no sistema.

Tabela 5: Requisitos de design.

Código de Requisito	Requisito Associado	Descrição	Classificação
RD1	N1	Desenvolver software no USV recorrendo à toolchain LSTS, integrando-o numa plataforma comum a outros veículos da marinha.	Essencial
RD2	N2	Deve ser criado perfil com parametrização específica ao USV / Configuração do perfil específico do USV.	Essencial
RD3	N1	Utilizar o standard IEC AIS (IEC 62320-1) para identificação do próprio USV aos restantes, e vice-versa.	Opcional
RD4	N3	No seguimento de um caminho, o USV não deve executar curva de cão. Será considerado um desvio significativo uma distância de 20 metros da trajetória executada em relação à linha de trajetória direta desejada	Opcional
RD5	N2	Cada funcionalidade desenvolvida em software deverá ser desenvolvida de forma a que seja independente e modular das demais funcionalidades.	Opcional

6.4 Requisitos Alocados

Sub-divisão de requisitos de mais alto nível, referidos nas secções anteriores.

Tabela 6: Requisitos alocados.

Código de Requisito	Requisito Associado	Descrição	Classificação
RA1	RF4	Desenvolver interface de sinal entre sensores e atuadores e o sistema de planeamento e execução.	Essencial
RA2	RD1	Todo o software desenvolvido no LattePanda deverá ser realizado no DUNE.	Essencial
RA3	RD1	Todo o software desenvolvido na GCS deverá ser realizado no Neptus.	Essencial
RA4	RF3, RF7	Desenvolver cálculos de caminho e de planeamento no Neptus/DUNE.	Essencial
RA5	RF5	Garantir o acesso a mapas e/ou cartas de navegação.	Essencial
RA6	RP3	O USV deverá ter acesso a comunicação por satélite.	Essencial.
RA7	RP3	O USV deverá ter acesso a comunicação por LoRa.	Essencial.
RA8	RP3, RP2	O USV deverá comunicar com a GCS por LoRa, e, caso tal não seja possível, recorrer a comunicação por satélite. O tempo definido para uma comunicação não sucedida deverá ser de 7 segundos, após o qual, deverá ser considerado o défice na comunicação, sendo realizada a troca para comunicação por satélite.	Essencial.
RA9	RF4	O LattePanda deve controlar os motores e lemes de forma a que o catamarã siga os waypoints.	Essencial
RA10	RF4	O LattePanda deve receber dados de velocidade, COG e orientação do LT1000, com uma cadência mínima de 0.5 leituras/segundo.	Essencial
RA11	RF10	Deve ser utilizado um caminho seguro, o que engloba navegar onde a profundidade excede o calado.	Opcional
RA12	RF6	Assegurar condições propícias de missão recorrendo às cartas de navegação segundo o standard ENC S-57.	Opcional.

6.5 Requisitos Derivados

Requisitos que advêm de outros requisitos, de mais alto nível, referidos nas secções anteriores.

Tabela 7: Requisitos derivados.

Código de Requisito	Requisito Associado	Descrição	Classificação
Rd2	RA1	Envio dos sinais de controlo dos atuadores do LattePanda para o Arduino.	Essencial
Rd4	RA12, RF5	Análise e tratamento de dados proveniente das Cartas de Navegação, pelo Neptus.	Essencial
Rd5	RD2	Assegurar funcionalidade dos periféricos através da configuração das tasks do toolchain do LSTS.	Essencial
Rd8	RD2	Configurar o perfil do USV, no DUNE, com os parâmetros da embarcação.	Essencial
Rd1	RF7	Interpretação das manobras, provenientes do Neptus, pelo LattePanda (DUNE), gerindo os controladores de manobra a utilizar.	Opcional
Rd6	RD4	Quando o veículo está em movimento (velocidade superior a 1 km/h), deverá utilizar o rumo (COG) como referência para os controladores dos motores.	Opcional
Rd7	RD4	Quando o veículo inicia movimento (velocidade inferior a 1 km/h), deverá utilizar a orientação da proa como referência para os controladores dos motores.	Opcional

