



ISEL – INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
ADEETC – ÁREA DEPARTAMENTAL DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

LEIM
LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA
UNIDADE CURRICULAR DE PROJETO

Sistemas de Bases de Dados de Navegação Sea2Future

(eventual) imagem ilustrativa do trabalho – *dimensão*: até 13cm x 4cm

Pedro Nunes (46322)

Samuel Ventura (46339)

Orientador(es)

Professor Carlos Gonçalves (Instituto Superior de Engenharia de Lisboa)

Professor Mário Assunção (Escola Superior Náutica Infante D. Henrique)

julho, 2021

Resumo

Este trabalho enquadra-se no projeto *Sea2Future* da Escola Superior Náutica Infante D. Henrique, tendo sido desenvolvido com o intuito de ser usado em junção com a embarcação autónoma desenvolvida pelos mesmos que produz um conjunto de dados que necessitam de ser processados e armazenados localmente temporariamente.

O processamento é aplicado sobre dados que são extraídos de tópicos publicados pela plataforma *Robot Operating System* (ROS, que é responsável pelo controlo da embarcação em si) que se encontra em funcionamento no *Raspberry Pi* pertencente à embarcação. Estes dados consistem em várias informações relevantes à embarcação captadas pelos vários sensores da mesma.

Situa-se no seguimento de 2 projetos prévios que implementaram o controlo da embarcação de modo remoto. Inicialmente, através de uma aplicação desktop. Posteriormente este conceito foi estendido noutra iteração de modo a permitir que o controlo fosse efetuado através de uma aplicação Android.

Durante uma viagem autónoma realizada pela embarcação existe um conjunto de dados de telemetria que devem ser guardados pela nossa aplicação a desenvolver. Estes dados devem ser primeiro armazenados localmente e posteriormente enviados para o servidor principal remoto para serem analisados e apresentados.

Consequentemente, este trabalho surge devido ao problema de armazenamento dos dados que são registados pela embarcação num ambiente onde existe armazenamento limitado. Adicionalmente deve, também, ser contemplada uma solução de armazenamento local e online, com a devida transmissão de dados quando possível para permitir a gestão de espaço limitado.

Esta aplicação *web* apresenta todos os dados das várias viagens da embarcação de uma forma mais *user-friendly* para facilitar a perceção do significado dos dados e o percurso realizado pela embarcação em si.

Abstract

NOT FINISHED

This module is part of the *Sea2Future* project from Escola Superior Náutica Infante D. Henrique, having been developed with the intention of being used in conjunction with the boat developed by them. . . .

This work arises due to the problem of storing the data that is recorded by the vessel in an environment where there is limited storage in the vehicle, additionally a local and online storage solution should also be contemplated, with appropriate data transmission where possible to enable management of limited space.

In this module a *Python script* was developed to be used in the *Raspberry Pi* belonging to the boat with the function of processing and temporarily storing the data until it is possible to perform an *upload* of said data to an external database. Later, these will be presented in a *web* application for visualisation by a user.

This *web* application presents all the data from the various trips of the vessel in a more *user-friendly* way to make it easier to understand the meaning and implications of the data, and the route taken by the vessel itself.

The processing is applied on data that is extracted from topics published by the *Robot Operating System* (ROS) that is running on the *Raspberry Pi*. This data consists of various information relevant to the vessel captured by the its' various sensors.

Agradecimientos

Eventual texto de agradecimiento ...

Eventual texto de dedicatória . . .

. . . mais texto,

. . . e o fim do texto.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	ix
Lista de Tabelas	xi
Lista de Figuras	xiii
1 Introdução	1
2 Trabalho Relacionado	3
2.1 Aplicações Anteriores	3
2.1.1 Desktop BoatCom	4
2.1.2 Android Boatcom	4
2.2 Robot Operating System	4
3 Modelo Proposto	5
3.1 Requisitos do Projeto	5
3.1.1 Requisitos Funcionais	6
3.1.2 Requisitos não Funcionais	7
3.2 Atributos do Sistema	8
3.3 Casos de Utilização	9
3.4 Fundamentos	10
3.5 Abordagem	10

4	Implementação do Modelo	13
5	Validação e Testes	15
6	Conclusões e Trabalho Futuro	17
A	Um Detalhe Adicional	19
B	Outro Detalhe Adicional	21
	Bibliografia	23

