

# **TRABALHO PARA A DISCIPLINA DE TÉCNICAS DE PROGRAMAÇÃO DO CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO DA UTFPR: *PROPOSTA DE PROJETO FINAL***

Eduardo Vanderlei do Santos Junior, Lucas Pujol de Souza  
eduardojunior@alunos.utfpr.edu.br, lucas.pujol@gmail.com

Disciplina: **Técnicas de Programação** / S73 – Prof. Dr. Robson R. Linhares  
**Departamento Acadêmico de Informática – DAINF** - Campus Curitiba  
**Universidade Tecnológica Federal do Paraná - UTFPR**  
Avenida Sete de Setembro, 3165 - Curitiba/PR, Brasil - CEP 80230-901

**Resumo** – A disciplina de Técnicas de Programação exige o desenvolvimento de um *software*, no formato de um sistema de informação, para fins de aprendizado de técnicas de programação orientada a objetos em Java/C++, incluindo uma introdução para técnicas de engenharia de software e padrões de projeto. Este documento elabora uma proposta de projeto final para a disciplina que consiste em um simulador de porto marítimo, onde o usuário gerencia os navios e funcionários, assim como toda a infraestrutura do porto em si (onde cada navio está atracado, quem está trabalhando nele, entre outros).

**Palavras-chave ou Expressões-chave:** Artigo-Relatório para o Trabalho em Técnicas de Programação, Trabalho Acadêmico Voltado a Implementação em Java/C++, Gerenciamento de portos, Exemplo de Conteúdos de Trabalho de Técnicas de Programação.

**Abstract** - The discipline of Programming Techniques requires the development of a software, in the form of an information system, for purposes of learning object-oriented programming techniques in Java / C ++, including an introduction to software engineering techniques and project patterns. This document elaborates a final project proposal for the discipline that consists of a simulator of sea port, where the user manages the ships and employees, as well as all the infrastructure of the port itself (where each ship is moored, who is working on it, among others).

**Key-words or Key-expressions:** Paper to the Final Project of Técnicas de Programação, Academic Work Related to Java/C++ Implementation, Internal Rules for Work Elaboration, Examples of Elements for the Work of a Programming Course.

## **INTRODUÇÃO**

Para fixação dos conteúdos ministrados em classe, nos foi proposto um projeto final que consiste na elaboração de um software que utilize os conceitos abordados durante a disciplina. Portanto, a equipe decidiu elaborar um artefato de software que tem como área a automatização de processos de um porto marítimo para o gerenciamento das suas diversas frentes de atuação. A linguagem utilizada para o desenvolvimento será a C++ por apresentar melhor familiaridade entre os membros do grupo. Será apresentado o diagrama de classes do projeto, assim como um arquivo de projeto do programa Code::Blocks contendo o código fonte do software em si. Posteriormente será apresentada em sala com o auxílio de uma apresentação no formato de slides para a turma de como se dá o funcionamento do software e as técnicas utilizadas para desenvolvê-lo, assim como as dificuldades encontradas no decorrer de seu desenvolvimento.

Em portos existe a demanda de controle de dados de diferentes áreas humanas, controle de tráfego, financeira, recursos humanos entre outras. Para facilitar administração dessas informações, seria ideal que elas estejam armazenadas na mesma plataforma que se encontra as técnicas que as manipulam, muitas vezes esses dados podem ser sensíveis, gerando transtornos se alguma operação for executada de maneira errônea. Esse projeto tem como objetivo atuar nessa área que em muitos portos é feita de forma mais desautomatizada, interferindo na qualidade e no prazo dos processos realizados.

Nas seções subsequentes serão desenvolvidos os seguintes assuntos: uma explicação do software implementado passando em suas funcionalidades; o desenvolvimento em forma de

paradigma orientado a objetos com a tabela de requisitos previstos inicialmente e; uma conclusão sobre a execução do trabalho.

## EXPLICAÇÃO DO SOFTWARE

Nesta seção será explicado o contexto geral do software, sua forma de execução, explicação das classes e da lógica geral como um todo.