7 Legislações

Para além dos requisitos, é necessário seguir algumas normas e legislações para uma implementação e utilização corretas. Contudo, como estas embarcações se tratam de uma tecnologia recente, que ainda está em desenvolvimento, não existe grande legislação nacional e até internacional específica para controlar o seu uso, embora existam esforços neste sentido. Contudo, é importante realçar a necessidade de criação de medidas normativas no âmbito global em relação a tais meios, assim como a necessidade do desenvolvimento de políticas públicas de âmbito marítimo, tanto nacionais como internacionais, acerca desta temática. Assim sendo, no que diz respeito à legislação portuguesa, de acordo com o decreto-Lei n.º 265/72, classificou-se a embarcação como sendo de investigação, ficando sujeita ao regime legal aplicável às embarcações auxiliares. Na falta de legislação nacional, foi utilizada a legislação das Nações Unidas (CNUDM).

Decreto-Lei n.º 265/72 ARTIGO 19.º

1 — As embarcações da marinha nacional, incluindo as do Estado não pertencentes à Armada, a forças e serviços de segurança interna e a outros órgãos do Estado com atribuições de fiscalização marítima, em conformidade com as actividades a que se destinam, classificam-se como:

- a) De comércio;
- b) De pesca;
- c) De recreio;
- d) Rebocadores;
- e) De investigação;
- f) Auxiliares;
- g) Outras do Estado.

ARTIGO 23.º-A

1 — As embarcações de investigação são as que dotadas de meios de propulsão mecânica se destinam, consoante a sua aptidão técnica, à investigação científica, oceânica ou costeira.

2 — As embarcações referidas no número anterior ficam sujeitas ao regime legal aplicável às embarcações auxiliares.

ARTIGO 19.º

(Pedidos de autorização)

2 - Os pedidos de autorização para a realização das actividades referidas no artigo anterior só poderão ser considerados: b) Desde que as actividades a realizar prossigam fins pacíficos, utilizem métodos científicos e técnicos e meios adequados, não interfiram com outras utilizações legítimas do mar, com respeito das leis e regulamentos em vigor, em particular os que visam proteger e preservar o ambiente aquático, os seus recursos e o património arqueológico subaquático;

Qual é a entidade que em Portugal tem competência para receber pedidos relativos a projetos de investigação científica marinha e dar o respetivo consentimento?

No caso de se subsumirem no quadro do Decreto-Lei n.º 38/2015 (Artigos 47.º e 57.º), os projetos de investigação científica marinha apresentados por entidades nacionais devem ser submetidos a Direção Geral de Recursos Naturais, Segurança e Serviços Marítimos que tem competência para emitir o título de utilização privativa do espaço marítimo nacional que autoriza a realização do projeto. Os projetos que incidam sobre as zonas marítimas adjacentes às Regiões Autónomas, ate as 200 milhas náuticas, devem ser submetidos aos organismos competentes das Regiões Autónomas: no caso da Região Autónoma dos Açores a Direção Regional dos Assuntos do Mar; no caso da Região Autónoma da Madeira a Secretaria Regional do Ambiente e Recursos Naturais — entidades com competência para a emissão do título de utilização privativa que autoriza a implementação do projeto. Os projetos de investigação científica marinha apresentados por entidades estrangeiras

devem ser submetidos ao Ministério dos Negócios Estrangeiros que reencaminha os pedidos para as autoridades competentes. (Decreto-Lei n.º 38/2015 e Decreto-Lei n.º 52/85)

Pode o Estado costeiro impor condições a entrada de navios não tripulados e UMVs nas suas águas interiores ou nos portos?

Sim. O Estado costeiro tem soberania sobre as suas águas interiores e pode impor as condições que desejar para a entrada de navios não tripulados a UMVs nas suas águas interiores ou portos. Assim, pode especificar os portos que estão abertos a navios não tripulados a UMVs, impor determinadas condições para a sua entrada, incluindo relativamente a questões de segurança, a pode recusar a sua entrada, mesmo nos casos em que o navio não tripulado ou o UMV esteja em situação de perigo, desde que tal recusa não seja discriminatória nem constitua abuso de direito.

(Artigo 2.º, CNUDM)

Quando os UMVs são usados em projectos de investigação científica marinha, é obrigatória a sua identificação no projeto?