Lista de Tabelas

3.1	Conjunto de requisitos funcionais associados aos componentes da embarcação.	6
3.2	Conjunto de requisitos funcionais associados as componentes da aplicação.	7
3.3	Conjunto de requisitos não funcionais associados à embarcação.	7
3.4	Conjunto de requisitos não funcionais associados à aplicação. .	8
3.5	Tabela de atributos relativos ao projeto.	9
5.1	Uma tabela	15

Lista de Figuras

2.1	Estrutura do projeto DesktopBoatCom	4
3.1	Casos de Utilização da aplicação.	10
3.2	Tabela que demonstra o tamanho considerado para cada dado.	11
3.3	Tabela que demonstra os diferentes tempos possíveis.	11
3.4	Tabela que demonstra os diferentes cenários ponderados.	12
3.5	Tabela que demonstra o espaço ocupado pela totalidade dos dados ao longo de vários períodos de tempo.	12
5.1	Uma figura	15

Capítulo 1

Introdução

Neste trabalho foi desenvolvido um *script* Python para ser usado no Raspberry Pi pertencente à embarcação com a função de processar e armazenar temporariamente os dados até que seja possível realizar um upload dos mesmos para uma base de dados externa. Posteriormente, estes serão apresentados numa aplicação web para a visualização por parte de um utilizador.

Capítulo 2

Trabalho Relacionado

Este projeto surge na continuidade dos projetos anteriores, Desktop BoatCom e Android BoatCom, de modo a permitir uma maior usabilidade e utilidade para o utilizador do projeto Sea2Future como um todo. Neste capítulo descrevem-se as principais funcionalidades dos trabalhos desenvolvidos anteriormente que se relacionam com o projeto desenvolvido, sendo abordado em vários capítulos as partes mais relevantes ao nosso projeto especificamente, bem como os que se enquadram diretamente no projeto, mas numa maior escala. Na secção 2.1 descrevem-se as aplicações desenvolvidas anteriormente e, na secção 2.2 apresenta-se de modo simplificado a arquitetura do *Robot Operating System*.

2.1 Aplicações Anteriores

O projeto de Sistemas de Base de Dados de Navegação surge no seguimento dos projetos desenvolvidos previamente por outros colegas de LEIM em conjunto com a ENIDH. Estes consistiam em possibilitar a comunicação com o embarcação, funcionalidade esta que é essencial ao funcionamento do projeto atual e através desta mesma funcionalidade, mover a embarcação. Nas secções 2.1.1 e 2.1.2 apresentam-se, respetivamente, de forma resumida os projetos anteriores, *Desktop BoatCom* e *Android BoatCom*.

A 1ª iteração destes projetos foi a Desktop BoatCom que inibia a comunicação e movimentação da embarcação através de um aplicativo em contexto de *desktop* em conjunto com um *joystick*, sendo posteriormente este projeto melhorado em forma de ser concebido um aplicação *Android*, providenciando,

assim, uma maior acessibilidade e portabilidade.

Comparativamente, este projeto irá tomar partido de tecnologias implementadas nestes projetos anteriores, nomeadamente a comunicação para permitir que sejam extraídos os dados essenciais da embarcação durante as suas viagens.

2.1.1 Desktop BoatCom

Este projeto consistiu no desenvolvimento de um módulo que permite o controlo da embarcação de maneira remota. Isto era possibilitado através duma aplicação *desktop*, que através dum joystick comunica os *inputs* para a embarcação via Robot Operating System (ROS), possibilitando a sua movimentação. A estrutura que este projeto seguiu encontra-se visível na figura 2.1

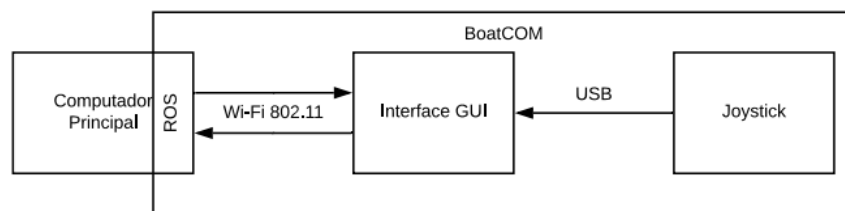


Figura 2.1: Estrutura do projeto DesktopBoatCom

2.1.2 Android Boatcom

2.2 Robot Operating System

O ROS (Robot Operating System) é uma *framework* flexível constituída por várias bibliotecas que facilitam a criação de aplicações para robôs para manipular os mesmos.