O software simula o contexto corporativo e operacional de um porto marítimo durante a sua atuação. Para o desenvolvimento deste contexto foi assumida duas frentes de trabalho a primeira denominada “Doca” onde ocorre toda a manipulação do operacional do porto, logo tempo de atracagem de navios, controle de cargas, entrada e saída de embarcações. A outra frente é denominada “Escritório”, nessa área é realizada a parte de recurso humanos e financeiro, toda a parte relacionada a controle de caixa, contas a pagar, contas a receber, folha de pagamento, contratações e demissões é realizada por essa frente de trabalho.

Para facilitar o entendimento do artefato vamos fazer um descritivo resumido das classes presentes no programa:

- **Classe Data:** na classe data são descritos os atributos referentes a manipulação de datas, vale salientar que para a criação de datas é utilizado a biblioteca “windows.h”, que retorna a data atual da máquina onde o software está sendo executado.
- **Classe Funcionario:** na classe funcionario é descrito de maneira mais abstrata, recebendo alguns atributos gerais a todas as variações de funcionário. A mesma lógica se aplica às funções nele presente.
- **Classe Escritorio:** apresenta os registros dos funcionários, caixa e é responsável pelas movimentações financeiras e de recursos humanos.
- **Classe Doca:** mantém os registros pertinentes às embarcações que o sistema está gerenciando e faz o controle operacional do porto.
- **Classe Porto:** é uma classe que herda as funcionalidades de “Escritorio” e “Doca” e adiciona algumas funcionalidades de gerar relatório.
- **Classe Menu:** é responsável por toda a lógica de navegação do programa.
- **Classe Navio:** descreve os componentes principais das embarcações atracadas no porto.

O software ainda contempla mais algumas classes que são generalizações das classes aqui descritas, para evitar a repetição de conceitos, foi escolhido não as descrever no documento, porém elas serão explicadas na apresentação do trabalho.

Para o melhor entendimento por completo do artefato, se faz necessário entender o método de navegação do aplicativo. Para que isso ocorra será explicado algumas das principais telas do software.