Sim. Quando um Estado ou uma organização internacional submete um projeto a um terceiro Estado para investigação científica marinha em área sob sua soberania ou jurisdição, tem de prestar ao Estado costeiro informação detalhada, nomeadamente sobre os métodos e os meios a utilizar, incluindo o nome, a tonelagem, o tipo e a categoria das embarcações e uma descrição do equipamento científico a ser usado na investigação. Por isso, sendo classificados como equipamentos, os UMVs devem ser devidamente identificados no documento do projeto.

Quais são as obrigações que a legislação nacional impõe aos UMVs usados em projetos de investigação científica marinha?

A entidade responsável pelo projeto de investigação está sujeita às seguintes obrigações no que aos UMVs diz respeito:

1. Obrigação de informação — a memória descritiva e justificativa do projeto deve conter informação sobre o equipamento usado, inclusive relativamente a UMVs;
2. Obrigação de manter o bom estado do ambiente marinho — esta é uma obrigação de todo o projeto, mas que se estende a todos os instrumentos e equipamentos usados. Por isso, os UMVs usados no projeto de investigação devem também cumprir esta obrigação;
3. Obrigação de remoção — qualquer equipamento usado, que, por qualquer motivo, se tenha afundado, deve ser removido do ambiente marinho após o término do projeto.

(Decreto-Lei n.º 38/2015 e Decreto-Lei n.º 52/85)

Quais são as obrigações dos titulares dos navios não tripulados e dos UMVs no caso em que o mesmo se afunde?

Os navios não tripulados devem ser obrigatoriamente removidos do ambiente marinho no caso de se terem afundado. Essa obrigação pertence ao proprietário, mas em certas condições de ameaça ao ambiente marinho, a remoção poderá ser feita pelas autoridades nacionais a expensas do proprietário. Os UMVs que sejam classificados como equipamento usados para projetos de investigação científica marinha devem ser obrigatoriamente removidos do ambiente marinho aquando da finalização do projeto, mesmo quando não se tenham afundado.

(Decreto-Lei n.º 64/2005 e Decreto-Lei 52/85)

As fotografias, os vídeos e os registos sonoros captados por UMVs podem ser aceites como prova perante os tribunais internacionais previstos na CNUDM?

Sim. As normas processuais que regulam os tribunais internacionais, nomeadamente o Tribunal Internacional de Justiça e o Tribunal Internacional do Direito do Mar, permitem que as partes possam apresentar, no âmbito da prova documental, fotografias, vídeos e registos sonoros. Não existem razões para fundamentar a recusa de tais provas, pelo facto de as mesmas terem sido recolhidas por UMVs.

As fotografias, os vídeos e os registos sonoros captados por UMVs podem ser aceites como provas perante os tribunais nacionais?

Em princípio, a prova que seja constituída por fotografias, vídeos e registos sonoros pode ser submetida aos tribunais nacionais na categoria de prova documental. A sua admissibilidade em processos crime é mais restrita do que nos processos cíveis porque as condições legais impostas para que as mesmas sejam consideradas admissíveis são mais exigentes e limitadas. No entanto, tal exigência não se relaciona com o facto de as mesmas terem sido recolhidas por UMVs.

7.1 AIS

O AIS é regulamentado em todo o mundo por uma série de organizações nacionais e internacionais. Este regulamento rígido garante que todos os produtos AIS atendam aos padrões mínimos de desempenho, funcionalidade e interoperabilidade. Em 2007, foi aprovada uma nova norma mundial para estações de base AIS, a norma IEC 62320-1, sendo a mais utilizada atualmente.

O AIS colocado no X-2601 é um AIS de classe B, seguindo segundo o regulamento as normas nacionais.

7.2 Certificados de conformidade de alguns dos equipamentos eletrónicos

Lattepanda:

<http://www.lattepanda.com/topic-f13t662.html>

Arduino:

https://content.arduino.cc/assets/CE_MEGA2560.pdf

https://content.arduino.cc/assets/FCC_MEGA2560.pdf

7.3 Algumas Fontes Relevantes

Direção Geral de recursos naturais, segurança e serviços marítimos [1].

Compilação de Legislação para as Actividades Marítimas [2].

