



**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

## 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

<b>1</b>	<b>Objetivo e Organização do Documento</b>	<b>3</b>
<b>2</b>	<b>Desafio do Projeto</b>	<b>4</b>
<b>3</b>	<b>Conceito de Sistema</b>	<b>5</b>
<b>A</b>	<b>Apêndice 1: Neptus</b>	<b>6</b>
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>B</b>	<b>Apêndice 2: Dune</b>	<b>8</b>
B.1	Introdução . . . . .	8
B.1.1	Arquitetura . . . . .	8
B.1.2	Perfis e Configuração . . . . .	8
B.2	Tarefas do DUNE . . . . .	9
	<b>Lista de Figuras</b>	<b>10</b>

# 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.

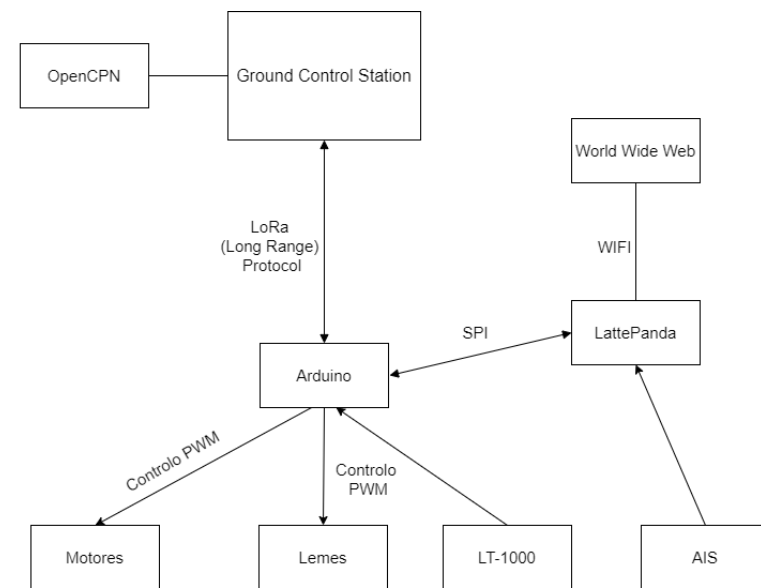


Figura 1: Estado inicial do sistema

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. interatue 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 compreensível 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.