Mais concretamente, aplicado ao projeto, é utilizado para permitir a comunicação com a embarcação, possibilitado pelo ROS através da arquitetura que este fornece para a troca de mensagens através do protocolo TCP/IP. Esta arquitetura funciona através da publicação de nós que devem ser subscritos para recolher a informação que está contida neles. Isto pode-se observar através da figura ??.

Capítulo 3

Modelo Proposto

Aqui mostra um caminho que inicia com requisitos (cf., secção 3.1), passa pela aplicação dos fundamentos (cf., secção 3.4) e continua até conseguir transmitir uma visão clara e um formalismo com nível de detalhe adequado a um leitor que tenha um perfil (competência técnica) idêntico ao seu.

Recorra, sempre que possível, a exemplos ilustrativos da utilização do seu modelo. Esses exemplos devem ajudar o leitor a compreender os aspectos mais específicos do seu trabalho.

O modelo aqui proposto deve ser (tanto quanto possível) independente de tecnologias concretas (e.g., linguagens de programação ou bibliotecas). No entanto deve fornecer os argumentos que contribuam para justificar uma posterior escolha (adopção) de tecnologias.

3.1 Requisitos do Projeto

Os requisitos funcionais e não funcionais representam o conjunto das funcionalidades que o projeto deverá implementar. Os requisitos funcionais definem o conjunto de funcionalidades que o sistema desempenha, cada um representado com a sua respetiva referência. Consoante as suas funções, estes encontram-se categorizados em três categorias: evidente, invisível ou adornos. Os requisitos evidentes são aqueles cuja sua utilização no projeto é visível para o utilizador, contrastando com os requisitos invisíveis que têm o propósito contrário. Ambas as categorias são, no entanto, de implementação obrigatória, ao contrário da categoria dos adornos cuja implementação é opcional dado que a sua existência resume-se a funcionalidades menos signifi-

cantes. Os requisitos não funcionais representam a maneira como o sistema realiza cada uma das suas funcionalidades. Assim como os requisitos funcionais, estes podem ser agrupados nas mesmas categorias: evidente, invisível e adornos.

Na secção 3.1.1 demonstra-se o conjunto de requisitos funcionais associados ao projeto, enquanto na secção 3.1.2 o conjunto de requisitos não funcionais.

3.1.1 Requisitos Funcionais

A tabela 3.1 dispõem do conjunto de requisitos funcionais aplicados à embarcação e a tabela 3.2 os requisitos associados à aplicação gráfica a implementar.

Tabela 3.1: Conjunto de requisitos funcionais associados aos componentes da embarcação.

Ref.#	Função	Categoria
R1.1	Registo e processamento dos dados fornecidos pelo veículo.	Invisível
R1.2	Armazenar dados registados numa base de dados local ou remota.	Invisível
R1.3	Criar canal de comunicação veículo-servidor para envio de dados.	Invisível
R2.1	Histórico de viagens com respetivos dados registados e rota realizada.	Invisível

O sistema a implementar tem de ser capaz de processar os dados obtidos (R1.1) e registá-los numa base de dados local à embarcação, ou numa base de dados remota num servidor (R1.2), caso exista a possibilidade de se ligar à *internet*. Desta forma, é necessário contemplar também a existência de uma canal de comunicação entre a embarcação e o servidor (R1.3) de modo a transferir os dados. A existência de várias viagens cria a necessidade de organizar os dados em estruturas ordenadas cronologicamente. Desta forma, a estruturação dos dados adquiridos devem de ser estruturados de forma a originar um histórico para os mesmos (R2.1).

Tabela 3.2: Conjunto de requisitos funcionais associados as componentes da aplicação.

Ref.#	Função	Categoria
R3.1	Listar os dados da viagem na aplicação web utilizando tabelas ou gráficos.	Evidente
R3.2	Registar rota num mapa interativo com base nos dados do GPS.	Evidente
R4.1	Autenticação de utilizador (User/Password).	Invisível

Os dados obtidos do histórico de viagens são listados na aplicação gráfica (R3.1), juntamente com as coordenadas Global Positioning System (GPS) da rota realizada, que é desenhada num mapa interativo (R3.2). Esta aplicação necessita de um sistema de autenticação de utilizador, através da introdução de um nome e palavra-chave. Isto possibilita a autenticação do utilizador de forma a verificar se este é, ou não, válido para o visionamento dos dados registados (R4.1).

3.1.2 Requisitos não Funcionais

A tabela 3.3 apresenta os requisitos não funcionais que dizem respeito às componentes relativas à embarcação, enquanto a tabela 3.4 representa os requisitos relativos ao lado da aplicação a complementar o projeto.