Regulamento Geral das Capitanias- Marinha Portuguesa [3].

COLREGS - International Regulations for Preventing Collisions at Sea [4].

Lista de Tabelas

1	Exemplo de tabela de requisitos.	4
2	Requisitos do cliente.	4
3	Requisitos funcionais.	5
4	Requisitos de performance.	6
5	Requisitos de design.	6
6	Requisitos alocados.	6
7	Requisitos derivados.	7

Lista de Figuras

1	Esquemático geral do sistema que se pretende desenvolver.	4
---	---	---

Referências

- [1] “Legislação marítima - dgrm,” <https://www.dgrm.mm.gov.pt/legislacao-maritima>, accessed: 2020-10-26.
- [2] “Compilação de legislação para as actividades marítimas,” https://www.marinha.pt/conteudos_externos/lexmar/PGPAT%20100/PGPAT_100_indice_internet.htm, accessed: 2020-10-26.
- [3] “Regulamento geral das capitania- marinha portuguesa,” [https://www.marinha.pt/Conteudos_Externos/lexmar/PGPAT%20100/PGPAT%201000%20-%20Cap%C3%ADtulos/Cap%C3%ADtulo%20I%20Identifica%C3%A7%C3%A3o%20e%20opera%C3%A7%C3%A3o%20emb/RGC/Decreto-Lei%20265-72%20\(31%20JUL\)%20CONS.htm](https://www.marinha.pt/Conteudos_Externos/lexmar/PGPAT%20100/PGPAT%201000%20-%20Cap%C3%ADtulos/Cap%C3%ADtulo%20I%20Identifica%C3%A7%C3%A3o%20e%20opera%C3%A7%C3%A3o%20emb/RGC/Decreto-Lei%20265-72%20(31%20JUL)%20CONS.htm), accessed: 2020-10-26.
- [4] “Colregs - international regulations for preventing collisions at sea,” [http://www.mar.ist.utl.pt/mventura/Projecto-Navios-I/IMO-Conventions%20\(copies\)/COLREG-1972.pdf](http://www.mar.ist.utl.pt/mventura/Projecto-Navios-I/IMO-Conventions%20(copies)/COLREG-1972.pdf), accessed: 2020-10-26.

Anexo D Conceito de Sistema

Faculdade de Engenharia da Universidade do Porto



**Sistema de Planeamento e Controlo de Execução para Veículo de
Superfície Não Tripulado - Conceito de Sistema**

Equipa E

Orientador João Sousa

Líder da Equipa Carlos Pinto

Conceito de Sistema realizado no âmbito da unidade curricular
Sistemas de Engenharia - Automação e Instrumentação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores

24 de novembro de 2020

Registo de Versões

Versão	Data	Descrição
0.1	31/10/2020	Criação do documento base segundo o template definido
0.2	1/11/2020	Desenvolvimento das várias secções pela equipa, de forma independente
0.3	1/11/2020	Revisão em Equipa
0.4	2/11/2020	Revisão pelo Orientador
0.5	2/11/2020	Revisão em Equipa: Alterações propostas pelo Orientador
0.6	2/11/2020	Reorganização das secções do documento
0.7	2/11/2020	Revisão final pelo coordenador de documentação

Índice

1 Objetivo e Organização do Documento	3
2 Desafio do Projeto	4
3 Conceito de Sistema	5
A Apêndice 1: Neptus	6
A.1 Papel do Neptus no projeto	6
A.1.1 Planeamento de missões futuras	6
A.1.2 Visualização de dados em tempo real	6
A.1.3 Revisão de informação de missões passadas	6
A.1.4 Conceitos a destacar	7
A.1.5 Parâmetros dos veículos/Configurações	7
A.1.6 Interfaces	7
B Apêndice 2: Dune	8
B.1 Introdução	8
B.1.1 Arquitetura	8
B.1.2 Perfis e Configuração	8
B.2 Tarefas do DUNE	9
Lista de Figuras	10

1 Objetivo e Organização do Documento

O presente documento tem como objetivo descrever o conceito de sistema a desenvolver, de uma forma genérica e direta. Permitirá estruturar o projeto em componentes e ilustrar as comunicações entre estes componentes, bem como o seu papel para o sistema no seu todo.