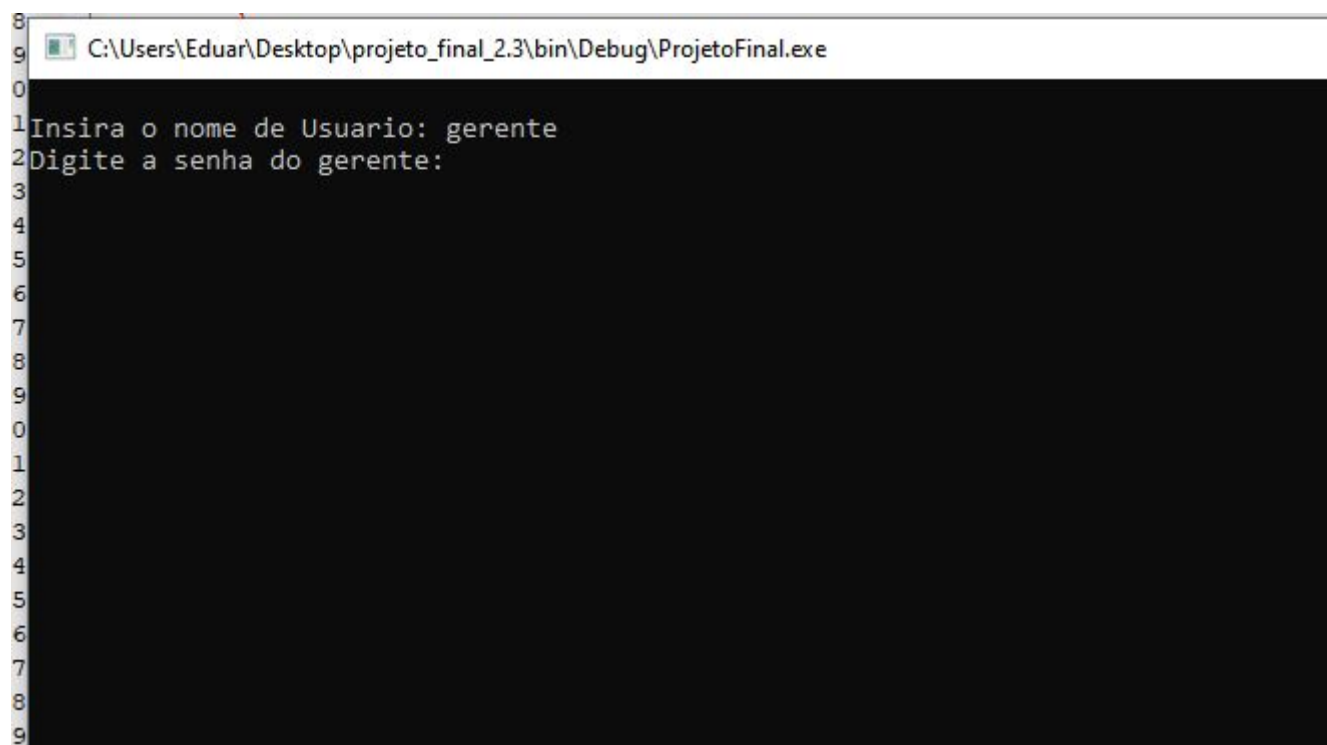


Figura 1- Tela de login.

A figura 1 representa a tela do primeiro contato do usuário com o artefato, nele o usuário pode acessar o sistema. Existe 2 usuários pré-definidos o “gerente” que detém o acesso ao controle de navios e de relatórios e o usuário “rh” que tem acesso ao controle de recursos humanos.

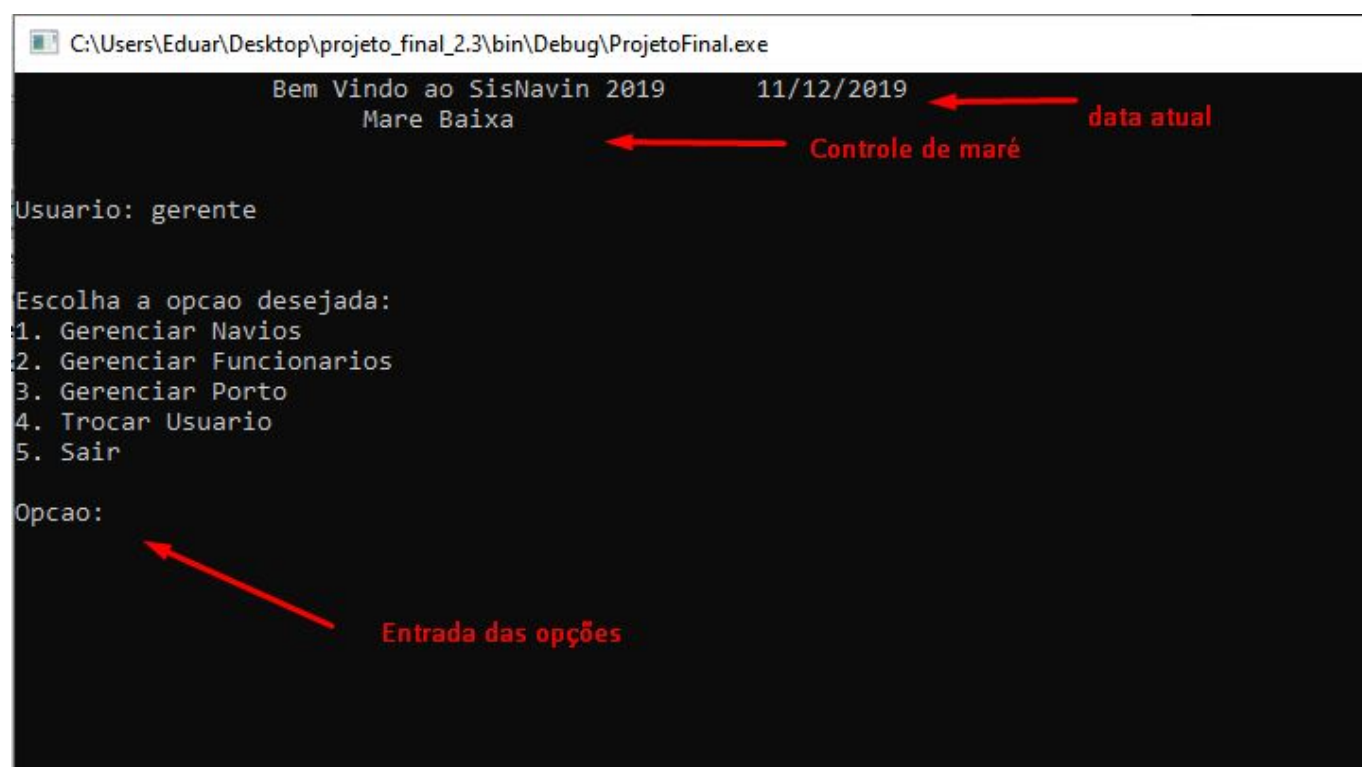


Figura 2- Menu principal.