Tabela 3.3: Conjunto de requisitos não funcionais associados à embarcação.

Ref.#	Função	Categoria
R1	Utilização de um script Python para o processamento e envio dos dados para uma base de dados NoSQL.	Invisível
R2	A comunicação deve ser feita com Wi-Fi e a ligação feita diretamente ao ROS.	Invisível

É vital a utilização de um *script* capaz de adquirir os dados de telemetria provenientes dos sensores da embarcação, de modo a processá-los para o posterior envio para uma base de dados que não recorre a Structured Query

Language (SQL) (R1). Dada a necessidade de se enviar os dados para uma base de dados, é preciso realizar uma ligação direta através do Wifi, entre o ROS presente na embarcação e a base de dados que será armazenada num determinado servidor (R2).

Tabela 3.4: Conjunto de requisitos não funcionais associados à aplicação.

Ref.#	Função	Categoria
R3	Estruturação da base de dados de modo a que cada conjunto de documentos siga uma estrutura cronológica de viagens para recolha de dados.	Invisível
R4	Aplicação Web à qual o utilizador se autentica para o visionamento da informação de cada viagem presente na base de dados remota.	Evidente

A estruturação da base de dados é importante pois necessita de apresentar uma estrutura que permite a disposição dos dados recolhidos de forma cronológica e organizada, de modo a facilitar a criação, por exemplo, do histórico de navegação (R3). O visionamento destes dados é efetuado através da aplicação *web* à qual o utilizador válido acede para visualizar, de forma intuitiva, os dados registados pela embarcação (R4).

3.2 Atributos do Sistema

Os atributos definem as características ou dimensões de um sistema. Os atributos dispõem também de um conjunto de classificações que depende do tipo de característica que desempenham no projeto, podendo ser classificados como: "Plataforma", para qualidades a nível das ferramentas a utilizar; "Interação homem-máquina", para propriedades relativas à utilização das ferramentas por parte do utilizador; "Facilidade de Utilização", como o nome indica, características que facilitam a utilização do sistema; ou "Tolerância a Falhas", para características permitem fortalecer o sistema. Para além disso, estes podem ser categorizados como "Obrigatórios" se tiverem de ser contemplados ou "Desejável" para características alcançáveis mas que não necessitam de ser obrigatoriamente atingidas.

A tabela 3.5 define o conjunto de atributos a aplicar a este projeto.

Tabela 3.5: Tabela de atributos relativos ao projeto.

Atributo	Detalhe / Restrição Fronteira	Categoria
Plataforma	Microcontrolador para processar, publicar e subscrever tópicos ROS.	Obrigatório
Interação homem-máquina	Aplicação Web que permite demonstrar a informação registada na base de dados.	Obrigatório
Interação homem-máquina	Gráficos e diagramas na app para visualização.	Desejável
Facilidade de Utilização	Base de dados em NoSQL para armazenar dados.	Obrigatório
Facilidade de Utilização	Serviço de mapas para registo de rotas no mapa da interface gráfica.	Desejável
Tolerância a Falhas	Módulos externos de armazenamento adicional para evitar potenciais lotações de memória.	Desejável

3.3 Casos de Utilização

Os casos de utilização descrevem, de forma visual, o funcionamento de determinados elementos do sistema a desenvolver. São caracterizados pela existência de um ou mais atores (entidades intervenientes) que desempenham um determinado conjunto de funcionalidades na utilização do sistema.

No contexto do projeto, o sistema vai apresentar como atores a embarcação, a plataforma gráfica e um utilizador. A embarcação, que contém o script, é capaz de recolher e processar os dados publicados, saber armazenar os dados localmente e enviar os dados para um servidor externo. A aplicação gráfica utiliza os dados do servidor para os apresentar, de forma gráfica e intuitiva. Consequentemente, o utilizador é capaz de aceder à aplicação de modo a visualizar os dados registados pela embarcação.

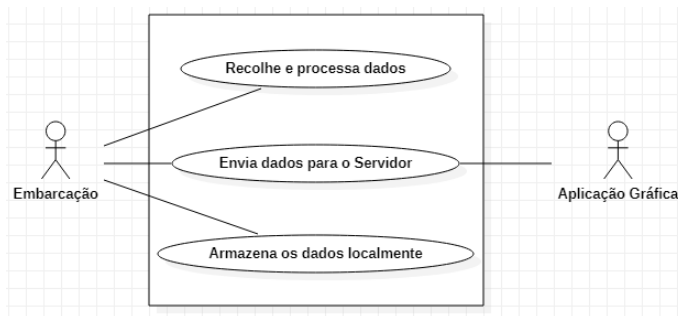


Figura 3.1: Casos de Utilização da aplicação.