Na secção 2 é apresentado o problema a resolver e os objetivos do projeto; a secção 3 aborda as soluções definidas para desenvolver o projeto no âmbito do desafio apresentado.

2 Desafio do Projeto

O projeto tem como objetivo a adaptação de um veículo de superfície não tripulado (USV), nomeadamente um catamarã, que será brevemente apresentado. O conceito de sistema do USV a adaptar está representado na figura 1.

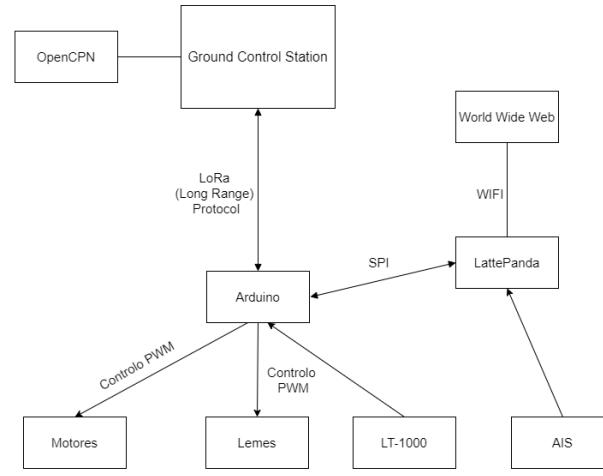


Figura 1: Estado inicial do sistema

O Centro de processamento On-Board consiste num Arduino, que processa a informação proveniente dos sensores integrados no módulo LT-1000 e encarrega-se de fazer o controlo PWM dos Lemes e Motores.

O Arduino encarrega-se da comunicação com a GCS (Ground Control Station), através do protocolo de comunicação LoRa, para as tarefas de planeamento e rastreamento. A posição em tempo real do USV é calculada pelo módulo AIS e baseia-se em GPS . Do Ground Control Station é possível aceder a uma carta marítima, por intermédio do software Open CPN, e enviar pontos para o USV se dirigir.

O LattePanda permite interagir, à distância, com o veículo e as suas configurações de código (através de TeamViewer).

O projeto baseia-se na adaptação do sistema, de forma a garantir que o USV:

1. interate com outros veículos da marinha;
2. tenha um grau de autonomia maior, no que toca a planeamento e decisão de manobras;
3. comunique perto e longe da costa;
4. opere de forma mais eficiente em termos energéticos;

Para tal, foi decidido analisar o sistema já implementado e reorganizá-lo de forma a cumprir os objetivos supra mencionados.

3 Conceito de Sistema

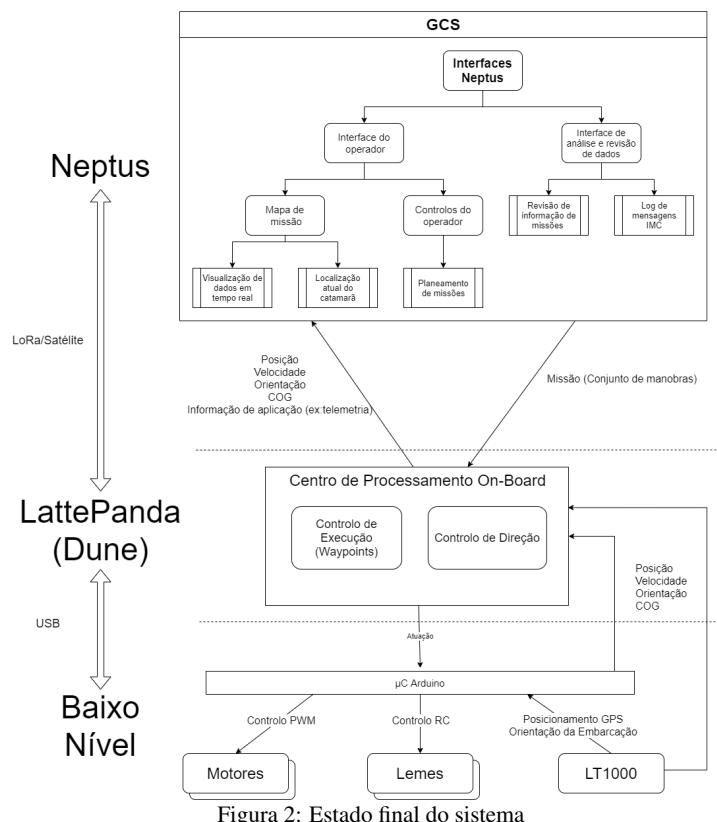
No final deste projeto é esperado obter uma nova estrutura, como demonstrado na figura 2.