A figura 2 mostra a tela que o usuário tem acesso aos menus das duas frentes de trabalho de software controle operacional e controle de recursos. Além disso vale salientar alguns recursos dessa tela, como a entrada de dados, o controle da maré e atualização de data baseada no sistema operacional.

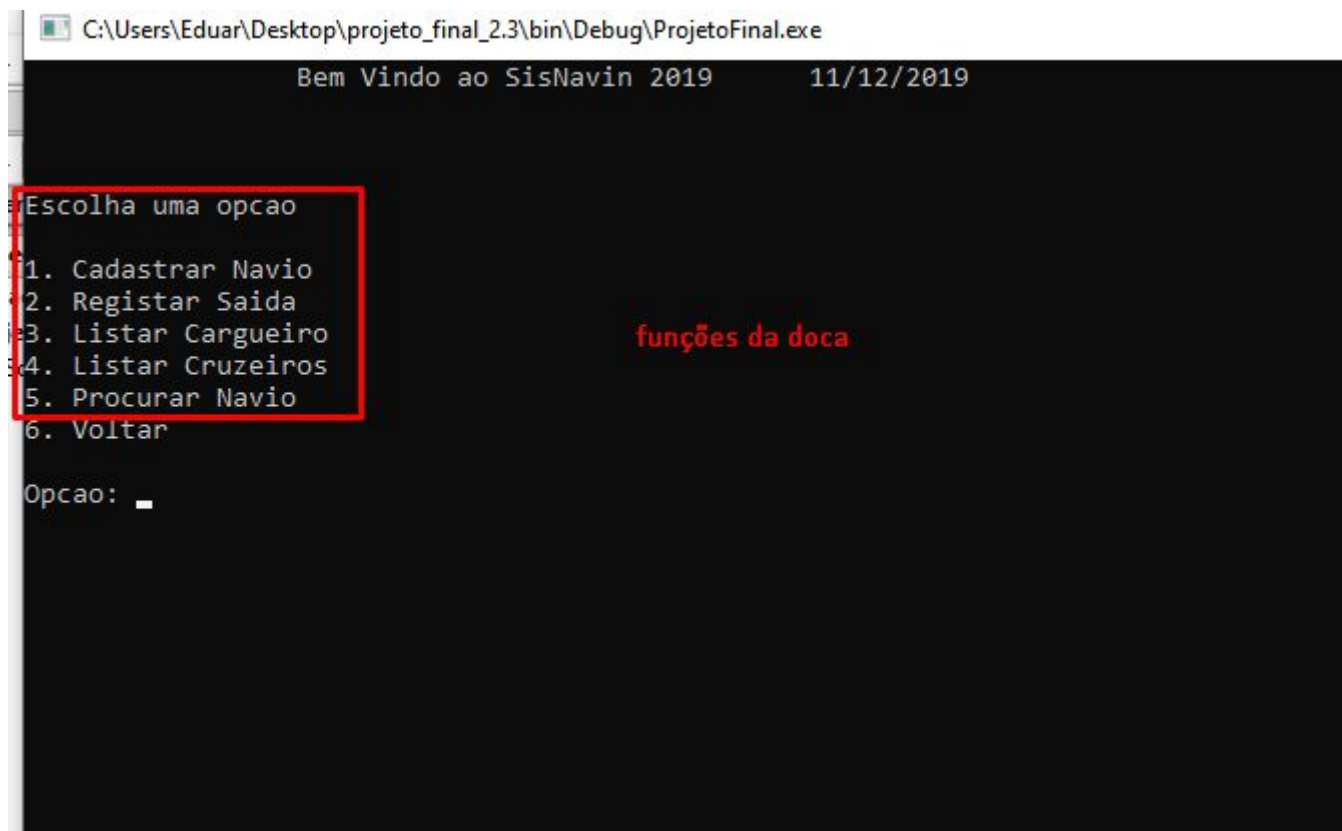


Figura 3-Menu gerenciar navios.