3.4 Fundamentos

Aqui o sustento formal (teórico / tecnológico) do trabalho realizado ...

3.5 Abordagem

Nesta secção aborda-se o tipo de formulações que se efetuaram de maneira a ter uma melhor noção do seguimento a tomar no desenvolvimento do/a modelo/estrutura proposto/a, permitindo adicionalmente, que seja possível ter uma análise mais crítica das decisões tomadas pelo grupo no seguimento do desenvolvimento do projeto.

De modo a guiar as opções de armazenamento foi realizado um estudo inicial para obter uma estimativa da capacidade de armazenamento necessária. Para tal, elaborou-se uma folha de cálculo onde foram colocadas todas as variáveis que representam os tópicos gerados pela telemetria da embarcação, bem como o espaço (em bytes) ocupado por cada variável.

De seguida, foram considerados vários cenários de utilização que descreviam a frequência de amostragem das diversas variáveis. Para cada cenário contabilizou-se o consumo de memória por unidade de tempo, por exemplo, quantidade de memória gasta por minuto, hora, semana, mês e ano.

Esta análise permitiu obter uma estimativa da memória a gastar em cada cenário de utilização (mencionar a tabela que demonstra estas coisas).

Nesta folha de cálculo criou-se uma estimativa negativista de quanto espaço seria necessário para armazenar os dados desejados sob as piores circunstâncias possíveis (implicando que todos os dados estariam a ser registados o com a maior frequência possível) em vários períodos de tempo.

Para efetuar estes cálculos começou-se por averiguar qual o tamanho que cada variável ocuparia, que para os dados de IMU, V/A e %Água no casco seria um valor numérico, que ocupa 8 bytes e, para o GPS e AIS optou-se por considerar 40 bytes para cada (sendo escolhido um tamanho de 20 caracteres para estes dados com 2 bytes por carácter). Todos estes valores podem ser vistos na figura 3.2.

Dados	Tipo de Variável	Tipo de Variável (MongoDB)	Espaço ocupado (bytes)
GPS	string	string	40
IMU	float3	number	8
AIS	string	string	40
Corrente e Tensão Elétrica	float3	number	8
Água no Casco	float3	number	8

Figura 3.2: Tabela que demonstra o tamanho considerado para cada dado.

Consecutivamente, procedeu-se a estimativas da frequência do registo em 4 patamares consoante a relevância desse dado numa determinada situação. Criaram-se os patamares *Very High*, *High*, *Medium* e *Low* com os valores de registo por a cada 5, 15, 45 e 120 segundos respetivamente (figura ??).

Prioridade de re	Registo/s
Very High	5
High	15
Medium	45
Low	120

Figura 3.3: Tabela que demonstra os diferentes tempos possíveis.

Com isto em mente criaram-se 4 cenários possíveis para a embarcação, figura 3.4, que estabelecem a importância de cada dado para cada cenário e, adicionalmente, contém o consumo para cada um destes por minuto e o total do somatório dos dados (em bytes) nesse minuto, nesse cenário.

Ocasião	Descrição Exemplo	Dados	Prioridade de registo	Consumo/min	Sum
Típico	Movimento do Veículo	GPS	Very High	480	580
		IMU	Medium	10,7	
		AIS	Medium	53,3	
		Corrente e Tensão Elétrica	High	32	
		Água no Casco	Low	4	
Menos Frequente	Movimento Lento/Parado	GPS	Low	20	52
		IMU	Low	4	
		AIS	Low	20	
		Corrente e Tensão Elétrica	Low	4	
		Água no Casco	Low	4	
Raro	Arranque/Aceleração ou Paragens rápidas	GPS	High	160	409,3
		IMU	Very High	96	
		AIS	Medium	53,3	
		Corrente e Tensão Elétrica	Very High	96	
		Água no Casco	Low	4	
Worst Case Scenario	Tudo a registar o mais frequentemente possível	GPS	Very High	480	1248
		IMU	Very High	96	
		AIS	Very High	480	
		Corrente e Tensão Elétrica	Very High	96	
		Água no Casco	Very High	96	

Figura 3.4: Tabela que demonstra os diferentes cenários ponderados.