O software Neptus (consultar Anexo A) terá o papel de comandar, controlar e monitorizar o sistema USV. Faz o planeamento de alto nível e constitui a interface entre a decisão e a lógica do sistema (o dito centro de processamento On-Board, na figura 2).

O DUNE (consultar Anexo B) é responsável pelo interface entre o sistema USV e os sensores e atuadores, bem como o controlo de execução. Em suma, faz a tradução entre o planeamento de alto nível e o controlo de baixo nível, neste sistema.

As alterações baseiam-se em:

1. O Planeamento e Controlo da missão passa a ser feito pelo Neptus;
2. O LattePanda passa a ser agora o centro de lógica do sistema: recebe o planeamento de alto nível do Neptus e transforma-o em informação comprehensível para os atuadores, recorrendo ao DUNE;
3. O Arduino torna-se o intermediário entre o centro de lógica e os periféricos do sistema, fazendo a leitura dos sensores (LT1000 e sonares) e a escrita para os atuadores (motores e lemes);
4. O Open CPN dá lugar às cartas náuticas incorporadas no Neptus.
5. A comunicação entre as partes de alto e baixo nível do sistema realiza-se com LoRA, até um alcance de 8Km. Em caso de perda de comunicação, será utilizada comunicação via satélite.



Com esta solução pretende-se dotar o catamarã de mais autonomia de navegação, tornar a sua ação mais precisa, as tarefas de controlo mais fáceis e promover a segurança por rotinas de fail-safe.

A Apêndice 1: Neptus

Neptus é um software desenvolvido pelo laboratório LSTS, que se foca essencialmente no comando, controlo e monitorização de sistemas marinhos não tripulados. A plataforma tem a vantagem de ser flexível e permitir uma fácil integração e comunicação entre vários veículos por: ser desenvolvida em Java, correr simultaneamente em sistemas operativos Windows e Linux e basear a sua comunicação no protocolo IMC. De igual modo, o Neptus dá suporte em todo o ciclo de vida da missão: planeamento, simulação e execução, sendo que recorrer a uma única ferramenta de alto nível constitui uma vantagem para sistemas críticos como o USV em questão. Desta forma, o caráter adaptativo e flexível vai ao encontro com os requisitos do cliente:

1. N1: Assegurar a interoperabilidade do USV entre os veículos da Marinha

Ao garantir a homogeneidade entre o protocolo de comunicação, garante-se também a interacção e interoperação deste catamarã com a restante frota. Adicionalmente, o planeamento de missões integrado dos veículos é agilizado e tornado mais eficiente pela utilização de uma única plataforma de controlo e comando

2. N2: O USV deverá dirigir-se, autonomamente, para os pontos de destino

É o Neptus que, neste sistema, tem a função de planear as rotas a adoptar pelo catamarã, tornando-se assim uma ferramenta fundamental para o controlo de alto nível e da navegação autónoma deste.

Esta framework recorre a diversos standards e normas, o que constitui uma preocupação no que toca aos veículos da Marinha. Um dos standards de navegação integrados é o ENC S-57, que permite a consulta de informação hidrográfica e escolha fundamentada da atuação no que toca às capacidades do veículo, como a manobrabilidade e duração da bateria, que vai ao encontro do terceiro requisito do cliente:

3. N3: O USV deve ser eficiente a nível energético

O Neptus permite a criação de um perfil único para cada USV, que contém informação sobre o sistema, nomeadamente a capacidade da bateria e as especificações da manobrabilidade. Uma das funcionalidades do Neptus é a possibilidade de planear missões recorrendo a diferentes tipos de dados geográficos, incluindo standards de navegação integrados. Estes planos podem ser simulados e validados de acordo com as capacidades do sistema. Desta forma, é possível identificar rotas e manobras mais eficientes a níveis energéticos.