Ao seleccionar a opção 1 na tela da figura 2 será apresentado a tela da figura 3, onde o usuário poderá navegar em opções de controle de operacional do porto.

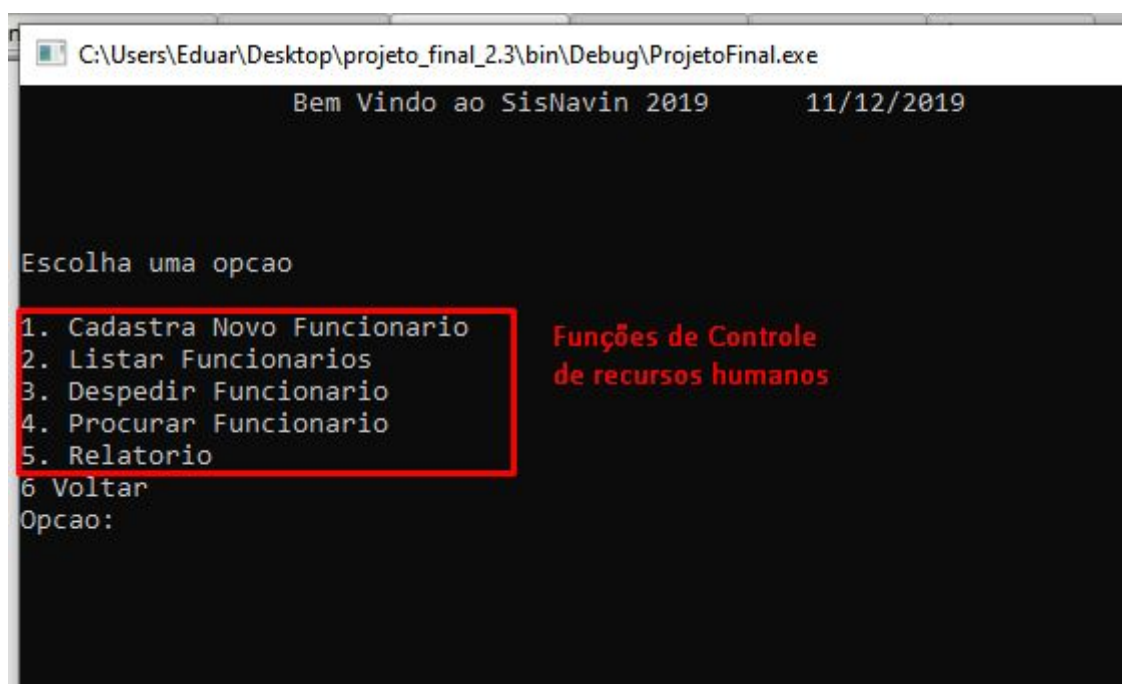


Figura 4- Menu de Funcionários.

Ao seleccionar a opção 2 na tela da figura 2 será apresentado a tela da figura 4, onde o usuário poderá navegar em opções de controle de recursos humanos do porto.