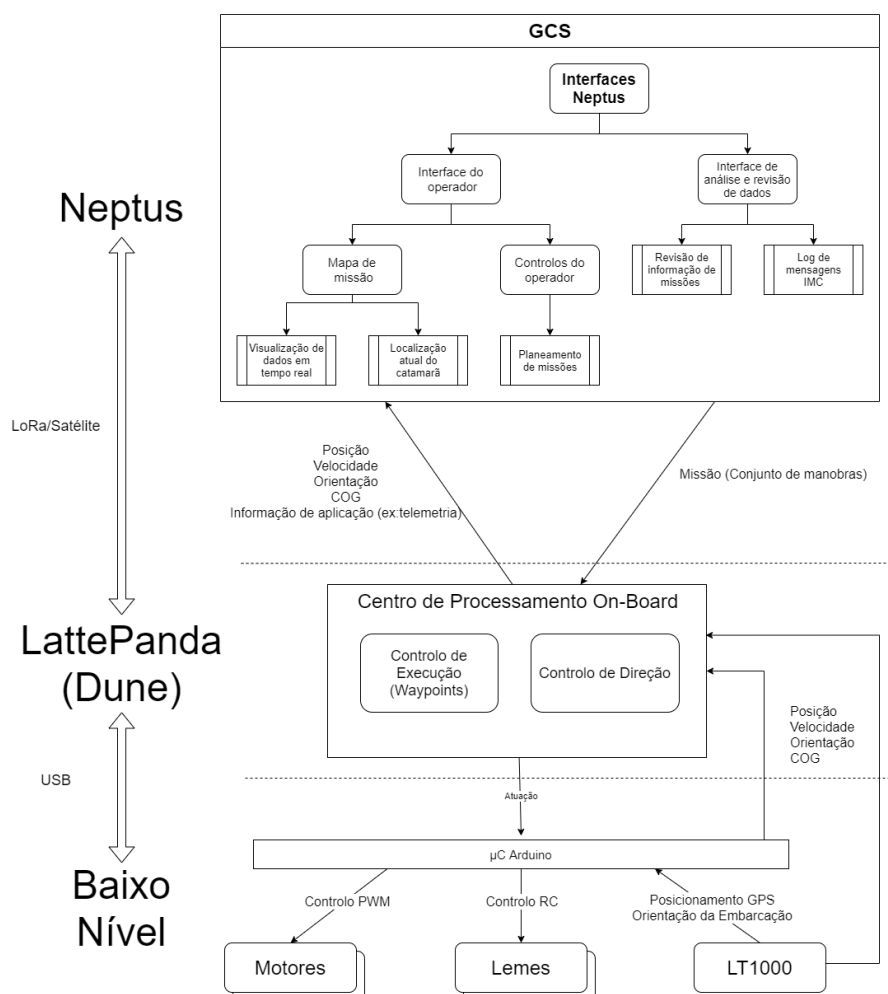


Figura 2: Estado final do sistema

Com esta solução pretende-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.

## 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 carácter 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 aceda 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.



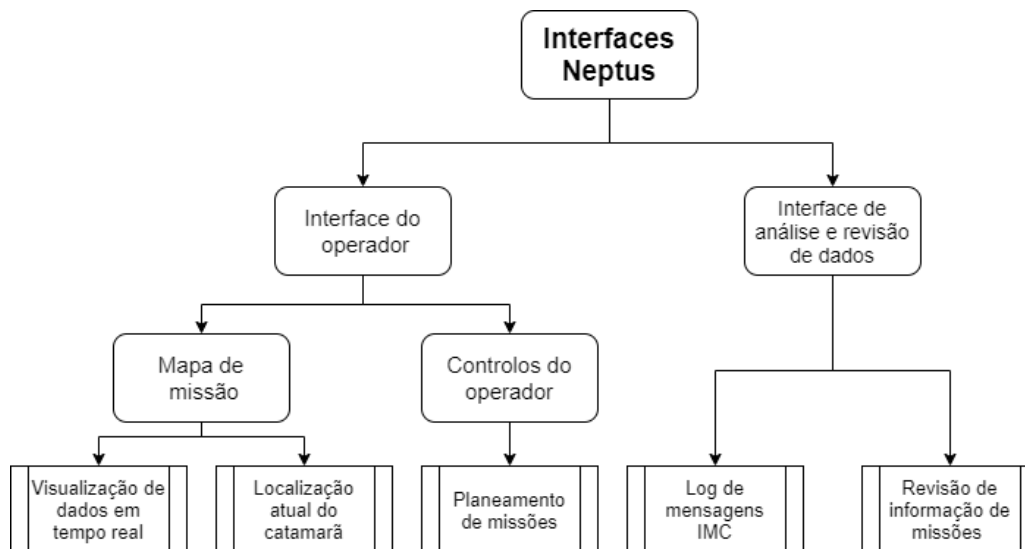


Figura 3: Interfaces do Neptus

## 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 recetores 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*).

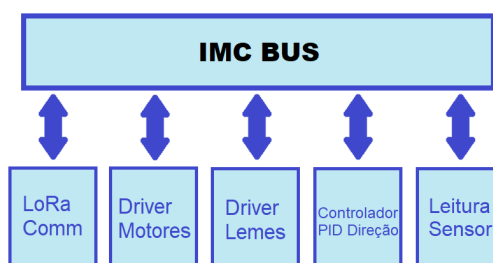


Figura 4: Exemplo de cinco tarefas independentes, que comunicam entre si.

#### 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 trajetória - 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 decodificar 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