A.1 Papel do Neptus no projeto

De seguida, enunciam-se quais as características principais do Neptus, que são úteis no desenvolvimento do projeto.

A.1.1 Planeamento de missões futuras

Uma das funcionalidades oferecidas pelo Neptus e identificadas como extremamente úteis ao desenvolvimento deste projeto é o planeamento de missões. Através da consola de comandos do operador, que constitui parte da interface do operador, é possível definir planos constituídos por simples manobras, como waypoints, ou manobras mais complexas.

A.1.2 Visualização de dados em tempo real

Ainda na interface do operador, no mapa de missão, é possível analisar os dados recebidos em tempo real do catamarã, nomeadamente a sua localização. Neste caso, será definido como mapa de missão a carta náutica adequada, que permite a criação de uma representação virtual do local da missão.

A.1.3 Revisão de informação de missões passadas

Numa segunda interface, de análise e revisão de dados, será possível analisar os dados produzidos ao longo da missão. Para além disso, são apresentadas nesta secção as mensagens IMC transmitidas ao longo da comunicação, dado que o Neptus faz um *logging* destas, permitindo assim, realizar testes na fase de desenvolvimento, e recolha de dados aquando da sua normal operação.

A.1.4 Conceitos a destacar

Para melhor esclarecimento do tipo ferramentas que o Neptus dispõe, abaixo apresentam-se alguns dos conceitos importantes ao operar este sistema.

1. Registo e Histórico

Um registo histórico contém todas as mensagens para um dispositivo e plano específicos. Cada vez que um plano muda ou um veículo não está em movimento, um novo registo é criado. Este registo é útil para fazer a monitorização do sistema, quer para fins de debug, na fase de desenvolvimento, quer na deteção de problemas, na sua normal operação.

2. Missão

Uma missão é a combinação de um local, planos e todos os elementos do mapa de missão usados. Cada missão deve ter um único objetivo e local. Esta constitui uma hierarquia para organizar as tarefas a executar pelo sistema. O Neptus fornece suporte para diferentes fases do ciclo de vida da missão, tais como a fase de planeamento, execução e de análises de dados. No mesmo ambiente integrado é possível ser usado para preparar a missão, monitorar a execução em tempo real e revisar os dados que foram conseguidos tanto no local como posteriormente, num ambiente controlado.

3. Mapas

- Recorre-se ao mapa mundial presente no Neptus, porque permite consulta offline da informação de navegação.
- Utiliza-se o mapa da missão para armazenar no arquivo de missão e tornar possível adicionar vários elementos do mapa: marcas, geometrias e imagens.

4. Plano e manobras

Uma **manobra** é uma ação básica da qual são compostos os planos.

Um **plano** é uma sequência de manobras.

A.1.5 Parâmetros dos veículos/Configurações

O Neptus contém e armazena as configurações internas dos veículos. Permite que se acceda ao veículo e configurar remotamente.

Existem quatro níveis de acesso: Global, Inativo, Plano (configurável em um plano) e Manobra (configurável na manobra em um plano).

A.1.6 Interfaces

Em suma, o Neptus tem o papel de interagir com diversas partes do sistema, sendo responsável pela criação de interfaces tanto com o operador, como na análise de dados. Como resumo, indica-se a figura 3.