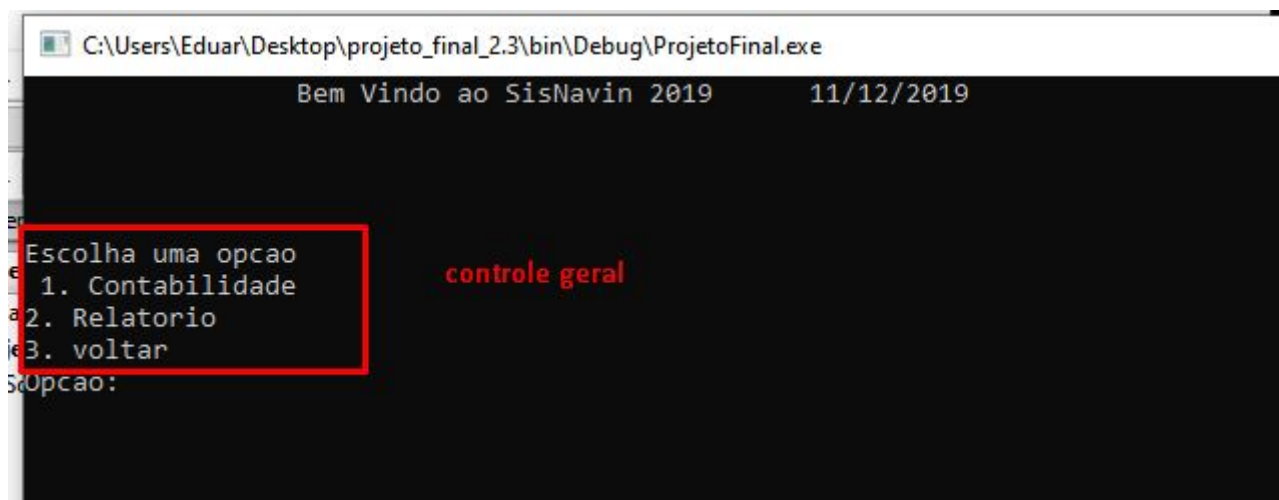


Figura 5- Menu do Porto.

Nessa tela da figura 5, o usuário pode acessar as opções relacionadas a financeiro e impressão de relatórios sobre as informações do sistema.

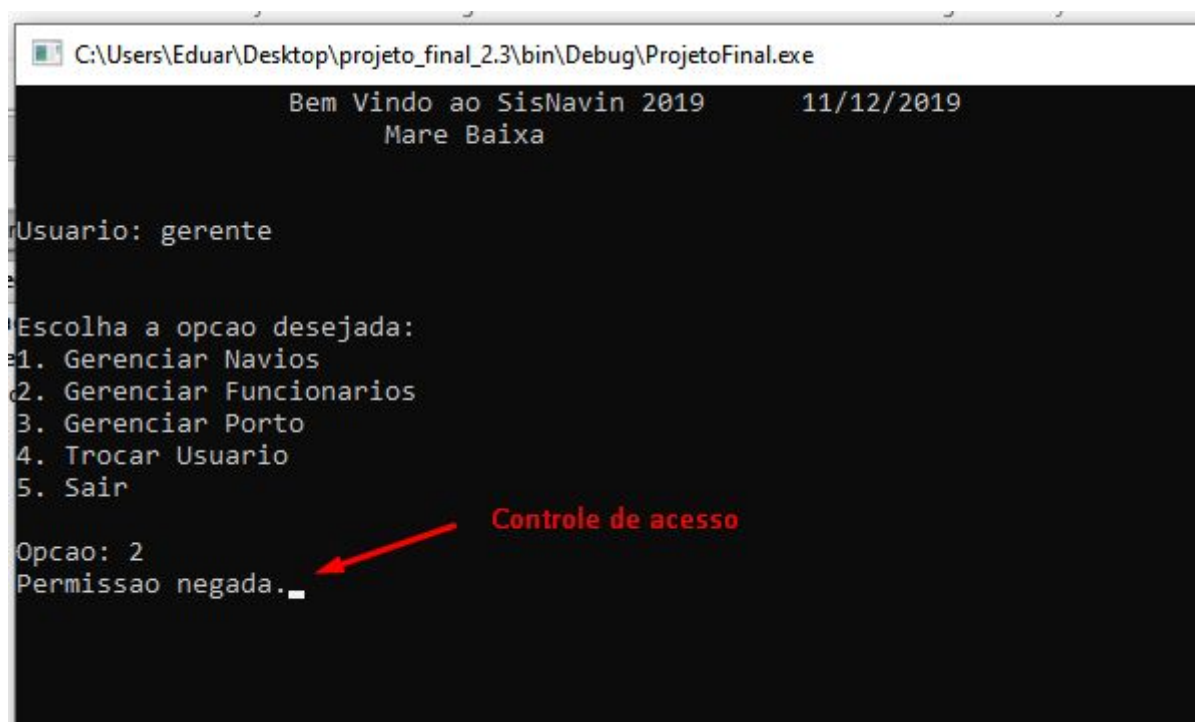


Figura 6- Permissões.

Uma funcionalidade que vale ser salientada, é restrição de acesso, os usuários pré-definidos podem ter acessos apenas a áreas que cabem a sua função.