Finalmente, com o objetivo de ver a longevidade da solução apresentada, efetuaram-se os cálculos finais que são baseados na estimativa pessimista (*Worst Case Scenario*) para visualizar a quantidade de espaço ocupado pelos nossos dados ao longo dos vários períodos de tempo, sendo estes, 1 hora, 1 dia, 1 semana, 1 mês e, finalmente, 1 ano, figura 3.5

Por hora	Por dia	Por semana	Por mês	Por ano	
74880	1797120	12579840	53913600	646963200	Bytes
0,00007	0,002	0,01	0,05	0,60	GB

Figura 3.5: Tabela que demonstra o espaço ocupado pela totalidade dos dados ao longo de vários períodos de tempo.

Capítulo 4

Implementação do Modelo

Implementação do modelo aqui ...; pode precisar de referir o capítulo 3 ...

Aqui identifica as opções teóricas e justifica as dependências tecnológicas assumidas neste projecto. Descreva com rigor formal e detalhe adequado e faça evidência de tudo o que foi proposto e desenvolvido especificamente no contexto deste projecto. Aqui a ênfase está naquilo que foi de facto concretizado neste projecto.

O leitor quer detalhes de concretização. Ele já está enquadrado no tema (cf., capítulo 2), já conhece os aspectos mais abstractos do sua proposta (cf., capítulo 3) e agora precisa de entender os detalhes para conseguir também interpretar as validações e testes que posteriormente (cf., capítulo 5) lhe irá apresentar.

Capítulo 5

Validação e Testes

Validação e testes aqui ...; pode precisar de referir o capítulo 3 ou alguma das suas secções, e.g., a secção 3.4 ...

Pode precisar de apresentar tabelas. Por exemplo, a tabela 5.1 apresenta os dados obtidos na experiência ...

c_1	c_2	c_3	$\sum_{i=1} c_i$
1	2	3	6
1.1	2.2	3.3	6.6

Tabela 5.1: Uma tabela

Para além de tabelas pode também precisar de apresentar figuras. Por exemplo, a figura 5.1 descreve ...



Figura 5.1: Uma figura

Atenção. Todas as tabelas e figuras, e.g., diagramas, imagens ilustrativas da aplicação em funcionamento, têm que ser devidamente enquadradas no texto antes de serem apresentadas e esse enquadramento inclui uma explicação da imagem apresentada e eventuais conclusões (interpretações) a tirar dessa imagem.

Capítulo 6

Conclusões e Trabalho Futuro

Conclusões e trabalho futuro aqui . . .

Quais as principais mensagens a transmitir ao leitor deste trabalho? O leitor está certamente interessado nos temas aqui abordados. Em geral procurará, neste projecto, pistas para algum outro objectivo. Assim, é muito importante que o leitor perceba rapidamente a relação entre este trabalho e o seu próprio (do leitor) objectivo.

Aqui é o local próprio para condensar a experiência adquirida neste projecto e apresentá-la a outros (futuros leitores).

O pressuposto é o de que este projecto é um “elemento vivo” que recorreu a outros elementos (cf., capítulo 2) para ser construído e que poderá servir de suporte à construção de futuros projectos.

Apêndice A

Um Detalhe Adicional

O “apêndice” utiliza-se para descrever aspectos que tendo sido desenvolvidos pelo autor constituem um complemento ao que já foi apresentado no corpo principal do documento.

Neste documento utilize o apêndice para explicar o processo usado na **gestão das versões** que foram sendo construídas ao longo do desenvolvimento do trabalho.

É especialmente importante explicar o objetivo de cada ramo (“branch”) definido no projeto (ou apenas dos ramos mais importantes) e indicar quais os ramos que participaram numa junção (“merge”).

É também importante explicar qual a arquitetura usada para interligar os vários repositórios (e.g., Git, GitHub, DropBox, GoogleDrive) que contêm as várias versões (e respetivos ramos) do projeto.

Notar a diferença essencial entre “apêndice” e “anexo”. O “apêndice” é um texto (ou documento) que descreve trabalho desenvolvido pelo autor (e.g., do relatório, monografia, tese). O “anexo” é um texto (ou documento) sobre trabalho que não foi desenvolvido pelo autor.

Para simplificar vamos apenas considerar a noção de “apêndice”. No entanto, pode sempre adicionar os anexos que entender como adequados.

Apêndice B

Outro Detalhe Adicional

Escrever aqui o detalhe adicional que melhor explique outro aspecto (diferente do que está no apêndice A) descrito no corpo principal do documento

...

Bibliografia