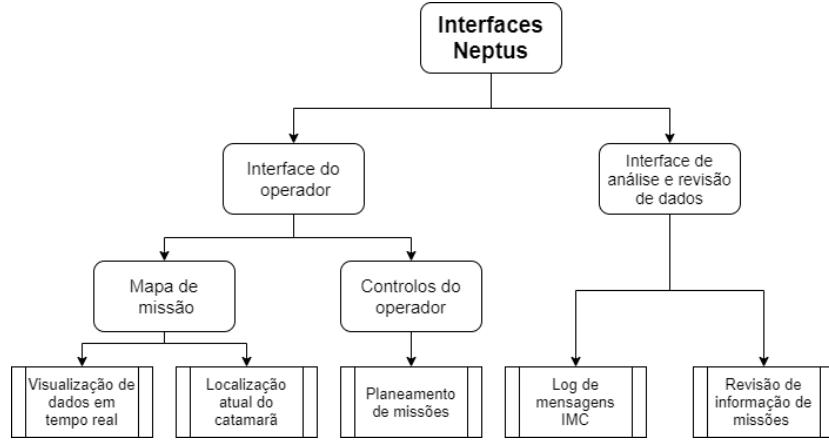


Figura 3: Interfaces do Neptus

B Apêndice 2: Dune

B.1 Introdução

Unified Navigational Environment (DUNE) é o software *on-board* que corre no veículo. É o responsável por todas as interações com sensores, carga e atuadores, bem como pelas comunicações, navegação, controlo, manobragem, execução do plano e supervisão do veículo.

B.1.1 Arquitetura

No DUNE, as funcionalidades dividem-se em unidades atómicas de execução - tarefas. Cada tarefa implementa uma função específica e independente do resto do sistema, numa *thread* de execução própria. Quando uma tarefa precisa de informação exterior (dados de sensores, dados que resultam de outras tarefas, etc.) estas são comunicadas por mensagens, cujo formato é especificado pelo protocolo IMC, de acordo com uma arquitetura semelhante ao do padrão de design de software *publish/subscribe*, em que tarefas que produzem mensagens, enviam-nas (i.e. *publish*), de forma transparente, para tarefas que necessitam delas (i.e. tarefas que estão *subscribed*), sendo que nem os receptores nem os emissores têm conhecimento uns dos outros e funcionam de forma independente. Segue um exemplo na figura 4.

Esta arquitetura permite que seja desenvolvido software de forma modular. Esta característica é favorável ao projeto, já que torna mais fácil o desenvolvimento paralelo das funcionalidades, e fomenta a escalabilidade do mesmo, no futuro. Adicionalmente, permite testar as tarefas de forma unitária (*unit testing*).

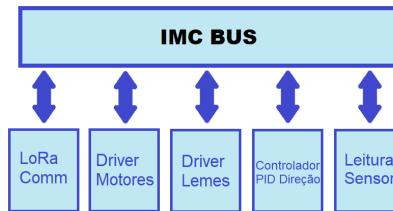


Figura 4: Exemplo de cinco tarefas independentes, que comunicam entre si.

B.1.2 Perfis e Configuração

O DUNE possui o benefício da reutilização de software desenvolvido para outros sistemas, alterando apenas os parâmetros que caracterizam o sistema em questão (no caso específico, um catamarã).

Pode também correr com diferentes perfis que, tirando vantagem do mecanismo de configuração, ativam e desativam conjuntos de tarefas conforme o objetivo do perfil. Por exemplo, o DUNE pode correr em modo de simulação, que desativa todos os drivers dos sensores e atuadores, e substitui-os por tarefas de simulação. Pode também correr em modo Hardware-in-the-loop, que permite que alguns drivers de sensores e atuadores sejam ativados junto com tarefas de simulação. Estas características são importantes numa perspetiva de desenvolvimento, dado que permitem testar e validar tarefas em modo offline, dispensando-se a utilização do sistema completo: não se compromete a integridade do sistema.

B.2 Tarefas do DUNE

Das principais tarefas do DUNE, a realizar em *threads* no LattePanda, destacam-se:

- Gestão de informação dos sensores - esta tarefa vai receber, atualizar e guardar a informação;
- Controlo dos atuadores - tarefa responsável por assegurar o estado e o correto funcionamento dos atuadores;
- Controlo de direção e velocidade - é nesta tarefa que se encontram os algoritmos de controlo, como PIDs que vão enviar os valores desejados de impulso dos motores e posições dos lemes;
- Manobra de seguimento de uma trajectória - esta tarefa é responsável por definir o rumo da embarcação e velocidade desejada, garantindo uma realização segura da trajetória pretendida;
- Processamento da informação enviada pelo NEPTUS - responsável por descodificar e executar as ordens recebidas do NEPTUS;

Lista de Figuras

1	Estado inicial do sistema	4
2	Estado final do sistema	5
3	Interfaces do Neptus	8
4	Exemplo de cinco tarefas independentes, que comunicam entre si.	8