## DESENVOLVIMENTO DO *SOFTWARE* NA VERSÃO ORIENTADA A OBJETOS

### TABELA DE REQUISITOS:

N.	Requisitos Funcionais	Peso atribuído	Situação final de realização do requisito		
			Total	Parcial	Não realizado
1	Mostrar menu de opções para o usuário.	5 pontos	X		
2	Permitir mais de um usuário	5 pontos	X		
3	Cadastrar e remover novos funcionários	10 pontos	X		
4	Cadastrar e remover novos navios	10 pontos	X		
5	Calcular salários dos funcionários e a permanência dos navios	5 pontos	X		
6	Calcular data atual e há quantos dias os navios estão atracados	5 pontos	X		
7	Manter todos os dados em arquivo em pelo menos 3 formatos diferentes	15 pontos	X		
8	Imprimir relatório de navios e funcionários em pelo menos 2 formatos diferentes	15 pontos	X		
9	Determinar diferentes níveis de acesso às funções do sistema conforme o cargo do usuário	5 pontos	X		
10	Permitir troca de usuário durante o uso do sistema	5 pontos	X		
11	Sistema de pesquisa de funcionários e navios	10 pontos	X		
12	Edição de informações de funcionários e navios pré cadastrados	10 pontos			X
13	Calcular o preço da estadia do navio ao ser removido do porto	2 pontos			X
14	Estipular uma data limite de Permanência dos navios	2 pontos			X
15	Calcular valor a ser pago ao funcionário demitido(FGTS, 13º Férias)	2 pontos	X		
16	Demonstrar como está a maré e a previsão do tempo daquele dia.	2 pontos	X		
17	Determinar teto de gastos com funcionários, impedindo o cadastro de novos caso seja atingido	2 pontos		X	
18	Determinar número máximo de navios atracados	2 pontos	X		
19	Determinar se o navio pode ser aceito no porto dependendo da maré (caso baixa, será recusado)	2 pontos	X		
20	Determinar se o navio pode ser liberado no porto dependendo da maré (caso baixa, será recusado)	2 pontos	X		

## TABELA DE CONCEITOS UTILIZADOS E NÃO UTILIZADOS

Tabela 2. Lista de Conceitos Utilizados e Não Utilizados no Trabalho.

N.	Conceitos	Uso	Onde / O quê
1	Elementares:		
	- Classes, objetos. & - Atributos (privados), variáveis e constantes. & - Métodos (com e sem retorno).	Sim	Todos .cpp
	- Métodos (com retorno <i>const</i> e parâmetro <i>const</i> ). & - Construtores (sem/com parâmetros) e destrutores	Sim	Todos .cpp
	- Classe Principal.	Sim	Main.cpp & Principal.cpp
	- Divisão em .h e .cpp.	Não	
2	Relações de:		
	- Associação direcional. & - Associação bidirecional.	Sim	
	- Agregação via associação. & - Agregação propriamente dita.	Sim	Escritorio.cpp Docas.cpp Menu.cpp
	- Herança elementar. & - Herança em diversos níveis.	Sim	Navio.cpp Funcionario.cpp
	- Herança múltipla.	Sim	Porto.cpp
3	Ponteiros, generalizações e exceções		
	- Operador <i>this</i> .	Não	
	- Alocação de memória ( <i>new</i> & <i>delete</i> ).	Sim	Em todo o trabalho
	- Gabaritos/ <i>Templates</i> criada/adaptados pelos autores (e.g. Listas Encadeadas via <i>Templates</i> ).	Sim	Porto.cpp
	- Uso de Tratamento de Exceções ( <i>try catch</i> ).	Não	
4	Sobrecarga de:		
	- Construtoras e Métodos.	Sim	Cargueiro.cpp

			<b>Cruzeiro.cpp</b>
	- Operadores (2 tipos de operadores pelo menos).	Não	
	Persistência de Objetos (via arquivo de texto ou binário)		
	- Persistência de Objetos.	Sim	
	- Persistência de Relacionamento de Objetos.	Sim	<b>Porto.cpp</b>
5	Virtualidade:		
	- Métodos Virtuais.	Sim	<b>Em todo o trabalho</b>
	- Polimorfismo	Sim	<b>Funções dos funcionários</b>
	- Métodos Virtuais Puros / Classes Abstratas	Não	
	- Coesão e Desacoplamento	Sim	<b>Em todo o trabalho</b>
6	<b>Organizadores e Estáticos</b>		
	- Espaço de Nomes ( <i>Namespace</i> ) criada pelos autores.	Não	
	- Classes aninhadas ( <i>Nested</i> ) criada pelos autores.	Não	
	- Atributos estáticos e métodos estáticos.	Não	
	- Uso extensivo de constante ( <i>const</i> ) parâmetro, retorno, método...	Sim	<b>Em todo o trabalho</b>
7	<b>Standard Template Library (STL) e String OO</b>		
	- A classe Pré-definida <i>String</i> ou equivalente. & - <i>Vector</i> e/ou <i>List</i> da <i>STL</i> (p/ objetos ou ponteiros de objetos de classes definidos pelos autores)	Sim	<b>Porto.cpp Escritorio.cpp</b>
	- Pilha, Fila, Bifila, Fila de Prioridade, Conjunto, Multi-Conjunto, Mapa OU Multi-Mapa.	Sim	<b>Porto.cpp Escritorio.cpp</b>
	<b>Programação concorrente</b>		
	- <i>Threads</i> (Linhas de Execução) no âmbito da Orientação a Objetos, utilizando Posix, C-Run-Time OU Win32API ou afins.	Não	
	- <i>Threads</i> (Linhas de Execução) no âmbito da Orientação a Objetos com uso de Mutex, Semáforos, OU Troca de mensagens.	Não	
8	<b>Biblioteca Gráfica / Visual</b>		



	- Funcionalidades Elementares. & - Funcionalidades Avançadas como: • tratamento de colisões • duplo <i>buffer</i>	Não	
	- Programação orientada e evento em algum ambiente gráfico. OU - RAD – <i>Rapid Application Development</i> (Objetos gráficos como formulários, botões etc).	Não	
	Interdisciplinaridades por meio da utilização de Conceitos de Matemática e/ou Física.		
	- Ensino Médio.	Não	
	- Ensino Superior.	Não	
9	Engenharia de Software		
	- Compreensão, melhoria e rastreabilidade de cumprimento de requisitos. &	Sim	Diagrama de Classes
	- Diagrama de Classes em <i>UML</i> .	Sim	Diagrama.mdj
	- Uso efetivo (quicá) intensivo de padrões de projeto (particularmente GOF).	Não	
	- Testes a luz da Tabela de Requisitos e do Diagrama de Classes.	Não	

## DISCUSSÃO E CONCLUSÕES

Durante o desenvolvimento deste artefato houve algumas dificuldades encontradas pela equipe desenvolvedora. Ao decorrer da reestruturação do software surgiu a demanda de aplicar o conceito de desacoplamento nos arquivos de código fonte, infelizmente houve um insucesso nessa atividade, por parte culpa da falta de experiência da equipe e parte pela a maneira que o artefato vem se estruturando desde o início. Outra barreira que apresentou dificuldades para ser ultrapassada foi a persistência em arquivo, na primeira versão do software ele já apresentava a escrita de dados em arquivo, mas leitura de dados apresentava um problema, pois a construção de uma lógica que trouxesse ao programa a capacidade ler os dados e interpretá-los durante a execução, após algumas horas de trabalho foi alcançado o êxito nesta demanda.

Para trabalhos futuros o grupo chegou ao consenso de duas ideias que agregarão mais valor ao projeto, a primeira é aplicar interdisciplinaridade no projeto, trazer conceitos de outras matérias presentes no curso como Interação Humano-Computador, banco de dados, design, entre outras. Esse tipo de contribuição deixa o trabalho cada vez mais sofisticado e interativo. E segunda ideia é

reestruturar o projeto que faça ele se tornar mais sustentável, ou seja que sejam fáceis a inserção e a manutenção desse artefato.

## **DIVISÃO DO TRABALHO**

<b>Atividades</b>	<b>Responsável</b>
Levantamento de Requisitos	Ambos
Diagrama de Classes	Ambos
Programação em C++	Mais Eduardo que Lucas
Implementação e utilização de Template	Eduardo
Escrita do Trabalho	Lucas
Revisão do Trabalho	Mais Lucas que Eduardo

## **REFERÊNCIAS**

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. 695 p. : ISBN 85-363-0358-1

DEITEL, H. M, DEITEL P.J. C++ Como Programar. 3. ed. Porto Alegre Bookman, 2001. xxviii, 1098 p. + CD-ROM ISBN 85-7307-740-9

<https://stackoverflow.com/>